

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»**

Кафедра інформаційних систем в економіці

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: дистанційна

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему:

Розроблення веб-сайту для віртуальної піцерії з використанням фронтенд-технологій

Здобувачки Моль Каріни Валентинівни

Науковий керівник:

к.е.н., доц.

Помазун О.М.

**Проект допущений до захисту
перед екзаменаційною комісією з
атестації здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент Тішков Б.О.

Київ 2025

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»

ПОГОДЖЕНО:

Керівник проєктної групи(гарант)
освітньо-професійної програми

_____ Помазун О.М.

“ ____ ” _____ 2025 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри

_____ Тішков Б.О.

“ ____ ” _____ 2025 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти ***Моль Каріни Валентинівни***
дистанційної форми навчання
на підготовку кваліфікаційної бакалаврської роботи
на тему: «Розроблення веб-сайту для віртуальної піцерії з використанням фронтенд-технологій»

Тему затверджено наказом ректора Університету від «7» березня 2025р. №468-ст.
Кваліфікаційний бакалаврська робота виконується на матеріалах усіх доступних публікацій та Інтернет джерел

План кваліфікаційного бакалаврської роботи:

Розділ I. *Характеристика та аналіз предметної галузі.*

Розділ II. *Розробка вимог і моделювання інформаційної системи.*

Розділ III. *Проектування та реалізація компонентів підсистеми.*

Об'єкт дослідження: *процес автоматизації замовлення їжі через веб-сайти.*

Предмет дослідження: *методи, технології та підходи до розробки веб-додатків для онлайн-замовлення їжі.*

Мета кваліфікаційної бакалаврської роботи: *розробка веб-сайту для віртуальної піцерії, що забезпечить зручний інтерфейс для перегляду меню, оформлення замовлення та взаємодії з адміністрацією закладу.*

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі І:

- *Дослідити і схарактеризувати предметну область. Зібрати інформацію та вивчити матеріали за тематикою кваліфікаційної бакалаврської роботи.*
- *Проаналізувати літературні джерела та практичний досвід використання віртуальної піцерії і технологій в предметній галузі.*

У розділі ІІ:

- *Проаналізувати вимоги до інформаційної системи віртуальної піцерії.*
- *Розробити алгоритм задач які перед собою ставить система.*
- *Змодельовати інформаційну підсистему віртуальної піцерії.*

У розділі ІІІ:

- *Розробити інформаційне забезпечення інформаційної системи.*
- *Розробити програмне забезпечення інформаційної системи.*
- *Розробити технічне забезпечення інформаційної системи.*
- *Показати результати реалізації віртуальної піцерії.*

**Завдання підготував
науковий керівник _____**

Помазун О.М.
“10” березня 2025 р.

Завдання одержав

с

т

у

д

е

н

т

“10” березня 2025 р.

Моль К.В.

Відгук
про кваліфікаційну бакалаврську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

на тему
Розроблення веб-сайту для віртуальної піцерії з використанням фронтенд-технологій

1. Актуальність теми:

У сучасних умовах цифровізації бізнесу й активного розвитку електронної комерції зростає потреба у веб-системах, які забезпечують зручну взаємодію з клієнтами, автоматизацію процесу оформлення замовлень та підвищення ефективності обслуговування. Створення веб-сайту для віртуальної піцерії є актуальним як з технічної, так і з прикладної точки зору, оскільки дає змогу малому бізнесу впроваджувати сучасні технології без посередництва платформ-агрегаторів. Тематика роботи повністю відповідає потребам ринку та сприяє формуванню прикладних навичок розробки веб-інтерфейсів.

2. Позитивні риси кваліфікаційної бакалаврської роботи:

Робота вирізняється чіткою логікою викладу, узгодженістю між теоретичними положеннями та їх реалізацією в програмному продукті. Авторка продемонструвала вміння аналізувати предметну область, визначати вимоги до інформаційної системи, розробляти її архітектуру та реалізовувати користувацький інтерфейс із застосуванням сучасних веб-технологій. До переваг також можна віднести правильне структурування змісту, якісне оформлення ілюстративного матеріалу, використання UML-діаграм, а також здатність здобувачки самостійно приймати технічні рішення щодо реалізації функціональних компонентів веб-сайту.

3. Наявність самостійних розробок автора

Здобувачка самостійно реалізувала повнофункціональний веб-застосунок для віртуальної піцерії, включаючи розробку структури інтерфейсу, логіки обробки замовлень, навігації та візуального оформлення. Усі ключові компоненти проекту – від моделювання системи до написання коду HTML, CSS і JavaScript – створено без використання сторонніх шаблонів або фреймворків. Власноруч побудовані діаграми прецедентів, класифікація функціональних елементів, формалізація вхідної та вихідної інформації свідчать про високий ступінь самостійності виконання. Це демонструє не лише технічну компетентність, а й здатність автора до комплексного вирішення практичних завдань.

4. Цінність теоретичних висновків та практичних рекомендацій:

У роботі сформульовано обґрунтовані теоретичні висновки, які свідчать про розуміння принципів побудови інформаційних систем, особливостей веб-розробки та специфіки автоматизації процесів онлайн-замовлення. Практичні рекомендації, викладені в заключній частині, мають прикладну спрямованість і можуть бути використані як для реального впровадження в ресторанному бізнесі, так і як базис для подальшої розробки подібних інформаційних систем. Результати роботи мають методичну цінність для навчального процесу з дисциплін, пов'язаних з веб-програмуванням.

5. Наявність недоліків:

Під час виконання роботи здобувачка загалом продемонструвала високий рівень підготовки, однак у роботі можна відзначити окремі незначні недоліки. Зокрема, у теоретичній частині деякі положення подані без достатнього порівняння з аналогічними сучасними рішеннями, а технічні характеристики інструментів розглянуто доволі стисло. Крім того, доцільним було б глибше висвітлити питання масштабованості та підтримки сайту в умовах зростання кількості користувачів. Вказані зауваження не знижують цінності виконаної роботи і не впливають на досягнення поставленої мети

6. Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК:

Кваліфікаційна бакалаврська робота Моль Каріни Валентинівни відповідає вимогам освітньої програми та критеріям, визначеним для підсумкової атестації. Робота має завершений характер, демонструє самостійність мислення, професійну компетентність та здатність автора до практичного застосування отриманих знань. **Робота заслуговує на позитивну оцінку та допускається до захисту перед екзаменаційною комісією.**

Науковий керівник

(підпис)

доцент, к.е.н. Помазун О.М.

“ ____ ” _____ 20__ р.

Рецензія
на кваліфікаційну бакалаврську роботу
здобувача вищої освіти
Моль Каріни Валентинівни

Тема: Розроблення веб-сайту для віртуальної піцерії з використанням фронтенд-технологій.

Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення

З огляду на стрімке зростання попиту на онлайн-сервіси в ресторанній сфері, розроблення веб-сайту для оформлення та адміністрування замовлень піци є надзвичайно актуальним завданням. У роботі порушено важливе прикладне питання впровадження інформаційних технологій у сфері громадського харчування. Обрана тема дозволяє продемонструвати комплексне застосування знань у галузі веб-розробки та моделювання інформаційних систем. Її практична цінність посилюється тим, що результати можуть бути використані для реального бізнесу без суттєвих доопрацювань.

Якість проведеного дослідження

Дослідження виконано на достатньо високому рівні. У першому розділі представлено аналіз предметної галузі та визначено ключові аспекти функціонування онлайн-сервісів для замовлення їжі. Другий розділ містить грамотно сформульовані вимоги до системи та варіанти їх реалізації, з використанням засобів структурного й поведінкового моделювання. У третьому розділі представлено результат програмної реалізації веб-сайту з урахуванням функціональних сценаріїв взаємодії користувача та адміністратора. Авторка продемонструвала володіння практичними навичками, чітко дотримувалась поставлених цілей і забезпечила відповідність отриманих результатів завданням дослідження.

Позитивні риси кваліфікаційної бакалаврської роботи

Робота відзначається цілісністю структури, логічною послідовністю викладу та поєднанням теоретичного аналізу з практичною реалізацією. Авторка грамотно визначила предмет дослідження, обґрунтувала вибір методології та сучасних технологій розробки веб-застосунку. Усі етапи створення інформаційної системи – від постановки задачі до її програмної реалізації – виконано послідовно, з дотриманням вимог проєктування. Варто відзначити наявність структурованої документації, використання моделей UML, а також оформлення повноцінного інтерфейсу, що відповідає сучасним стандартам веб-розробки. Робота має прикладну цінність і демонструє сформовані фахові компетентності.

Зауваження

До незначних недоліків можна віднести обмежене висвітлення питань безпеки користувацьких даних та інтеграції з серверною частиною, що є важливим аспектом для реального впровадження системи. Також у роботі варто було б детальніше описати методи тестування працездатності веб-застосунку. В окремих фрагментах теоретичної частини спостерігається надмірна загальність формулювань. Незважаючи на ці зауваження, загальний рівень виконання роботи залишається високим, а виявлені недоліки не впливають суттєво на якість та завершеність проєкту.

Практична значимість висновків і рекомендацій

Теоретичні положення роботи побудовано на базі аналізу літературних джерел і практичного досвіду у сфері створення веб-сайтів для закладів громадського харчування. Авторка вдало систематизувала знання щодо структури інформаційних систем, їх функціональних компонентів і вимог до інтерфейсів користувача. Практичні рекомендації, що стосуються логіки роботи сайту, структури меню, інтеграції з базою даних і зручності користування, є релевантними для малого бізнесу, що прагне цифровізації без значних витрат. Висновки можуть бути використані також у навчальному середовищі для демонстрації прикладу завершеного веб-проєкту.

Місце роботи та посада рецензента

Системний адміністратор

Компанія Тежноторг-Дон _____ Кіндрицький А. В.
(підпис, ПІБ)

Підпис засвідчую: _____
(посада, підпис)

Місце печатки організації, де працює рецензент

АНОТАЦІЯ

Кваліфікаційної бакалаврської роботи студентки 4 курсу
Навчального-науково інституту «Інститут інформаційних технологій в
економіці»

Моль Каріни Валентинівни, виконаної на тему:
Розроблення веб-сайту для віртуальної піцерії з використанням фронтенд-технологій»

Київ: кафедра інформаційних систем в економіці, 2025р.

Кваліфікаційна бакалаврська робота присвячена розробці веб-сайту для віртуальної піцерії з метою забезпечити зручний інтерфейс для перегляду меню, оформлення замовлення та взаємодії з адміністрацією закладу. Дипломний проект включає в себе три взаємопов'язані розділи, кожен з яких сприяє досягненню загальної мети дослідження.

Перший розділ починається з детального огляду предметної області, включаючи аналіз завдань, які потрібно вирішити, та огляд наявних програмних засобів, які підходять для виявлення проблем. Ретельно досліджуючи цю область, розділ створює основу для подальших етапів дослідження.

Наступний розділ присвячений проектуванню системи та детальному розгляду обраної методології. Його метою є обґрунтування вибору методології детально аналізуючи та вказавши доцільність її використання в даному проекті. Також розділ зосереджується на розробці архітектурної платформи, формулюючи чітке завдання, яке висловлює основні проблеми для вирішення та підвищує ефективність процесу.

Третій розділ є основним цього проекту, адже в ньому показано розробку технічного та програмного забезпечення, організаційної підтримки, необхідної для реалізації функціональних можливостей сайту. Також в ньому проведено тестування деяких об'єктів сайту та показана повна його реалізація, разом з елементами та їхнім описом.

У висновках вказані результати цього дослідницького проекту, та були надані цінні рекомендації щодо доцільності, потенційних переваг впровадження розробленого веб-сайту.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 72 сторінки, 10 таблиць, 46 рисунків, список літератури із 31 джерела, 8 додатків

«РОЗРОБЛЕННЯ ВЕБ-САЙТУ ДЛЯ ВІРТУАЛЬНОЇ ПІЦЕРІЇ З ВИКОРИСТАННЯМ ФРОНТЕНД-ТЕХНОЛОГІЙ»

Перелік ключових слів: веб-сайт, віртуальна піцерія, Java-Script, HTML, CSS, проектування, користувацький інтерфейс, тестування, прототип.

Предметом дослідження є методи, технології та підходи до розробки веб-додатків для онлайн-замовлення їжі.

Об'єктом дослідження є процес автоматизації замовлення їжі через веб-сайти.

Мета кваліфікаційної бакалаврської роботи полягає в розробці веб-сайту для віртуальної піцерії, що забезпечить зручний інтерфейс для перегляду меню, оформлення замовлення та взаємодії з адміністрацією закладу.

Завданням кваліфікаційної бакалаврської роботи є:

- Дослідити предметну область і проаналізувати існуючі підходи до автоматизації онлайн-замовлення їжі.
- Визначити функціональні та нефункціональні вимоги до веб-сайту.
- Спроекувати архітектуру веб-системи, включаючи структуру бази даних (за необхідності), поведінкові моделі та дизайн користувацького інтерфейсу.
- Реалізувати веб-сайт на основі HTML, CSS та Java Script.
- Провести тестування розробленої системи для виявлення помилок і перевірки відповідності вимогам.

Апаратні та програмні засоби, що використовувалися при проектуванні: Microsoft Windows 11, lucidchart, текстовий редактор Visual Studio Code, браузер для тестування (Google Chrome), а також системи контролю версій Git і GitHub

Результати досягнуті в процесі роботи: було успішно спроектовано та реалізовано веб-сайт для віртуальної піцерії, який включає в себе функції перегляду меню, додавання товарів до кошика, оформлення замовлень, а також зручний інтерфейс для користувачів. Всі функції були протестовані, і система відповідає вимогам до зручності користування та ефективності.

Розроблений веб-сайт може бути використаний: для створення онлайн-сервісів для продажу піци або інших продуктів, що потребують інтерактивного інтерфейсу для замовлень. Результати проектування також можуть бути застосовані для навчання веб-розробці, зокрема для реалізації практичних завдань з використанням HTML, CSS і Java Script.

Рік виконання кваліфікаційної бакалаврської роботи: 2025.

Рік захисту кваліфікаційної бакалаврської роботи: 2025.

3MICT

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 84

Аналіз літературних джерел та практичного досвіду використання віртуальних піцерій і

A P P A L T H E O R I C S

.....

.....

100

.....

ВСТУП

У сучасному світі інформаційні технології відіграють важливу роль у розвитку бізнесу, автоматизації процесів та підвищенні рівня обслуговування клієнтів. Особливо це стосується ресторанного бізнесу, де цифрові платформи дозволяють спростити процес замовлення, підвищити швидкість обслуговування та покращити взаємодію з клієнтами. У зв'язку з цим створення веб-сайтів для онлайн-замовлення їжі є актуальною задачею для сучасних закладів харчування.

Дана робота присвячена проектуванню та розробці веб-сайту для віртуальної піцерії на основі технологій HTML, CSS та Java Script. Вибір цієї теми обумовлений зростаючою популярністю онлайн-сервісів для замовлення їжі та необхідністю створення зручних і функціональних веб-ресурсів для ресторанного бізнесу.

Актуальність дослідження: ринок онлайн-замовлень їжі стрімко розвивається, і споживачі все частіше віддають перевагу сервісам, що дозволяють швидко та зручно оформити замовлення без необхідності телефонного дзвінка або особистого відвідування закладу. Віртуальні піцерії та ресторани мають потребу в якісних веб-рішеннях, що забезпечують не лише привабливий інтерфейс, а й ефективну обробку замовлень.

Наразі існує низка сучасних інформаційних систем для автоматизації процесу онлайн-замовлення їжі, зокрема глобальні платформи доставки (Glovo, Uber Eats, Bolt Food), які дозволяють ресторанам приєднуватися до їхніх сервісів, проте потребують сплати комісії.

Індивідуальні веб-сайти ресторанів, які надають можливість оформлення замовлення безпосередньо на сайті закладу, зменшує витрати на посередників і дозволяє налаштовувати власну систему лояльності та зацікавити споживачів.

Попри існуючі рішення, багато малих та середніх закладів потребують власних веб-сайтів, які відповідатимуть їхнім бізнес-процесам і дозволятимуть

контролювати всі етапи обробки замовлення. Саме тому, розробка спеціалізованого веб-сайту для піцерії є актуальним завданням.

Мета дослідження є розробка веб-сайту для віртуальної піцерії, що забезпечить зручний інтерфейс для перегляду меню, оформлення замовлення та взаємодії з адміністрацією закладу.

Для досягнення поставленої мети були розроблені наступні завдання:

- Дослідити предметну область і проаналізувати існуючі підходи до автоматизації онлайн-замовлення їжі.
- Визначити функціональні та нефункціональні вимоги до веб-сайту.
- Спроекувати архітектуру веб-системи, включаючи структуру бази даних (за необхідності), поведінкові моделі та дизайн користувацького інтерфейсу.
- Реалізувати веб-сайт на основі HTML, CSS та Java-Script.
- Провести тестування розробленої системи для виявлення помилок і перевірки відповідності вимогам.

Об'єкт дослідження – процес автоматизації замовлення їжі через веб-сайти.

Предмет дослідження – методи, технології та підходи до розробки веб-додатків для онлайн-замовлення їжі.

Для виконання роботи використовуються такі методи дослідження:

- аналіз літературних джерел та існуючих рішень у сфері автоматизації замовлення їжі; методи системного аналізу для визначення вимог та функціональних характеристик системи;
- методи програмування та веб-розробки для реалізації проекту на базі HTML, CSS та Java Script;
- методи тестування програмного забезпечення для перевірки працездатності веб-сайту.

Теоретичне значення полягає у систематизації знань щодо веб-розробки, аналізі сучасних підходів до створення онлайн-платформ для ресторанного бізнесу та визначенні ключових вимог до таких систем.

Практичне значення полягає у створенні веб-сайту для віртуальної піцерії, який може бути використаний як основа для реальних комерційних проектів у сфері онлайн-замовлення їжі.

Результатом виконання роботи став веб-сайт, що дозволить клієнтам піцерії переглядати меню, оформлювати замовлення та взаємодіяти із закладом, що сприятиме покращенню сервісу та автоматизації процесу обробки замовлень.

Структура роботи: робота складається зі вступу, трьох розділів, висновків, списку використаної літератури та додатків. Перший розділ присвячений аналізу існуючих веб-сайтів піцерій та визначенню їх переваг і недоліків. Другий розділ містить опис методології розробки сайту, включаючи алгоритми та моделі, що використовуються для аналізу даних. Третій розділ присвячений розробці веб-сайту піцерії, тестуванню окремих його об'єктів та оцінці ефективності.

РОЗДІЛ 1

ХАРАКТЕРИСТИКА ТА АНАЛІЗ ГАЛУЗІ ОНЛАЙН-ЗАМОВЛЕННЯ ЇЖІ

1.1 Характеристика галузі онлайн-замовлення їжі та об'єкта дослідження

У сучасних умовах цифрового бізнесу онлайн-замовлення їжі стало невід'ємною частиною ресторанної індустрії. Віртуальна піцерія, яка надає можливість користувачам замовляти страви через веб-сайт, є ефективним рішенням для оптимізації процесу продажу та розширення аудиторії закладу.

Основною метою моделювання предметної області є створення узагальненої схеми взаємодії користувачів із системою та визначення бізнес-логіки веб-сайту. Для цього необхідно ідентифікувати ключові об'єкти, що беруть участь у функціонуванні системи, та їхню роль у процесах. Веб-сайт піцерії слугує інтерфейсом між клієнтом, який бажає здійснити замовлення, і менеджером, який забезпечує його обробку та виконання.

Крім того, важливими складовими є адміністратор сайту, що контролює його функціонування, та база даних, яка містить інформацію про меню, користувачів і замовлення.

Модель предметної області подано ієрархією об'єктно-процесних діаграм мовою моделювання ОРМ (рисунок 1.1):

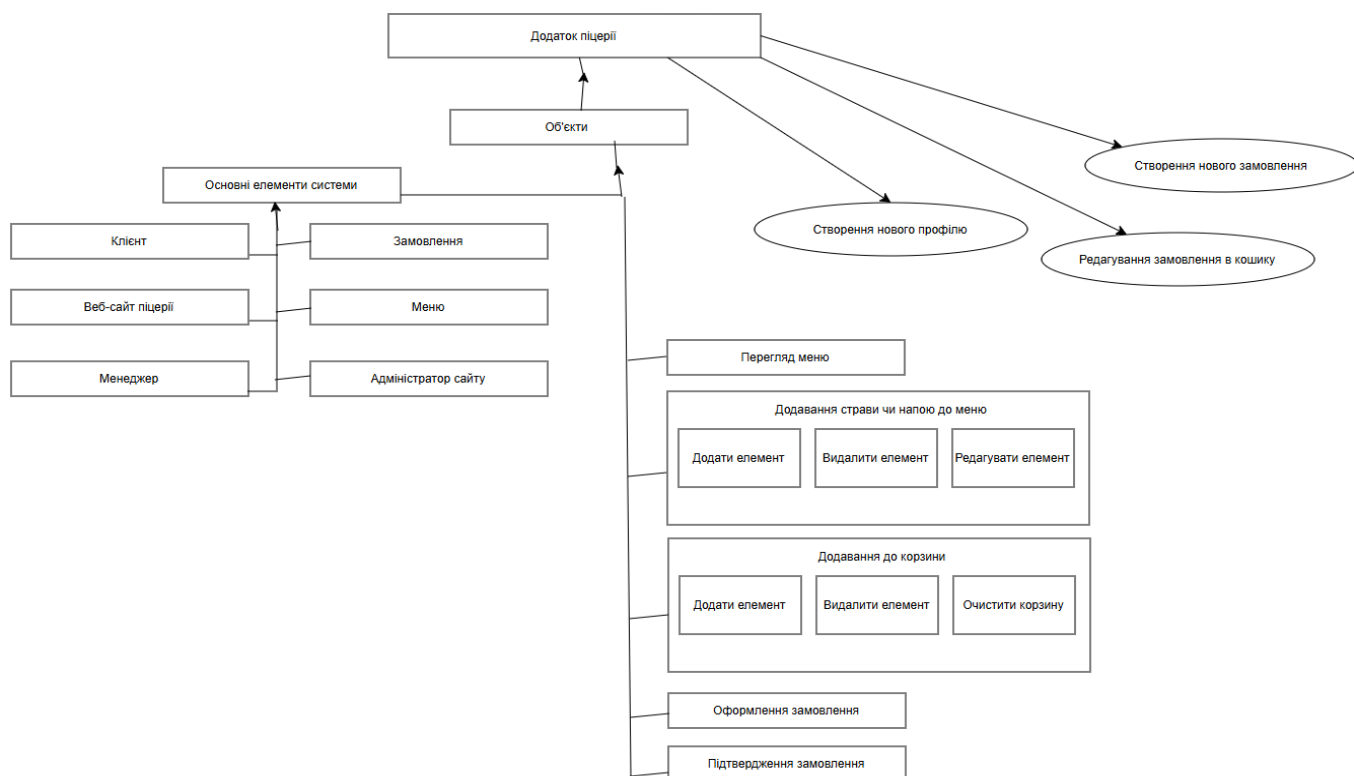


Рисунок 1.1 – Головна діаграма

Джерело: розроблено автором самостійно

На верхньому рівні система складається з кількох основних елементів: клієнта, веб-сайту піцерії, менеджера, адміністратора сайту, замовлення піци чи напою як продукту.

Клієнт є головним користувачем, який взаємодіє із сайтом, переглядаючи меню, додаючи страви до кошика, оформлюючи замовлення та здійснюючи оплату.

Веб-сайт виступає як посередник, який приймає запити від клієнта, обробляє їх і передає відповідним виконавцям.

Менеджер, у свою чергу, отримує інформацію про замовлення, підтверджує її виконання та організовує процес приготування й доставки.

Адміністратор веб-сайту відповідає за його належне функціонування, оновлення меню, зміну цін та управління користувачами.

Діаграми нижчого рівня деталізують головний процес системи шляхом розгортання підпроцесів та пов'язаних з ними об'єктів (рисунок 1.2):

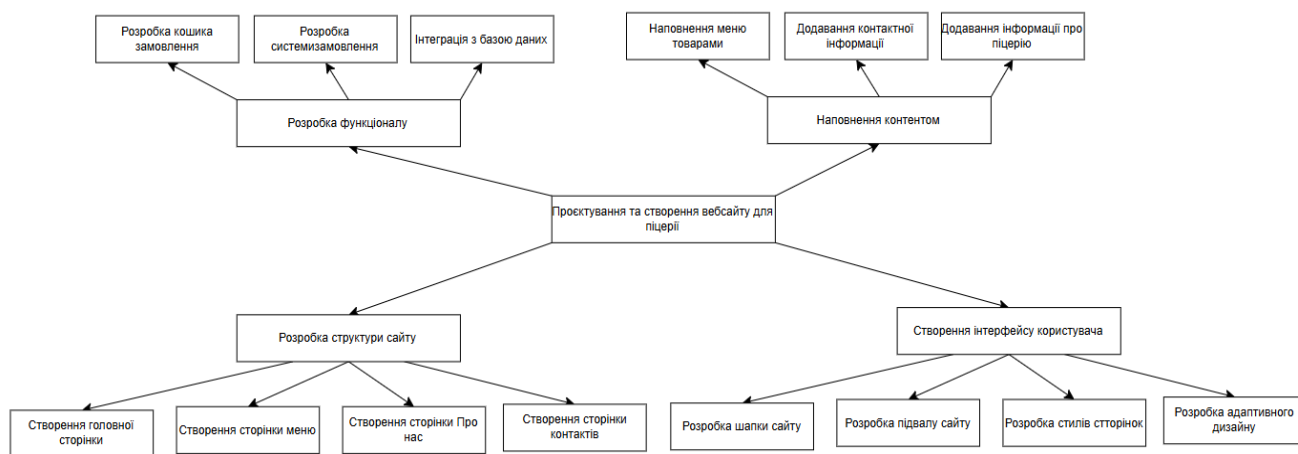


Рисунок 1.2 – Розгортання процесу «Розроблення веб-сайту для віртуальної піцерії»

Джерело: розроблено автором самостійно

З точки погляду процесів, система охоплює кілька ключових механізмів: замовлення піци чи напою, авторизацію користувачів, управління меню та обробку замовлень.

Процес замовлення включає в себе кілька етапів: клієнт переглядає меню, обирає бажану піцу чи напої, додає її до кошика, після чого переходить до оформлення замовлення, вибираючи спосіб оплати та доставки. Важливою складовою є механізм підтвердження замовлення, що забезпечує його коректне збереження у базі даних і надсилання відповідного повідомлення клієнту.

Процес авторизації користувачів дозволяє клієнтам створювати особисті облікові записи, які містять історію їхніх замовлень та контактну інформацію. Він складається з реєстрації нових користувачів, входу в систему, відновлення пароля та виходу з особистого кабінету. Авторизація є важливим компонентом системи, оскільки вона дає змогу персоналізувати досвід користувача та підвищити зручність використання веб-сайту.

Окрему роль відіграє управління меню, яке забезпечується адміністратором сайту. Меню має містити актуальний перелік страв, їхній склад,

зображення та ціни. У процесі управління меню адміністратор додає нові позиції, редагує характеристики наявних страв, видаляє нерелевантні позиції та оновлює вартість продукції. Для оптимізації роботи веб-сайту важливо передбачити можливість управління інгредієнтами та формування спеціальних пропозицій чи знижок.

Ще одним важливим процесом є обробка замовлень. Після підтвердження клієнтом система має передавати інформацію про замовлення менеджеру, який його перевіряє, готує страву та організовує доставку. Важливою частиною цього процесу є можливість відстеження статусу замовлення клієнтом у режимі реального часу, що підвищує зручність та довіру до сервісу.

Для якісного моделювання предметної області використовуються різні рівні деталізації. На системному рівні визначається загальна структура сайту та його функціональні компоненти. Функціональний рівень деталізує ключові процеси, такі як авторизація, управління меню та замовлення. Операційний рівень зосереджується на конкретних підпроцесах і сценаріях взаємодії користувачів із системою.

Щоб забезпечити гнучкість моделі, застосовуються механізми абстрагування, які дозволяють приховувати або деталізувати певні аспекти роботи системи. Це дає змогу створювати багаторівневі діаграми процесів, де на одному рівні можуть бути представлені лише загальні взаємозв'язки, а на іншому – деталізовані сценарії роботи користувачів.

Таким чином, проведене моделювання предметної області веб-сайту піцерії дозволяє отримати чітке уявлення про ключові об'єкти системи, їхні взаємозв'язки та бізнес-логіку функціонування. Запропонована модель дає можливість ефективно проектувати веб-сайт, орієнтуючись на потреби кінцевих користувачів та оптимізацію внутрішніх процесів роботи піцерії.

1.2 Аналіз літературних джерел та практичного досвіду використання віртуальних піцерій і технологій в предметній галузі

Сьогодні веб-сайт для піцерії є не просто ресурсом, а потужним інструментом для залучення клієнтів, збільшення продажів та оптимізації бізнес-процесів. Розвиток інтернет-технологій та зростання популярності онлайн-замовлень їжі зумовили появу великої кількості різноманітних веб-сайтів та інформаційних систем для піцерій чи інших закладів харчування.

Проаналізувавши існуючі рішення, можна виділити такі моменти:

- орієнтація на користувача: сучасні веб-сайти піцерій відрізняються інтуїтивно зрозумілим інтерфейсом, зручною навігацією та адаптивним дизайном для різних пристроїв (комп'ютери, планшети, смартфони). Особлива увага приділяється візуалізації страв за допомогою якісних фотографій та відео, а також наданню детальної інформації про склад, ціну та час приготування;
- інтерактивність та персоналізація: веб-сайти часто включають інтерактивні елементи, такі як онлайн-конструктор піци(де можна самостійно скласти собі піцу із своїх улюблених інгредієнтів), додавання коментарів до замовлення, а також системи персональних рекомендацій на основі попередніх замовлень та вподобань користувачів;
- інтеграція з іншими системами: багато веб-сайтів інтегровані з платіжними системами для онлайн-оплати, службами доставки для відстеження замовлень, CRM-системами для управління клієнтською базою та системами обліку для автоматизації складського обліку та фінансової звітності;
- використання сучасних frontend-технологій: для створення швидких, інтерактивних та зручних веб-сайтів широко використовуються сучасні

frontend-фреймворки та бібліотеки, такі як React, Angular, Vue.js, а також технології HTML, CSS та JavaScript.

Для глибокого розуміння сучасного стану розробки веб-сайтів для піцерій необхідно проаналізувати відповідні ресурси. Для порівняння існуючих інформаційних систем, було взято 3 існуючих веб-сайти з доставки піци: <https://pizzaday.ua/>, <https://dominos.ua/> та <https://terramare.vn.ua/>. Всі вони пропонують схожий функціонал та асортимент. В таблиці 1.1 можна наглядно переглянути мою власну оцінку елементів функціоналу кожного із сервісів.

Таблиця 1.1 – Порівняльний аналіз елементів функціоналу веб-сайтів

| Критерій | Pizza Day
(pizzaday.ua) | Dominos
(dominos.ua) | TerraMare
(terramare.vn.ua) |
|-----------------------------------|----------------------------|-------------------------|--------------------------------|
| Зручність користування | 9 | 7 | 7 |
| Інтуїтивність навігації | 8 | 9 | 7 |
| Простота замовлення | 9 | 9 | 6 |
| Швидкість завантаження сторінок | 8 | 10 | 8 |
| Адаптивність | 10 | 10 | 9 |
| Чіткість інформації | 8 | 9 | 8 |
| Зручність кошика | 6 | 9 | 9 |
| Наявність пошуку та фільтрів | 1 | 8 | 8 |
| Дизайн та візуальне оформлення | 8 | 9 | 7 |
| Естетична привабливість | 8 | 9 | 6 |
| Якість зображень | 9 | 10 | 8 |
| Кольорова схема та шрифти | 7 | 9 | 7 |
| Брендування | 8 | 10 | 7 |
| Використання анімацій | 5 | 9 | 5 |
| Функціональність | 7 | 9 | 8 |
| Повне меню | 7 | 10 | 7 |
| Онлайн-оплата | 6 | 10 | 7 |
| Відстеження замовлення | 9 | 8 | 7 |
| Система реєстрації/авторизації | 6 | 9 | 8 |
| Розділ "Акції" / "Спецпропозиції" | 8 | 9 | 6 |
| Зворотний зв'язок | 8 | 9 | 7 |
| Інтеграція з соціальними мережами | 8 | 9 | 8 |
| Геолокація/вибір адреси доставки | 8 | 10 | 9 |
| Наявність блогу/новин | 8 | 10 | 8 |
| Email-розсилка | 2 | 8 | 8 |
| Відгуки та рейтинги | 5 | 9 | 8 |
| Інформація про переваги піцерії | 7 | 9 | 8 |
| Використані технології | 8 | 9 | 8 |
| Продуктивність | 7 | 9 | 8 |
| Мови | 2 | 8 | 3 |
| Контактна інформація | 9 | 10 | 9 |
| Онлайн-чат | 9 | 8 | 7 |

Джерело: розроблено автором самостійно

Проаналізувавши таблицю, яка в мене вийшла, можна зробити висновки, що майже всі сайти мають схожу оцінку по 10-ти бальній шкалі, лише дещо

відрізняються один від одного, але більшість цих моментів це суто моя думка, тому вона може відрізнятись, проте, такі пункти як відсутність інших мов наявність фільтрів для зручного пошуку позицій в меню значно погіршує загальне враження та зручність користування тим чи іншим сайтом.

Таким чином, я змогла сформулювати мету даного дослідження: розробка веб-сайту для віртуальної піцерії, що забезпечить зручний інтерфейс для перегляду меню, оформлення замовлення та взаємодії з адміністрацією закладу.

Для досягнення поставленої мети були розроблені наступні завдання:

- Дослідити предметну область і проаналізувати існуючі підходи до автоматизації онлайн-замовлення їжі.
- Визначити функціональні та нефункціональні вимоги до веб-сайту.
- Спроекувати архітектуру веб-системи, включаючи структуру бази даних (за необхідності), поведінкові моделі та дизайн користувацького інтерфейсу.
- Реалізувати веб-сайт на основі HTML, CSS та Java Script.
- Провести тестування розробленої системи для виявлення помилок і перевірки відповідності вимогам.

РОЗДІЛ 2

РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЕБ-САЙТУ ПІЦЕРІЇ

2.1 Аналіз і специфікація вимог до інформаційної системи веб-сайту піцерії

Розробка веб-сайту для піцерії ґрунтується на аналізі бізнес-вимог та бізнес-цілей як кінцевих користувачів, так і власників бізнесу. Основною метою створення цієї системи є забезпечення зручного, швидкого та ефективного способу замовлення піци, що задовольняє потреби клієнтів і водночас сприяє розвитку бізнесу.

Нижче представлена діаграма, що відображає бізнес-вимоги та бізнес-цілі кожного учасника, який взаємодіє із системою (рисунок 2.1). Основними зацікавленими сторонами є розробник та кінцевий користувач.

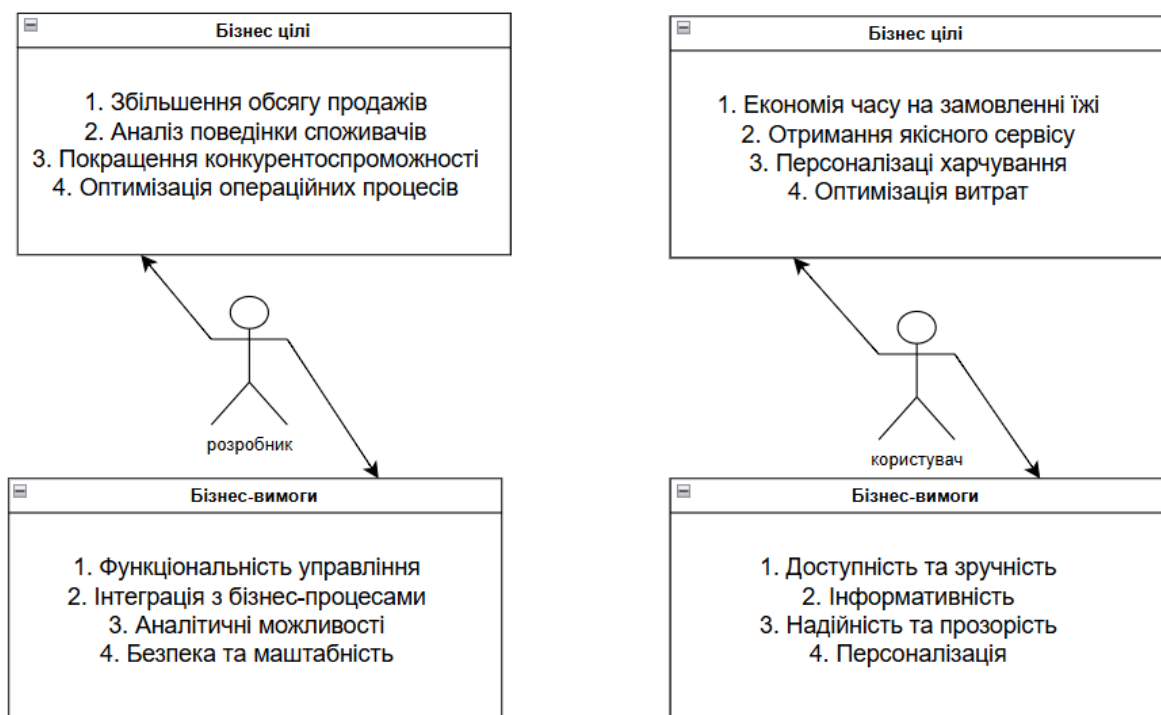


Рисунок 2.1 – Діаграма визначення бізнес-цілей та бізнес-вимог

Джерело: розроблено автором самостійно

З точки погляду користувачів, головними бізнес-цілями є економія часу, отримання якісного сервісу, персоналізація харчування та оптимізація витрат. Сучасний ритм життя вимагає швидкого доступу до меню та простого оформлення замовлення, без необхідності витратити час очікування на телефонній лінії чи повторне введення тих самих даних. Клієнти також прагнуть отримувати якісний сервіс, що включає точну інформацію про страви, гарантію виконання замовлення згідно з усіма зазначеними параметрами та своєчасну доставку. Додатково, важливим аспектом є персоналізація замовлення – можливість обирати інгредієнти відповідно до власних смакових уподобань, доступ до інформації про алергени та калорійність, а також створення індивідуальних комбінацій страв. Не менш важливим є контроль витрат, що реалізується через доступ до акційних пропозицій, можливість порівняння цін та чітке відображення загальної вартості замовлення.

Водночас для власників бізнесу та розробників веб-сайту головною метою є збільшення обсягу продажів, оптимізація операційних процесів, покращення конкурентоспроможності та аналіз поведінки споживачів. Впровадження онлайн-платформи дозволяє значно розширити клієнтську базу, збільшити середній чек та підвищити частоту повторних замовлень завдяки зручності користування сайтом і програмою лояльності. Автоматизація прийому замовлень не тільки зменшує навантаження на персонал, а й скорочує кількість помилок під час обробки замовлень, що своєю чергою покращує загальний рівень обслуговування. Наявність сучасного каналу продажів позитивно впливає на імідж піцерії, підвищує упізнаваність бренду та дозволяє компанії ефективно конкурувати на ринку. Окрім того, система повинна містити аналітичні інструменти, які дадуть змогу збирати дані про вподобання клієнтів, оцінювати ефективність маркетингових кампаній і прогнозувати попит на певні категорії страв, що сприятиме кращому плануванню закупівель і оптимізації витрат.

Щодо конкретних вимог до системи, вони також поділяються на користувацькі та бізнесові. Веб-сайт повинен бути доступним у будь-який час доби, мати адаптивний дизайн, що зручно та гарно відображається на різних

пристроях, і дозволяти оформлення замовлення за мінімальну кількість кроків. Важливою є підтримка різних методів оплати, щоб надати клієнтам максимальну гнучкість. Система повинна містити детальний опис кожної страви з фотографіями, надавати інформацію про склад, алергени та калорійність, а також дозволяти переглядати відгуки інших користувачів. Додатково, веб-сайт має забезпечувати прозорість процесу замовлення, включаючи автоматичне підтвердження прийняття заявки, можливість відстеження статусу виконання, збереження історії попередніх замовлень та зручний зворотний зв'язок з менеджером. Персоналізація є ще одним ключовим аспектом, тому система повинна дозволяти збереження особистих даних та адрес доставки, створення «улюблених» комбінацій страв, отримання індивідуальних пропозицій та налаштування сповіщень про нові акції.

Для власників піцерії веб-сайт має стати не лише зручним інструментом продажу, а й ефективною платформою управління бізнесом. Він повинен містити можливості швидкого оновлення меню, гнучкого налаштування цін та акцій, управління доступністю товарів та контролю статусів замовлень. Важливим є інтеграція системи з іншими бізнес-процесами, зокрема з програмами обліку товарів, фінансовими сервісами, логістичними службами та платіжними платформами. Крім того, система повинна мати аналітичні функції для оцінки ефективності акційних пропозицій, формування звітів про продажі, аналізу поведінки користувачів і сегментації клієнтської бази. Особливу увагу необхідно приділити безпеці даних – система повинна захищати особисту інформацію клієнтів, бути стійкою до навантажень у пікові години та передбачати можливість масштабування та розширення функціоналу в майбутньому.

З огляду на всі зазначені бізнес-вимоги, у подальшому вони будуть представлені у вигляді відповідних моделей та діаграм, що дозволять деталізовано відобразити їхню структуру та взаємозв'язки. Впровадження цих вимог у розробку системи допоможе створити ефективний, зручний та конкурентоспроможний сервіс для користувачів і водночас сприятиме розвитку бізнесу піцерії.

Функціональні вимоги визначають ключові можливості та функції, які має забезпечувати додаток для управління базою знань піцерії, щоб відповідати потребам користувачів і досягати бізнес-цілей проекту. Вони гарантують, що інформаційна система матиме необхідний інструментарій для ефективного керування піцерією. Візуальне подання функціональних вимог представлено на діаграмі (рисунок 2.2), а їхню детальну специфікацію, створену за допомогою менеджера специфікацій (Specification Manager) Enterprise Architect, можна переглянути на рисунках 2.3–2.5.

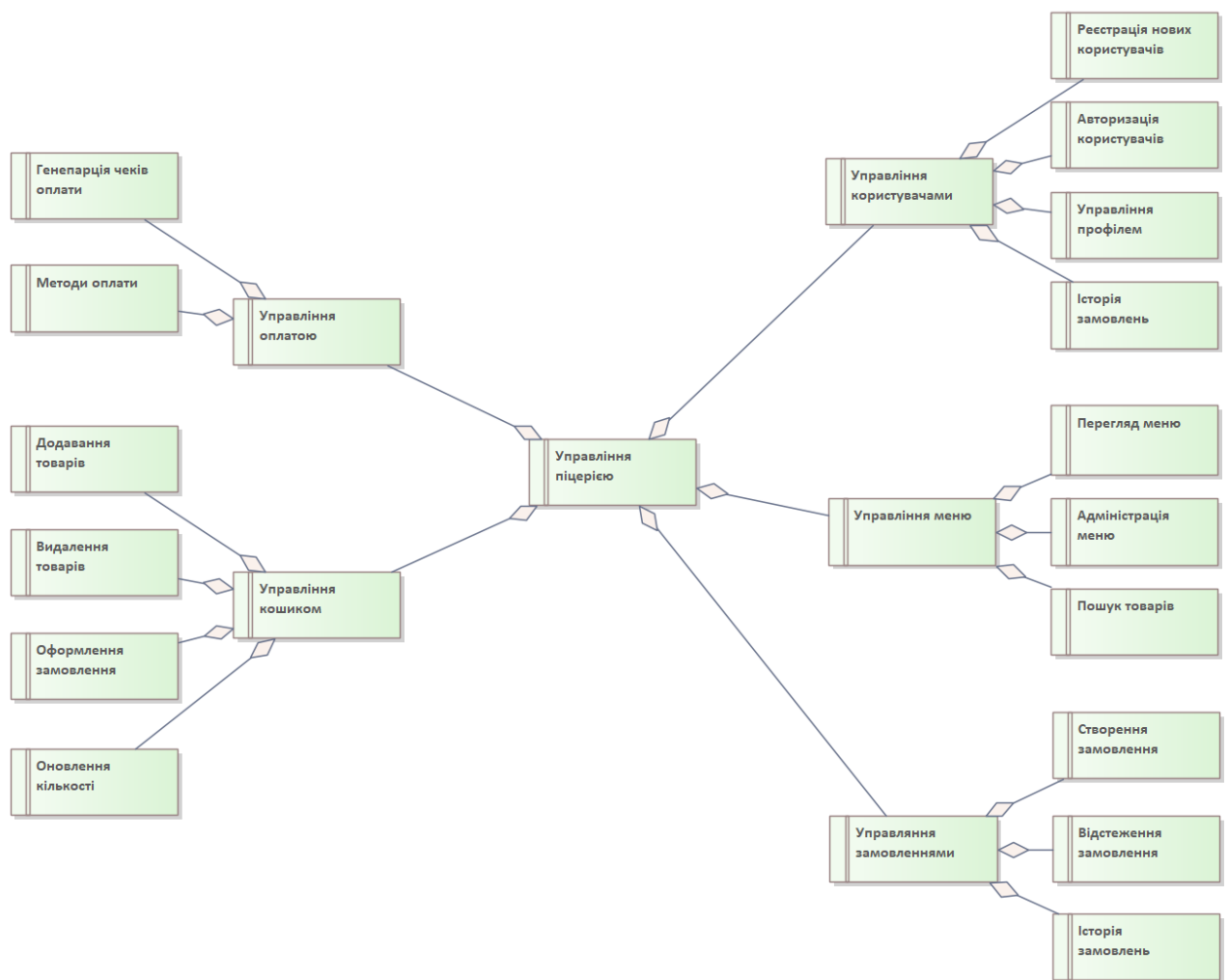


Рисунок 2.2 – Діаграма функціональних вимог

Джерело: розроблено автором самостійно

| Start Page *Функціональні вимоги Specification Manager x | | | | |
|--|------------|-------------|------------|----------|
| Model Specification | | | | |
| Item | Stereotype | Status | Difficulty | Priority |
| ☒ Авторизація користувачів | Functional | Implemented | Low | High |
| ☒ Адміністрація меню | Functional | Proposed | Medium | Medium |
| ☒ Видалення товарів | Functional | Implemented | Low | High |
| ☒ Відстеження замовлення | Functional | Proposed | High | High |
| ☒ Генерация чеків оплати | Functional | Proposed | Medium | High |
| ☒ Додавання товарів | Functional | Implemented | Low | High |
| ☒ Історія замовлень | Functional | Proposed | Medium | Medium |
| ☒ Історія замовлень | Functional | Proposed | Low | Medium |
| ☒ Методи оплати | Functional | Proposed | Medium | High |

Рисунок 2.3 – Специфікації вимог

Джерело: розроблено автором самостійно

| Model Specification | | | | |
|---|------------|-------------|------------|----------|
| Item | Stereotype | Status | Difficulty | Priority |
| ☒ Оновлення кількості | Functional | Proposed | Medium | Medium |
| ☒ Оформлення замовлення | Functional | Implemented | Medium | Medium |
| ☒ Перегляд меню | Functional | Implemented | Low | High |
| ☒ Пошук товарів | Functional | Proposed | Low | Medium |
| ☒ Реєстрація нових користувачів | Functional | Implemented | Low | High |
| ☒ Створення замовлення | Functional | Proposed | High | High |
| ☒ Управління користувачами | Functional | Implemented | Medium | Medium |
| ☒ Управління кошиком | Functional | Implemented | High | High |
| ☒ Управління меню | Functional | Proposed | Medium | Medium |

Рисунок 2.4 – Специфікації вимог

Джерело: розроблено автором самостійно

| Item | Stereotype | Status | Difficulty | Priority |
|--|------------|-------------|------------|----------|
| ☒ Управління оплатою | Functional | Implemented | High | High |
| ☒ Управління піцерією | Functional | Implemented | High | High |
| ☒ Управління профілем | Functional | Proposed | Low | Medium |
| ☒ Управління замовленнями | Functional | Implemented | High | High |
| ☒ Фільтрація товарів за категоріями | Functional | Implemented | Medium | Medium |

Рисунок 2.5 – Специфікації вимог

Джерело: розроблено автором самостійно

На основі представленої діаграми сформовано детальний опис нефункціональних вимог для веб-сайту віртуальної піцерії. Одним із ключових аспектів є продуктивність системи, яка має забезпечувати швидку дію та оптимізацію ресурсів. Веб-сайт повинен завантажуватися не більше ніж за три секунди при стандартному інтернет-з'єднанні, а час відгуку на дії користувача має бути не довшим за одну секунду. Обробка замовлення повинна займати не більше п'яти секунд, а під час пікового навантаження система повинна підтримувати одночасну роботу не менше тисячі користувачів без втрати продуктивності. Для цього необхідно ефективно використовувати серверні ресурси та оптимізувати зображення й базу даних для швидкого пошуку товарів.

Зручність використання також є критичною характеристикою. Навігація по сайту має бути інтуїтивно зрозумілою, а процес оформлення замовлення не повинен перевищувати п'яти кроків. Адаптивний дизайн забезпечить коректне відображення сайту на різних пристроях, а відповідність стандартам доступності гарантує зручність користування для людей із порушеннями зору. Важливою вимогою є можливість масштабування інтерфейсу та використання контрастної кольорової гами.

Щодо надійності, система має працювати без збоїв. У разі аварійного завершення роботи автоматичне відновлення має здійснюватися протягом п'яти хвилин. Щоденне резервне копіювання даних відбуватиметься без впливу на функціонування системи. Також передбачена надійна обробка помилок: користувачі отримуватимуть зрозумілі повідомлення, а всі критичні збої будуть автоматично реєструватися для подальшого аналізу. Важливим аспектом є збереження цілісності даних, що передбачає дотримання принципів ACID та неможливість втрати замовлень навіть у разі збою системи.

Масштабованість передбачає як горизонтальне, так і вертикальне розширення, що дозволить збільшувати потужності без значних змін у коді. Додавання нових серверів і автоматичний розподіл навантаження дозволять підтримувати стабільну роботу під час збільшення кількості користувачів. Крім того, архітектура має бути достатньо гнучкою для розширення функціоналу, інтеграції з новими платіжними системами та сервісами доставки без зниження продуктивності.

Безпека є важливим аспектом, який включає захист персональних даних, безпеку платежів та протидію кібератакам. Всі персональні дані користувачів мають бути зашифровані, доступ до них має бути обмежений лише авторизованому персоналу, а система відповідатиме вимогам GDPR. Платіжні транзакції повинні відбуватися через захищене з'єднання з використанням стандарту PCI DSS, при цьому дані карток не повинні зберігатися на серверах піцерії. Для запобігання шахрайству передбачена багатофакторна автентифікація та обмеження кількості невдалих спроб входу. Додатково система повинна мати захист від поширених веб-загроз, таких як CSRF, XSS, SQL-ін'єкції, а також механізми автоматичного моніторингу підозрілої активності та регулярного оновлення програмного забезпечення.

Усі ці нефункціональні вимоги відображені у вигляді SysML-діаграми (рисунок 2.6). Для кожної вимоги будуть визначені метрики та критерії перевірки, що дозволить об'єктивно оцінити їх виконання під час розробки та тестування системи.

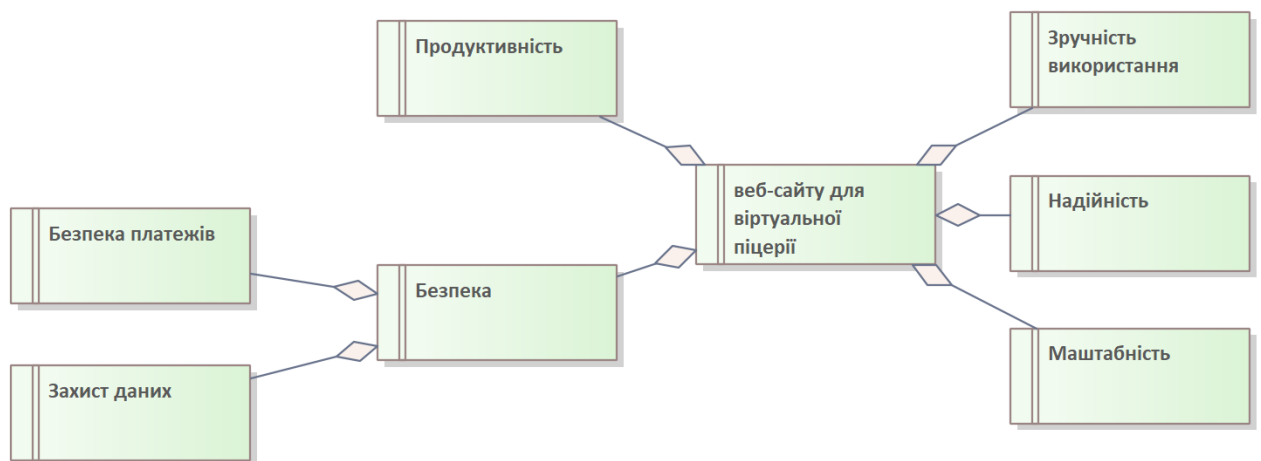


Рисунок 2.6 – Діаграма нефункціональних вимог

Джерело: розроблено автором самостійно

.2 Постановка та алгоритм розв’язання задач

Техніко-економічна сутність задачі полягає у забезпеченні безперебійного процесу прийому замовлень, автоматизації взаємодії з клієнтами та оптимізації внутрішніх процесів піцерії. Потреба розв’язання цієї задачі за допомогою електронно-обчислювальних машин обґрунтовується такими факторами:

- розширення ринку та клієнтської бази: веб-сайт дозволяє охопити значно ширшу аудиторію, за рахунок замовлення піци з дому чи з будь-якого іншого місця перебування клієнта;
- підвищення ефективності: автоматизація прийому замовлень зменшує навантаження на персонал, мінімізує помилки та прискорює обробку кожного замовлення та дозволяє робити відстеження в режимі реального часу;
- покращення користувацького досвіду: зручний інтерфейс приваблює клієнтів та сприяє повторним замовленням;

- зниження витрат: сайт піцерії дозволяє економити на оренді великих торгових приміщень та утриманні великого штату персоналу, проте є і інша сторона, наприклад утримання сайту, що теж є досить;
- збір даних та аналітика: система дозволить збирати дані про переваги клієнтів, обсяги продажів, що є цінним для прийняття бізнес-рішень;
- конкурентоспроможність: в наш час, наявність сайту який рекламує твій бізнес, є однією з найважливіших умов успішного бізнесу в цілому.

Задача розв'язується для управління такими об'єктами:

- клієнти: реєстрація, авторизація, історія замовлень, персональні дані;
- меню піцерії: каталог піц, закусок, напоїв, десертів, їх опис, ціни, наявність;
- замовлення: створення, обробка, відстеження статусу, історія;
- платежі: інтеграція з платіжними системами;
- доставка: адреси, зони доставки, час доставки;
- акції та спеціальні пропозиції: управління промокодами, знижками;
- відгуки та рейтинги: збір та відображення зворотного зв'язку від клієнтів.

Вихідна інформація призначена для інформування клієнтів про стан їхніх замовлень, демонстрації меню та цін, а також для надання адміністрації піцерії даних про продажі, замовлення та клієнтів. Вона використовується для прийняття рішень щодо оптимізації асортименту, маркетингових кампаній та покращення сервісу обслуговування.

Періодичність розв'язання і термін видачі вихідної інформації:

- меню та ціни: формується за запитом користувача при кожному відвідуванні веб-сайту. Термін видачі – до кількох секунд;
- статус замовлення: оновлюється в реальному часі після кожного етапу виконання замовлення (наприклад прийнято, замовлення готується, в дорозі, доставлено). Термін видачі – миттєво після зміни статусу;
- підтвердження замовлення або оплати: видається миттєво після успішного оформлення замовлення (або платежу платежу);

- історія замовлень або профіль користувача: формується за запитом авторизованого користувача. Термін видачі – до кількох секунд;
- звіти для адміністрації: формуються за запитом адміністратора, періодичність може бути щоденною, щотижневою, щомісячною. Термін видачі – від кількох секунд до кількох хвилин залежно від обсягу даних.

Автоматизоване розв'язання задачі припиняється за таких умов:

- технічний збій: помилки в роботі сервера, бази даних, мережевого з'єднання, що унеможливають функціонування сайту;
- планове технічне обслуговування сайту: проведення планових робіт з оновлення системи або усунення несправностей;
- відсутність підключення до мережі інтернету: з боку користувача або сервера;
- припинення функціонування піцерії: закриття бізнесу по будь-якій причині.

Задача розробки фронтенду веб-сайту віртуальної піцерії тісно пов'язана з такими задачами:

- розробка backend-системи: frontend взаємодіє з backend для отримання даних про меню, оформлення замовлень, обробку платежів, управління користувачами;
- інтеграція з платіжними системами: необхідна для здійснення оплати онлайн;
- інтеграція з картографічними сервісами: для відображення зон доставки та відстеження кур'єрів;
- система управління замовленнями: frontend передає дані про замовлення в цю систему.

Розподіл дій між персоналом і технічними засобами за різних ситуацій розв'язання задачі представлені в таблиці 2.1:

Таблиця 2.1. Розподіл дій між персоналом і технічними засобами

| Ситуація | Дії персоналу (людини) | Дії технічних засобів (веб-сайт, сервер, БД) |
|----------------------------|---|--|
| Користувач обирає піцу | Перегляд меню, вибір товарів, додавання до кошика | Відображення меню, фільтрація, оновлення кошика |
| Оформлення замовлення | Введення контактних даних, адреси, способу оплати | Валідація введених даних, передача замовлення на бекенд |
| Здійснення оплати | Вибір способу оплати, введення платіжних реквізитів | Передача даних до платіжної системи, отримання підтвердження |
| Відстеження замовлення | Перегляд статусу замовлення | Відображення актуального статусу замовлення |
| Адміністратор оновлює меню | Введення нових страв/змін через адмін-панель | Оновлення даних в базі даних, відображення на сайті |
| Технічний збій | Повідомлення користувачу про помилку, аналіз логів | Видача повідомлення про помилку, призупинення роботи |

Джерело: розроблено автором самостійно

Взаємозв'язки між компонентами системи зображені на інформаційній моделі задачі, що зображена на рисунку 2.7:

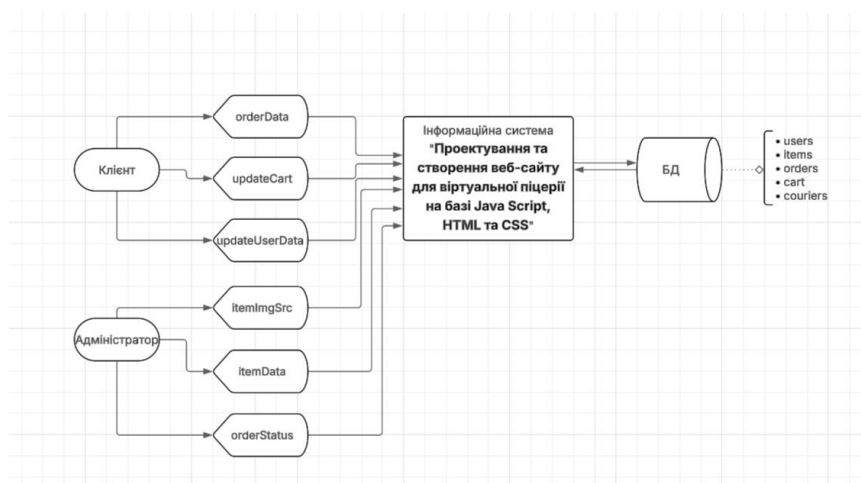


Рисунок 2.7 - Інформаційна модель задачі

Джерело: розроблено автором самостійно

Вхідна інформація призначена для ініціалізації процесів замовлення, автентифікації користувачів, оновлення інформації про клієнтів та забезпечення актуальності даних системи. Одержується переважно через форми веб-сайту, а також від зовнішніх систем.

Таблиця 2.2 – Перелік і опис вхідних повідомлень

| з/п | Назва вихідного повідомлення | Ідентифікатор | Форма подання і вимоги до неї | Періодичність видання | Термін видання і допустимий час затримки | Користувачі інформації |
|-----|--|----------------|--|--|--|------------------------|
| 1 | Інформація про вибрані позиції меню, адреса доставки, спосіб оплати. | orderData | JSON-запит через інтерфейс сайт. Має містити: ID користувача, ID товарів, кількість, адресу, спосіб оплати | За кожне нове замовлення | До 2 секунд | Клієнт |
| 2 | Оновлення кошика: додавання/видалення позицій | updateCart | JSON-запит. Необхідно: унікальний ID кошика, список змін | У режимі реального часу при кожній зміні | До 1 секунд | Клієнт |
| 3 | Зміна контактних даних користувача (телефон, email, пароль тощо) | updateUserData | Форма HTML. Перевірка формату email, пароля, номеру телефону | При редагуванні профілю | До 1 секунд | Клієнт |
| 4 | Завантаження зображень нових позицій меню. | itemImgSrc | Завантаження зображення (JPEG/PNG). Не більше 2MB, розширення JPG/PNG, 1:1 формат | При додаванні/редагуванні товару | До 3 секунд | Адміністратор |
| 5 | Додавання або редагування назви, опису, ціни, категорії. | itemData | Форма HTML/JSON. Назва, опис, ціна (у гривнях), категорія, ID товару | За потреби оновлення меню | До 2 секунд | Адміністратор |
| 6 | Зміна статусу замовлення (прийнято, готується, доставлено) | orderStatus | JSON-запит/API. ID замовлення, новий статус, час зміни | Після кожного оновлення статусу | До 3 секунд | Адміністратор |

Джерело: розроблено автором самостійно

Перелік та опис вхідних повідомлень:

orderData - інформація про вибрані позиції меню, адреса доставки, спосіб оплати.

Має містити: ID користувача, ID товарів, кількість, адресу, спосіб оплати.

- оновлення кошика: додавання/видалення позицій. Необхідно: унікальний ID кошика, список змін.

- зміна контактних даних користувача (телефон, email, пароль тощо). Перевірка формату email, пароля, номеру телефону.
- завантаження зображень нових позицій меню. Не більше 2MB, розширення JPG/PNG, 1:1 формат.
- додавання або редагування назви, опису, ціни, категорії. Назва, опис, ціна (у гривнях), категорія, ID товару.
- зміна статусу замовлення (прийнято, готується, доставлено). ID замовлення, новий статус, час зміни.

Вихідна інформація призначена для інформування клієнтів про стан їхніх замовлень, демонстрації меню та цін, а також для надання адміністрації піцерії даних про продажі, замовлення та клієнтську активність. Вона використовується для прийняття рішень щодо оптимізації асортименту, маркетингових кампаній та покращення сервісу.

Таблиця 2.3 – Перелік і опис вихідних повідомлень

| з/п | Назва вихідного повідомлення | Ідентифікатор | Форма подання і вимоги до неї | Періодичність видання | Термін видання і допустимий час затримки | Користувачі інформації |
|-----|--|-------------------|--|--------------------------------------|--|------------------------|
| 1 | Дані про страви з бази даних (назва, ціна, фото, опис) | menuData | HTML + JSON. 3 групування за категоріям и, опис, ціна, фото | При кожному запиті головної сторінки | До 3 секунд | Клієнт |
| 2 | Поточний стан кошика клієнта | cartData | JSON. Актуальний кошик клієнта | Після кожної взаємодії з кошиком | До 2 секунд | Клієнт |
| 3 | Повідомлення про успішне оформлення замовлення | orderConfirmation | Вспливаюче повідомлення + email. Містить номер замовлення, час, підсумок | Одразу після підтвердження | До 1 секунд | Клієнт |
| 4 | Поточний статус доставки (наприклад: «в дорозі») | deliveryStatus | Сповіщення + API. Короткий текстовий статус (наприклад: "В дорозі") | Зі зміною статусу замовлення | До 2 секунд | Клієнт |
| 5 | Перелік нових замовлень з деталізацією | ordersList | Таблиця (інтерфейс системи). | Оновлюється автоматично в | До 2 секунд | Адміністратор |

| | | | | | | |
|---|--|-------------|--|---|-------------|---------------|
| | | | ID,
дата/час,
статус,
клієнт | режимі
реального часу | | |
| 6 | Перелік
zareestrovani
x користувачів | clientsList | Таблиця.
Прізвище,
email, дата
реєстрації | Один раз на
день або за
запитом | До 3 секунд | Адміністратор |
| 7 | Дані про всі
позиції меню | itemList | Таблиця.
Назви,
категорії,
статус
активності | За запитом,
при
завантаженні
редактора
меню | До 1 секунд | Адміністратор |

Джерело: розроблено автором самостійно

Перелік і опис вихідних повідомлень:

- дані про страви з бази даних (назва, ціна, фото, опис). З групуванням за категоріями, опис, ціна, фото.
- поточний стан кошика клієнта. Подається у форматі JSON.
- повідомлення про успішне оформлення замовлення. Містить номер замовлення, час, підсумок.
- поточний статус доставки (наприклад: «в дорозі»). Короткий текстовий статус (наприклад: "В дорозі").
- перелік нових замовлень з деталізацією. ID, дата/час, статус, клієнт.
- перелік зареєстрованих користувачів. Прізвище, email, дата реєстрації.
- дані про всі позиції меню. Назви, категорії, статус активності.

Математична модель розрахунку основних показників, що формуються у процесі розв'язання задачі, полягає в обчисленні загальної вартості замовлення.

Для розрахунку загальної вартості замовлення можна використати наступну формулу:

$$C_{total} = \sum_{i=1}^N (P_i \times Q_i) \times (1 - D) + C_{delivery}$$

Де:

– загальна вартість замовлення.

N – кількість позицій у кошику.

– ціна i -ї позиції (піци, закуски, напою).

– кількість i -ї позиції.

D – коефіцієнт знижки (від 0 до 1), застосовується у разі використання промокоду або акції.

– вартість доставки (може бути 0, якщо сума замовлення перевищує певний поріг або якщо це самовивіз).

Загальний алгоритм роботи фронтенду веб-сайту віртуальної піцерії:

Ініціалізація:

- завантаження веб-сторінки;
- запит до бекенд API для отримання актуального меню;
- перевірка наявності даних сесії користувача (авторизація, кошик);
- відображення головної сторінки з меню.

Взаємодія з меню:

- користувач переглядає меню;
- при кліку на піцу/товар: запит для відображення детального опису та опцій;
- користувач обирає кількість, модифікації (якщо є) і додає товар до кошика;
- оновлення корзини та відображення суми.

Оформлення замовлення:

- користувач переходить до кошика;
- при натисканні "Оформити замовлення":
 - запит (якщо авторизований) або форма для введення контактних даних клієнта та адреси доставки;
 - запит та застосування знижки, якщо введено промокод;
 - розрахунок загальної суми;
 - відображення підсумкової інформації;
- вибір способу оплати (дані для оплати);
- клік "Підтвердити замовлення".

Обробка замовлення та оплати:

- формування нового замовлення на основі корзини, контактних даних клієнта, адреси доставки, загальної суми та передача на бекенд (API);
- якщо обрано онлайн-оплату:
 - формування та перенаправлення на платіжний шлюз;
 - очікування підтвердження від платіжної системи через бекенд;
- при успішному оформленні/оплаті:
 - відображення оформленого замовлення та початкового статусу замовлення;
 - збереження замовлення та оновлення профілю на бекенді;

Відстеження замовлення:

- користувач може перейти до розділу "Мої замовлення" або використовувати пряме посилання;
- запит до бекенду для отримання актуального статусу;
- відображення статусу замовлення.

Інші функції:

- реєстрація/авторизація: обробка даних реєстрації/авторизації, створення/відновлення;
- відгуки: прийом відгуку клієнта та передача на бекенд;
- адмін-панель (фронтенд частина): отримання зміни даних меню та зміни статусу замовлення від адміністратора та передача на бекенд.

Нижче наведено алгоритм розв'язання задачі у вигляді схеми (рисунок 2.8):

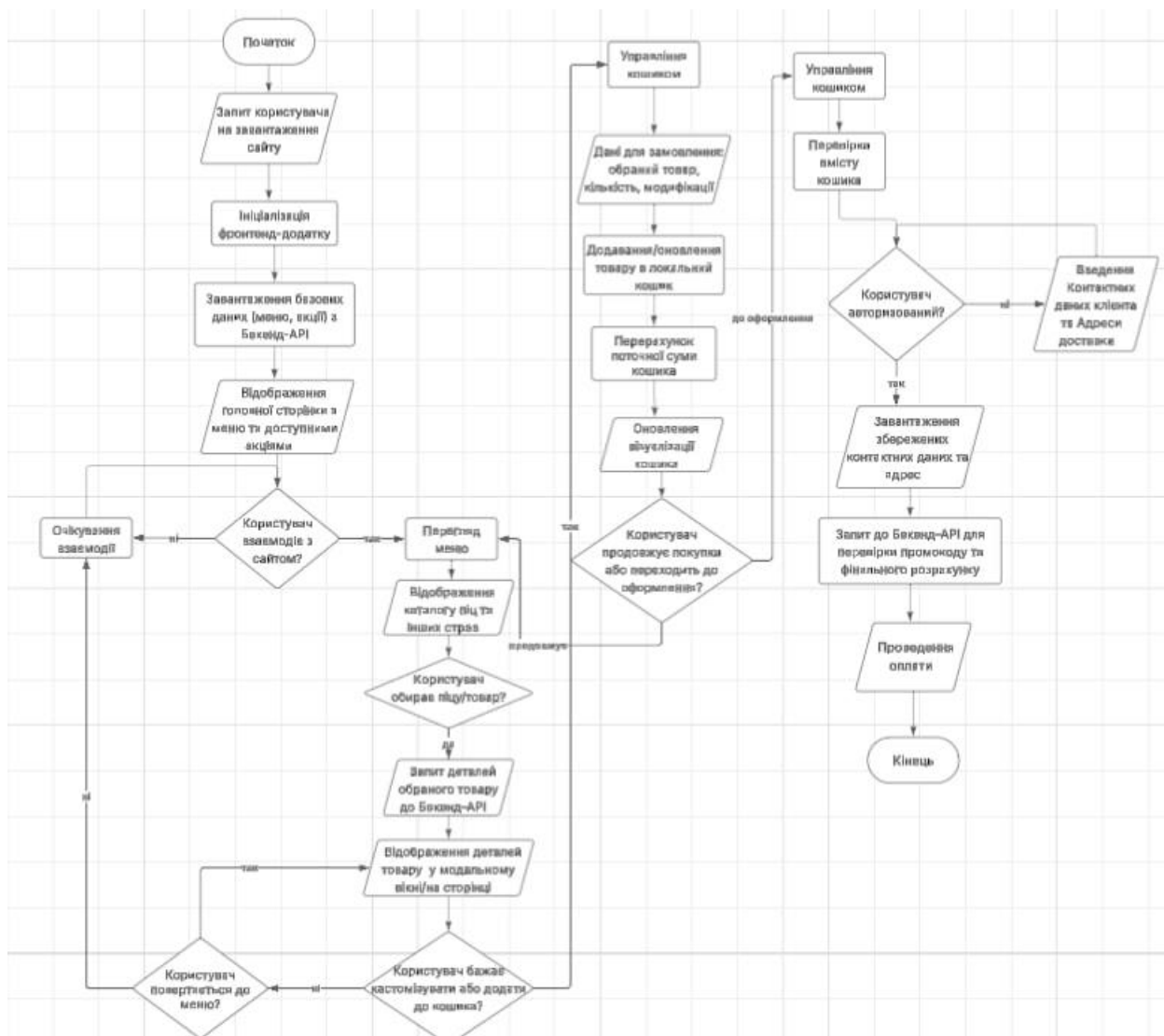


Рисунок 2.8 - Алгоритм розв'язання задачі

Джерело: розроблено автором самостійно

2.3 Моделювання інформаційної системи віртуальної піцерії

Аналізуючи представлену діаграму прецедентів (Use Case Diagram) для веб-сайту віртуальної піцерії «PIZZALICIOUS», можна відзначити комплексне моделювання взаємодії різних акторів із функціональними можливостями системи. Діаграма відображає структуру взаємодії користувачів із системою та

демонструє ієрархію функціональних процесів, що забезпечують її коректну роботу.

У представленій моделі виокремлено три основні категорії акторів (рисунок 2.9): користувач, розробник (адміністратор) та платіжна система, а також додатковий компонент у вигляді бази даних. Користувач є основним відвідувачем сайту, який взаємодіє з меню, кошиком та процесом оплати. Адміністратор наділений розширеними можливостями для управління контентом, замовленнями та аналітикою продажів. Платіжна система виконує роль зовнішнього сервісу, що відповідає за обробку фінансових транзакцій, а база даних забезпечує збереження всієї необхідної інформації про товари, користувачів та замовлення.

Прецеденти, що відображають функціональні можливості, згруповані за основними блоками. Для користувача передбачено управління меню, яке включає можливість перегляду асортименту товарів. Процес аутентифікації реалізований через реєстрацію та авторизацію користувачів. Функціонал роботи з кошиком передбачає додавання та видалення товарів, оформлення замовлення та його відстеження. Крім того, користувачеві надається можливість вибору методу оплати, що є складовою управління фінансовими транзакціями.

Для адміністратора доступні розширені можливості редагування контенту сайту. Управління меню передбачає додавання, редагування та видалення товарів, що дозволяє оперативно оновлювати інформацію про продукцію. Крім того, адміністратор здійснює контроль над акціями та процесом обробки замовлень, а також аналізує продажі для подальшої оптимізації бізнес-процесів.

Платіжна система відповідає за генерацію чеку та обробку оплати, забезпечуючи безпечне проведення транзакцій. Взаємодія цієї системи з веб-сайтом гарантує коректне відображення платежів та їхню інтеграцію з базою даних. Остання, у свою чергу, виконує функцію збереження всієї інформації, забезпечуючи швидкий доступ до даних та їхню цілісність.

Усі зв'язки між акторами та прецедентами представлені у вигляді відповідних ліній, що відображають взаємозалежності між різними складовими

системи. Використання відношень включення (include) дозволяє простежити логіку виконання функціональних процесів та їхню взаємозалежність. Таким чином, діаграма наочно ілюструє, як користувачі взаємодіють із веб-сайтом, які функціональні можливості їм доступні та яким чином відбувається обробка інформації в системі, забезпечуючи її ефективне функціонування.

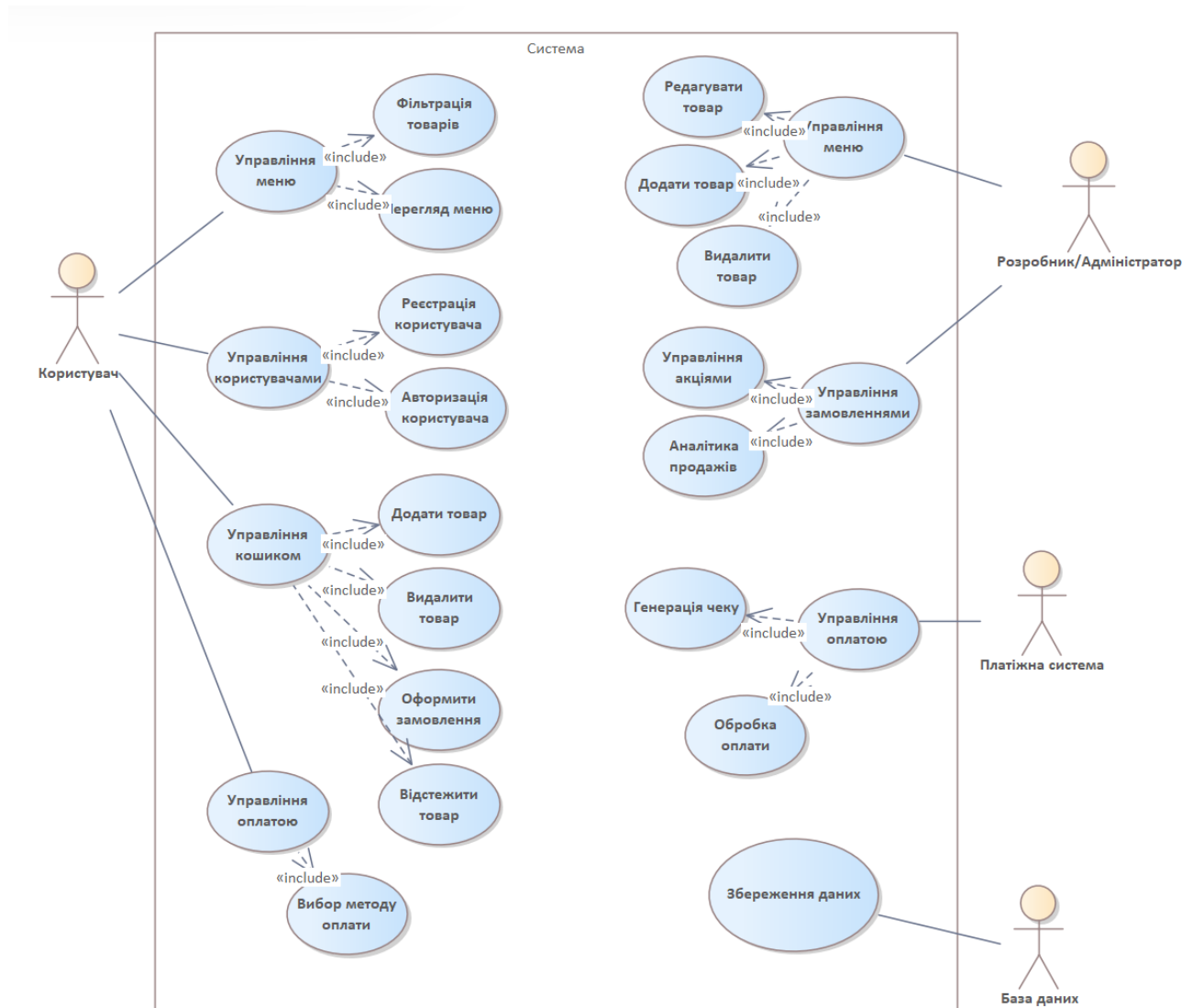


Рисунок 2.9 – Діаграма прецедентів

Джерело: розроблено автором самостійно

Один із найважливіших сценаріїв – процес оформлення замовлення. У типовій послідовності користувач переглядає меню піцерії, додає вибрані товари до кошика та переходить до оформлення замовлення. Далі система відображає форму введення даних доставки, де клієнт вносить необхідну інформацію. Після цього користувачу пропонуються доступні методи оплати, з яких він обирає

відповідний варіант і підтверджує замовлення. У разі вибору онлайн-оплати система перенаправляє клієнта на відповідну сторінку, де здійснюється транзакція. Після успішної оплати система генерує підтвердження оформлення замовлення та надсилає клієнту відповідне повідомлення електронною поштою. У результаті замовлення реєструється в системі та передається на кухню для подальшого приготування. Послідовність цього процесу наведена на рисунку 2.10:

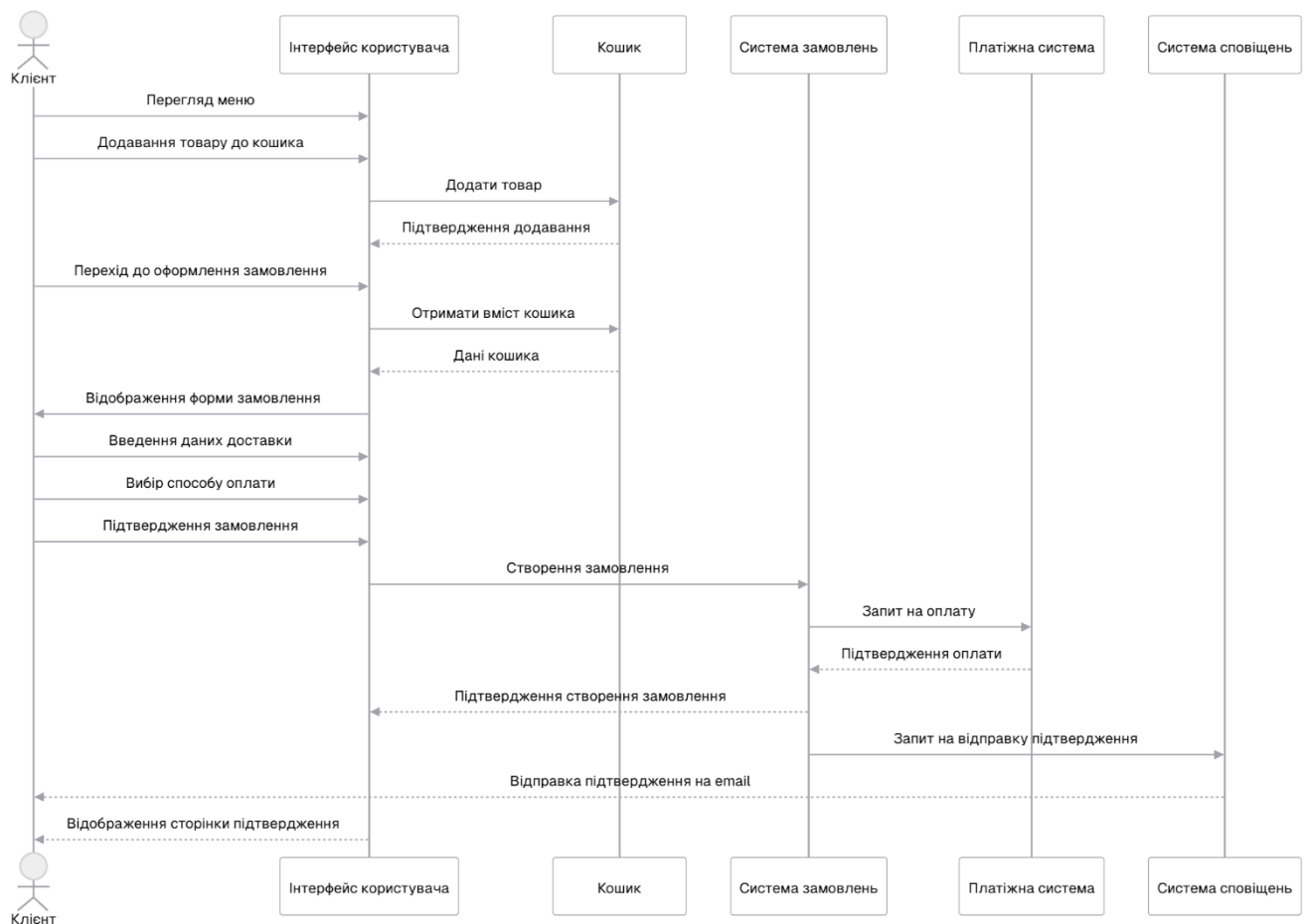


Рисунок 2.10 - Процес оформлення замовлення

Джерело: розроблено автором самостійно

У випадку, коли доставка за вказаною клієнтом адресою недоступна, сценарій набуває альтернативного розвитку. Система повідомляє про неможливість здійснення доставки, після чого користувач може змінити адресу або обрати самовивіз. Після внесення відповідних коригувань процес продовжується відповідно до основного сценарію.

Ще один можливий альтернативний розвиток подій стосується ситуації з помилкою під час оплати. Якщо транзакція не проходить, система інформує користувача про помилку та пропонує повторити спробу або обрати інший спосіб оплати. Після виправлення ситуації процес оформлення замовлення триває, починаючи з етапу оплати.

Окрім користувачів, важливу роль у функціонуванні системи відіграють адміністратори, які мають можливість керувати меню закладу (рисунк 2.11). Сценарій управління меню починається з авторизації адміністратора в системі. Далі він переходить до відповідного розділу, де система відображає перелік товарів. Адміністратор може додавати нові позиції, редагувати існуючі або видаляти застарілі товари. Усі внесені зміни проходять валідацію, після чого зберігаються в системі та стають доступними для перегляду клієнтами на веб-сайті.

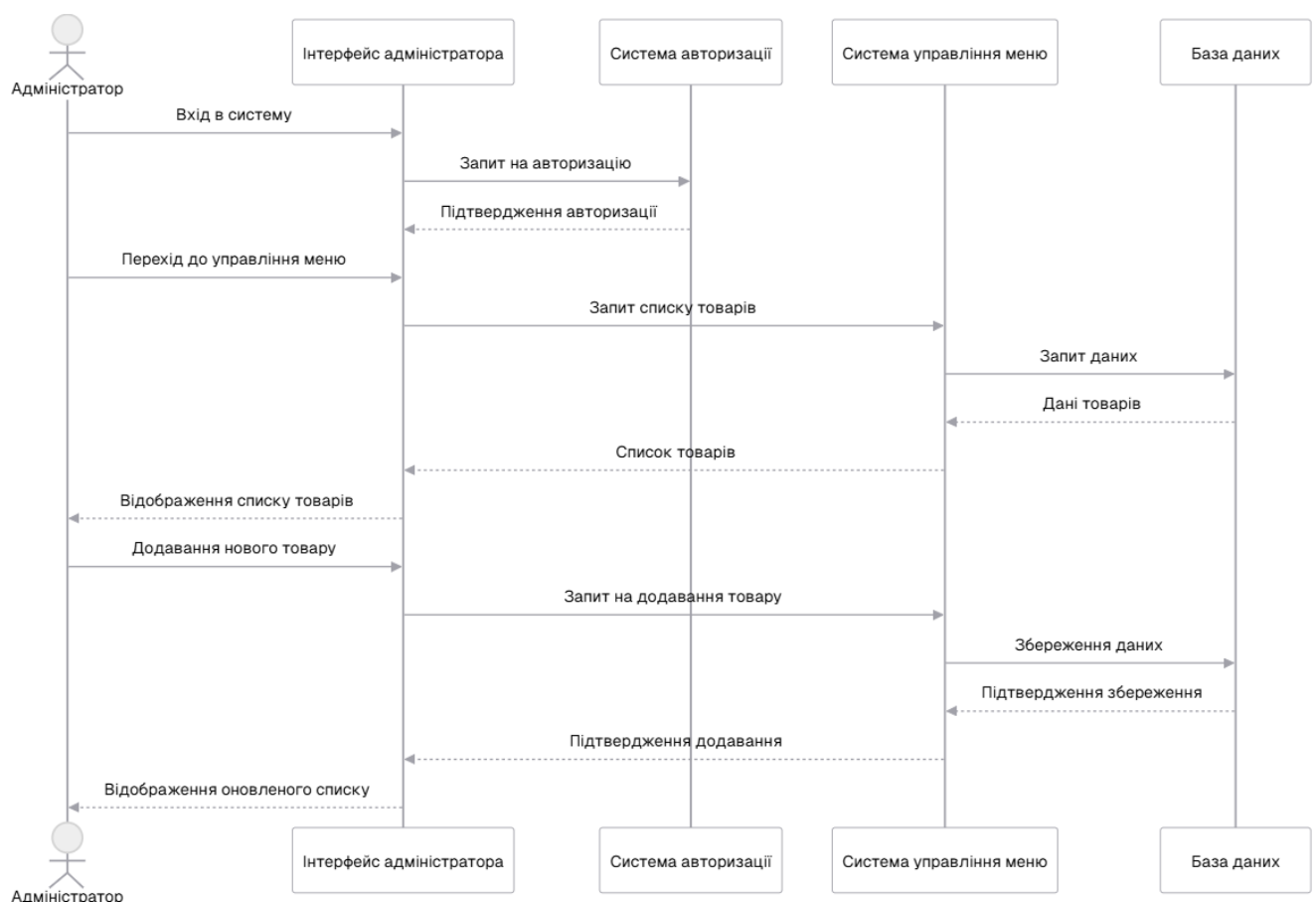


Рисунок 2.11 - Сценарій управління меню

Джерело: розроблено автором самостійно

Таким чином, текстові сценарії варіантів використання деталізують функціональну поведінку системи та допомагають розробникам зрозуміти, як користувачі та адміністратори будуть взаємодіяти з платформою «PIZZALICIOUS». Вони також дозволяють проаналізувати потенційні проблемні ситуації та передбачити їхнє коректне опрацювання, що є важливим етапом у процесі розробки системи.

Процес проектування веб-сайту віртуальної піцерії «PIZZALICIOUS» передбачає побудову структурних моделей, які відображають внутрішню організацію системи, її складові компоненти та взаємозв'язки між ними. Для цього використовуються структурні діаграми SysML, зокрема, діаграми визначення блоків (Block Definition Diagram, BDD) та діаграми внутрішніх блоків (Internal Block Diagram, IBD), а також діаграми класів UML для моделювання статичної структури системи.

На етапі моделювання загальної структури системи створюється діаграма визначення блоків (BDD), яка представляє основні складові веб-сайту у вигляді ієрархії блоків (рисунк 2.12). Система «PIZZALICIOUS» поділяється на такі основні підсистеми: технічне забезпечення, програмне забезпечення та база даних. Технічне забезпечення включає серверну інфраструктуру, клієнтські пристрої, мережеві компоненти та зовнішні інтеграції, такі як платіжні системи. Програмне забезпечення складається з трьох основних рівнів: frontend-частина (HTML, CSS, Java Script), backend-логіка, що забезпечує роботу серверної частини, та база даних, яка зберігає інформацію про користувачів, замовлення, меню тощо.

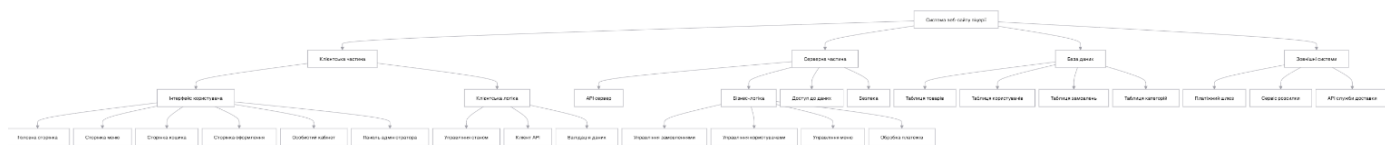


Рисунок 2.12 - Діаграма визначення блоків (BDD)

Джерело: розроблено автором самостійно

Для детального подання зв'язків між внутрішніми компонентами системи створюється діаграма внутрішніх блоків (IBD). Вона відображає, яким чином

окремі частини взаємодіють між собою: клієнтський інтерфейс команікує з backend -частиною через HTTP-запити, backend отримує дані з бази даних через SQL-запити, а також передає їх користувачам. Потоки даних включають запити на отримання інформації про меню, надсилання замовлення, обробку оплати та оновлення статусу доставки. Взаємодія між frontend і сервером відбувається через REST API, що забезпечує чітку логіку обміну даними між компонентами (рисунок 2.13).

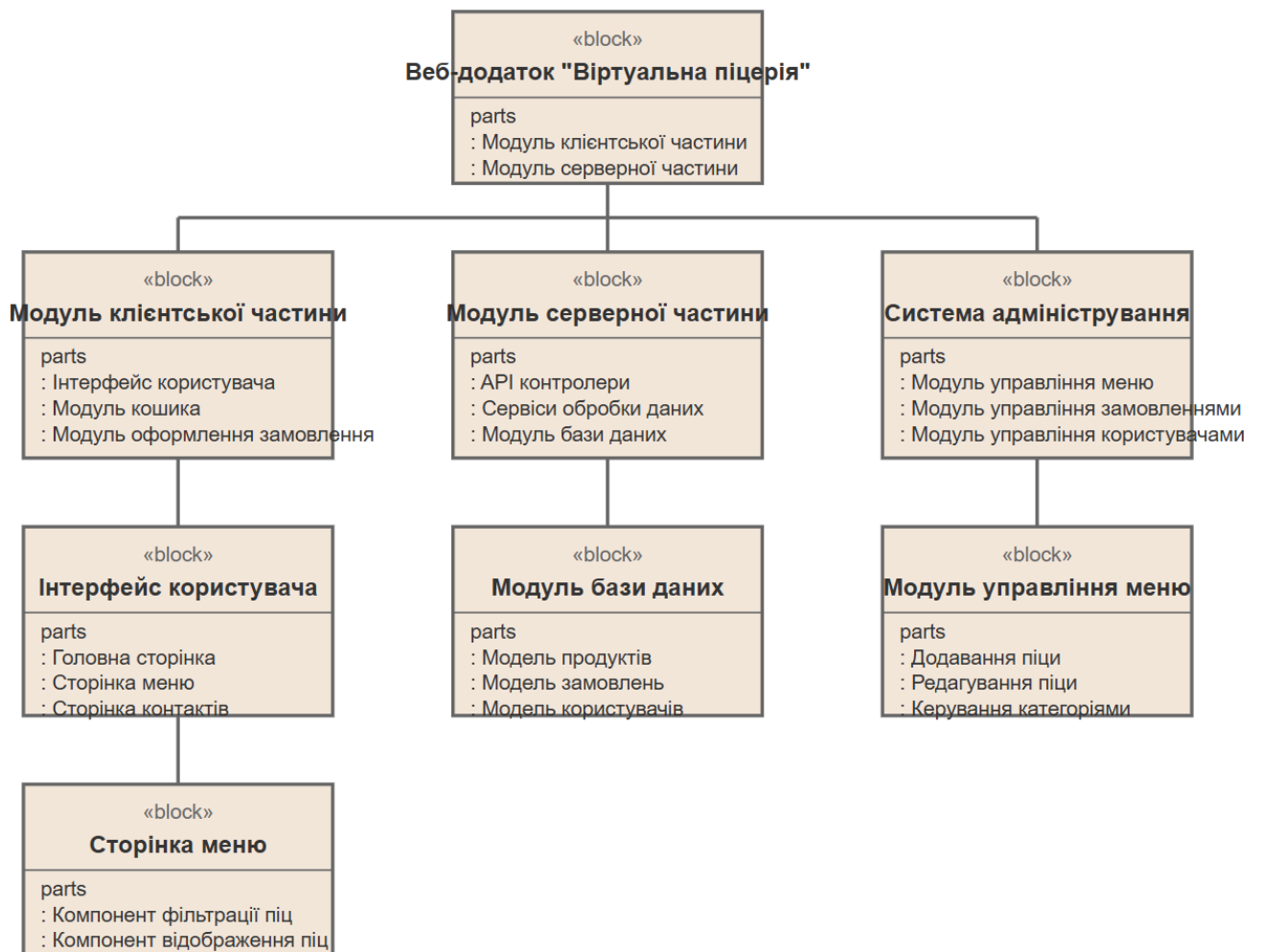


Рисунок 2.13 - Діаграма внутрішніх блоків (IBD)

Джерело: розроблено автором самотійно

Для деталізації програмної реалізації системи використовується UML-діаграма класів, яка відображає основні програмні сутності та зв'язки між ними (рисунок 2.14).

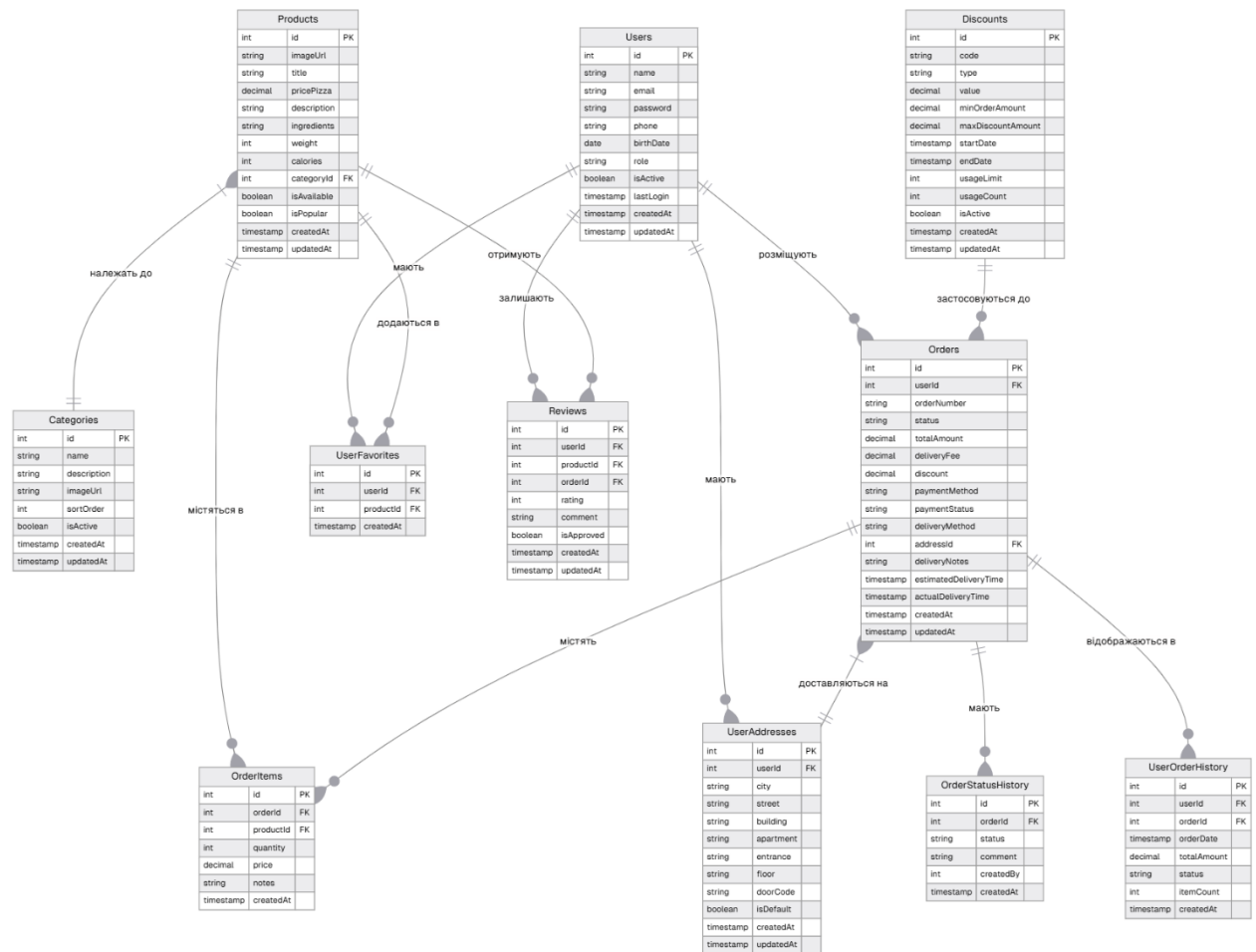


Рисунок 2.14 - UML-діаграма класів

Джерело: розроблено автором самостійно

Центральними елементами моделі є таблиці, які забезпечують збереження необхідних даних та визначають зв'язки між ними.

Основу бази даних складають кілька ключових таблиць. Таблиця Products містить інформацію про товари, зокрема назву, опис, ціну, інгредієнти та інші атрибути, що описують позиції меню. Для класифікації товарів передбачено таблицю Categories, яка дозволяє групувати страви за певними параметрами, спрощуючи навігацію та пошук. Дані користувачів зберігаються у таблиці Users, що включає такі поля, як ім'я, електронна пошта, пароль і номер телефону, забезпечуючи ідентифікацію та аутентифікацію клієнтів.

Для обробки замовлень використовується таблиця Orders, яка містить інформацію про кожне замовлення, включаючи його унікальний номер, статус виконання, загальну суму та інші параметри. Деталізацію кожного замовлення забезпечує таблиця OrderItems, у якій фіксуються всі товари, що входять до

конкретного замовлення, з зазначенням їхньої кількості та ціни. Окремо реалізована таблиця `UserAddresses`, що зберігає адресні дані для доставки, забезпечуючи зручність оформлення замовлень для користувачів, які регулярно використовують певні адреси.

Система також підтримує механізм знижок, який реалізовано через таблицю `Discounts`. Вона містить інформацію про доступні знижки, промокоди та їхні значення, що дозволяє автоматично застосовувати відповідні акції до замовлень користувачів.

У перспективі розвитку функціоналу бази даних передбачено впровадження додаткових таблиць, які розширять можливості платформи. Зокрема, планується реалізація таблиці `Reviews`, що забезпечить можливість залишення відгуків про товари, формуючи систему оцінювання якості продукції. Для персоналізованого досвіду користувачів буде створена таблиця `UserFavorites`, яка дозволить зберігати улюблені товари для швидкого доступу. Таблиця `UserOrderHistory` стане окремим сховищем для історії замовлень кожного користувача, спрощуючи навігацію та повторне замовлення. Крім того, таблиця `OrderStatusHistory` дозволить відстежувати всі зміни статусів замовлення, забезпечуючи прозорість логістичних процесів.

Взаємозв'язки між таблицями базуються на логічних відношеннях, що визначають структуру даних та їхню інтеграцію. Так, товари належать до певних категорій, кожен користувач має можливість створювати замовлення, а кожне замовлення містить один або кілька товарів. Адреси доставки пов'язані з конкретними користувачами, а знижки можуть застосовуватися до замовлень або окремих товарів. Додатково передбачені механізми, які дозволяють відстежувати зміни статусів замовлень та зберігати історичні дані.

Загальна структура бази даних є добре спроектованою, забезпечуючи оптимальну організацію даних для ефективного функціонування системи електронної комерції (рисунки 2.15). Її гнучкість дозволяє легко масштабувати систему, додавати нові функціональні можливості та підтримувати високий рівень продуктивності.

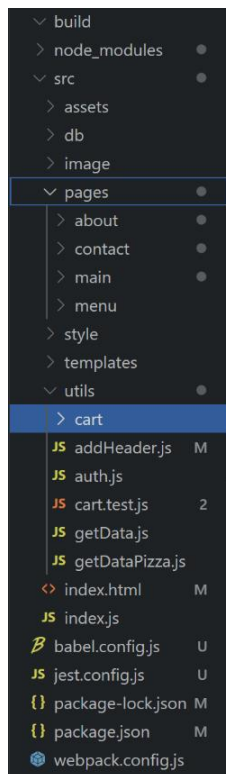


Рисунок 2.15 - Структура файлів, які необхідні для роботи програми

Джерело: розроблено автором самостійно

Таким чином, моделювання структури системи є важливим етапом проектування веб-сайту віртуальної піцерії, оскільки забезпечує чітке уявлення про взаємозв'язки між компонентами та їхню організацію. Використання структурних діаграм SysML дозволило визначити ієрархію блоків, зокрема технічного та програмного забезпечення, а також внутрішні взаємозв'язки між елементами системи. Діаграми класів UML відобразили статичну структуру класів, їхні атрибути, операції та відношення, що сприяло побудові логічної моделі даних. Завдяки цьому вдалося створити структуровану та гнучку архітектуру, яка забезпечує ефективне управління користувачами, замовленнями, товарами та іншими ключовими елементами системи.

Веб-сайт віртуальної піцерії «PIZZALICIOUS» передбачає кілька ключових екранних форм, кожна з яких виконує певну функцію та забезпечує зручну взаємодію користувача із системою.

Головна сторінка є стартовою точкою для користувача, на якій представлено інформацію про піцерію, акційні пропозиції та найпопулярніші страви. Візуально вона містить великий банер із вигідними пропозиціями, блок

із найпопулярнішими позиціями меню, коротку інформацію про заклад та кнопку швидкого замовлення для спрощеного доступу до процесу купівлі.

Сторінка меню забезпечує перегляд усього асортименту піцерії. Інтерфейс складається з категорій товарів, карток страв, що містять зображення, ціну та кнопку додавання до кошика.

Сторінка кошика відображає всі обрані користувачем товари та дозволяє керувати їхньою кількістю, видаляти непотрібні позиції та переглядати загальну вартість замовлення. Для зручності передбачено автоматичний перерахунок підсумкової ціни при зміні кількості товарів. Ключовим елементом є кнопка переходу до оформлення замовлення, що дозволяє розпочати наступний етап взаємодії з сайтом.

Сторінка оформлення замовлення призначена для збору необхідної інформації щодо доставки та оплати. Вона містить форму введення адреси, вибір способу оплати, підсумок замовлення із зазначенням його загальної вартості та кнопку підтвердження. Інтерфейс розроблений таким чином, щоб забезпечити швидке заповнення форми та мінімізувати ймовірність помилок.

Особистий кабінет дозволяє користувачеві керувати своїми персональними даними та переглядати історію замовлень. У цьому розділі передбачена можливість редагування профілю, збереження кількох адрес доставки, перегляду попередніх замовлень із їхнім статусом, а також налаштувань сповіщень про нові акції або оновлення в меню.

Окремий функціональний блок становить панель адміністратора, яка забезпечує керування контентом сайту та обробку замовлень. Інтерфейс адміністративної частини містить меню управління, що дозволяє додавати, редагувати та видаляти товари з меню, змінювати акційні пропозиції та контролювати статуси замовлень. Додатково реалізовані аналітичні інструменти, що відображають статистику продажів та ефективність маркетингових кампаній у вигляді таблиць та діаграм. Візуалізація взаємозв'язків у системі наведена на рисунку 2.16 що ілюструє архітектуру адміністративної панелі та способи взаємодії її компонентів.

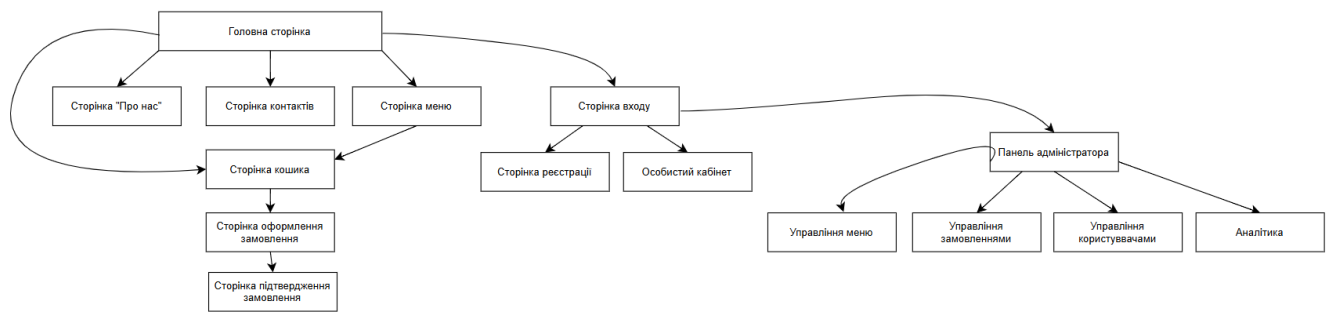


Рисунок 2.16 - Моделювання користувацького інтерфейсу

Джерело: розроблено автором самостійно

Організаційне забезпечення визначає структуру процесів, що забезпечують функціонування інформаційної системи – веб-сайту віртуальної піцерії. Для опису станів і дій системи в UML використовуватимуться діаграми станів та діаграми діяльності.

Діаграма станів (рисунок 2.17) відображає можливі стани об'єктів у системі та переходи між ними. Наприклад, для об'єкта "Замовлення" в системі можуть бути такі стани:

- нове замовлення: замовлення щойно створено користувачем, але ще не підтверджено або не оплачено;
- очікує оплати: користувач перейшов до етапу оплати, але транзакція ще не завершена;
- оплачено: оплата за замовлення успішно отримана;
- готується: піцерія почала приготування замовлення;
- готово до доставки: замовлення приготовано та очікує на відправлення кур'єром;
- в дорозі: замовлення передано кур'єру та прямує до клієнта;
- доставлено: замовлення успішно доставлено клієнту;
- скасовано: замовлення було скасовано клієнтом або адміністрацією (наприклад, через відсутність оплати або недоступність інгредієнтів).



Рисунок 2.17 - Діаграма станів об'єкта "Замовлення"

Джерело: розроблено автором самостійно

Діаграма діяльності (рисунок 2.18) описує послідовність дій або робочих процесів у системі. Наприклад, процес оформлення замовлення користувачем:

Основні кроки процесу "Оформлення замовлення":

вибір страв:

користувач переглядає меню веб-сайту;

користувач додає бажані страви до кошика;

користувач може змінювати кількість страв або видаляти їх з кошика.

ведення даних доставки:

користувач переходить до оформлення замовлення;

користувач вводить або обирає адресу доставки (вулиця, будинок, квартира), контактний телефон та ім'я;

система перевіряє коректність введених даних.

вибір способу оплати:

ористувач обирає бажаний спосіб оплати (наприклад, карткою онлайн, готівкою кур'єру).

ідтвердження замовлення:

ористувач переглядає підсумкову інформацію про замовлення (склад, сума, адреса доставки);

ористувач підтверджує замовлення.

бробка замовлення:

истема реєструє нове замовлення в базі даних;

истема надсилає підтвердження замовлення користувачеві;

истема сповіщає адміністрацію піцерії про нове замовлення.



Рисунок 2.18 - Діаграма діяльності "Оформлення замовлення"

Джерело: розроблено автором самостійно

РОЗДІЛ 3

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ ВІРТУАЛЬНОЇ ПІЦЕРІЇ

3.1 Інформаційне забезпечення ІС

Інформаційне забезпечення веб-сайту віртуальної піцерії складається з бази даних, що зберігає інформацію про меню, замовлення, користувачів, та файлової системи для зберігання зображень страв та інших медіа-файлів. Принципи організації ІЗ базуються на забезпеченні актуальності, цілісності, доступності та конфіденційності даних.

Загальні вимоги до організації збору і передання інформації:

- безпека: всі дані, що передаються між клієнтом і сервером, повинні бути зашифровані за допомогою HTTPS;
- валідація: всі вхідні дані повинні бути ретельно перевірені на відповідність встановленим правилам перед обробкою та збереженням;
- ефективність: процеси збору та передачі інформації повинні бути оптимізовані для мінімізації затримок та навантаження на систему;
- зручність: інтерфейси для введення даних повинні бути інтуїтивно зрозумілими та зручними для користувача та адміністратора.

Структура інформаційних масивів:

асив «Користувачі (Users)»:

Таблиця 3.1 – Масив «Користувачі»

| Найменування | Значення |
|---------------------------|---|
| Найменування масиву | Користувачі (Users) |
| Ідентифікатор масиву | |
| Найменування полів | |
| Максимальний об'єм масиву | Залежить від доступних системних ресурсів |
| Довжина запису | Змінна (залежить від довжини текстових полів) |
| Метод організації | Прямий доступ за ID, індексований доступ за Email |

| | |
|---------------------|--|
| Ключі упорядкування | Первинний ключ: ID> Вторинний ключ: Email (як унікальний ідентифікатор). |
|---------------------|--|

Джерело: розроблено автором самостійно

асив «Категорії Страв (Categories)»:

Таблиця 3.2. Масив «Категорії страв»

| Найменування | Значення |
|---------------------------|------------------------------|
| Найменування масиву | Категорії Страв (Categories) |
| Ідентифікатор масиву | |
| Найменування полів | |
| Максимальний об'єм масиву | Не визначено |
| Довжина запису | Змінна |
| Метод організації | Прямий доступ за ID |
| Ключі упорядкування | Первинний ключ: ID |

Джерело: розроблено автором самостійно

асив «Страви (Dishes)»:

Таблиця 3.3. Масив «Страви»

| Найменування | Значення |
|---------------------------|--|
| Найменування масиву | Страви (Dishes) |
| Ідентифікатор масиву | |
| Найменування полів | |
| Максимальний об'єм масиву | Не визначено |
| Довжина запису | Змінна |
| Метод організації | Прямий доступ за ID, індексований доступ за CategoryID |
| Ключі упорядкування | Первинний ключ: ID> Зовнішній ключ: CategoryID (посилається на ID в масиві "Категорії Страв"). |

Джерело: розроблено автором самостійно

асив «Замовлення (Orders)»:

Таблиця 3.4. Масив «Замовлення»

| Найменування | Значення |
|---------------------------|--|
| Найменування масиву | Замовлення (Orders) |
| Ідентифікатор масиву | |
| Найменування полів | |
| Максимальний об'єм масиву | Не визначено |
| Довжина запису | Змінна |
| Метод організації | Прямий доступ за ID, індексований доступ за UserID |
| Ключі упорядкування | Первинний ключ: ID> Зовнішній ключ: UserID (посилається на ID в масиві "Користувачі"). |

Джерело: розроблено автором самостійно

асив «Позиції замовлення (Order Items)»:

Таблиця 3.5. Масив «Позиції замовлення»

| Найменування | Значення |
|---------------------------|--|
| Найменування масиву | Позиції замовлення (Order Items) |
| Ідентифікатор масиву | |
| Найменування полів | |
| Максимальний об'єм масиву | Не визначено |
| Довжина запису | Змінна |
| Метод організації | Прямий доступ за ID, індексований доступ за OrderID та |
| Ключі упорядкування | Первинний ключ: ID> Зовнішній ключ: OrderID (посилається на ID в масиві "Замовлення")> Зовнішній ключ: DishID (посилається на ID в масиві "Страви"). |

Джерело: розроблено автором самостійно

Інформаційне забезпечення включає:

- базу даних (БД): основний засіб зберігання даних (Додаток А);
- класифікатори і коди: для ідентифікації об'єктів системи;
- метадані: забезпечують інтеграцію і взаємодію компонентів системи;
- інтерфейсні кадри: обробляють взаємодію між користувачем і системою.

Таблиця 3.6. Основні сутності бази даних

| Таблиця | Атрибути |
|---------|----------|
|---------|----------|

| | |
|--------------------|---|
| Користувачі | ID, Email, PasswordHash, Name, Phone, Address, RegistrationDate |
| Категорії Страв | ID, CategoryName, Description |
| Страви | ID, DishName, Description, Price, ImageURL, CategoryID, IsAvailable |
| Замовлення | ID, UserID, OrderDate, TotalAmount, DeliveryAddress, DeliveryPhone, Status, Comment |
| Позиції замовлення | ID, OrderID, DishID, Quantity, UnitPrice |

Джерело: розроблено автором самостійно

Інфологічна модель:

- Користувачі – Замовлення: 1 – Б;
- Категорії Страв – Страви: 1 – Б;
- Страви – Позиції Замовлення: 1 – Б;
- Замовлення – Позиції Замовлення: 1 – Б;

Для збереження даних використовувалася реляційна БД MySQL (рисунок

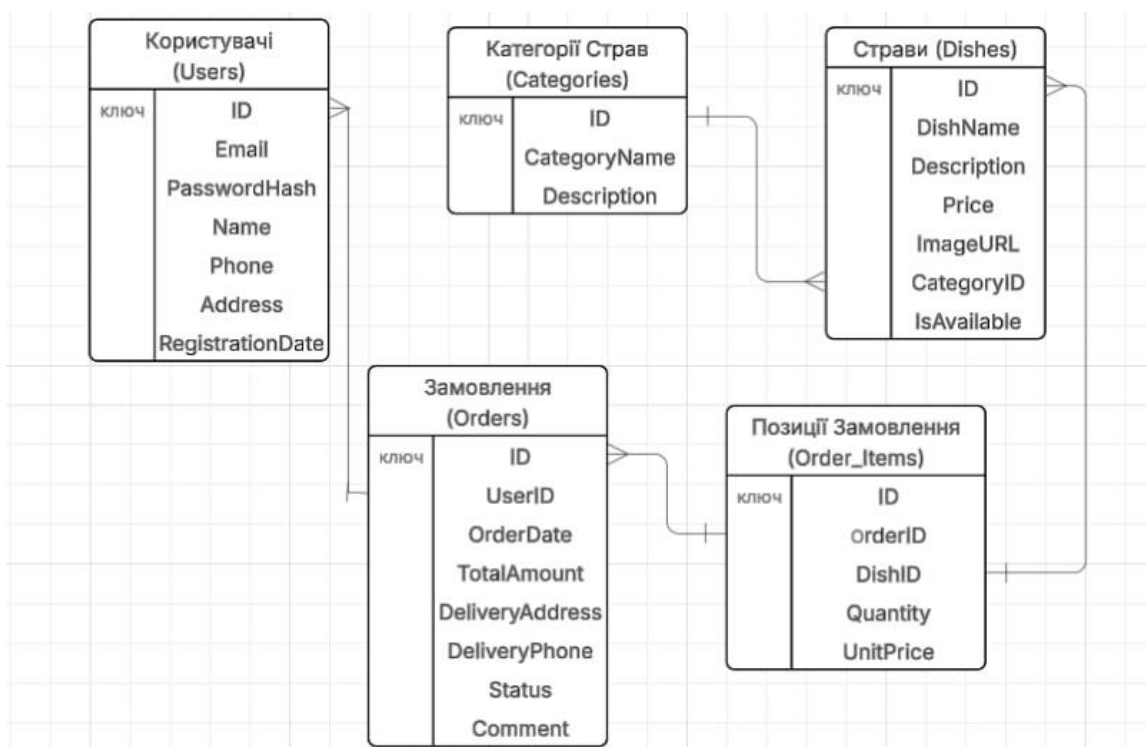


Рисунок 3.1 – Логічна модель бази даних

Джерело: розроблено автором самостійно

3.2 Технічне забезпечення ІС

Для забезпечення ефективного функціонування веб-сайту віртуальної піцерії з використанням фронтенд-технологій необхідно обрати сучасні, продуктивні та доступні технічні засоби (рисунки 3.2).

Рішення має забезпечити:

- продуктивність: швидку обробку запитів та підтримку багаторазового доступу;
- масштабованість: можливість розширення системи без значних витрат;
- доступність: постійний доступ до системи з будь-якого пристрою;
- надійність: мінімізацію простоїв та втрат даних.

Серверна інфраструктура:

- сервери додатків для обробки запитів, виконання серверної логіки (API для взаємодії з фронтендом);
- сервер бази даних для зберігання даних про замовлення, меню, користувачів тощо.

Клієнтські пристрої:

- ПК та ноутбуки: для адміністраторів, менеджерів піцерії та клієнтів (для повноцінного перегляду та взаємодії);
- мобільні пристрої: для клієнтів (зручний доступ до меню та швидкого замовлення з будь-якого місця).

Мережеве обладнання:

- маршрутизатор для забезпечення локальної мережі (для офісу піцерії/адміністративного доступу) та підключення до Інтернету;
- комутатор для розподілу мережевого трафіку між пристроями;
- доступ до Інтернету: мінімальна швидкість — 50 Мбіт/с (рекомендована для комфортної роботи користувачів та обробки замовлень).

Хмарна інфраструктура:

- Хостинг: Amazon AWS, Google Cloud або Microsoft Azure (для розміщення фронтенду, бекенду та бази даних);

- переваги: резервне копіювання, висока доступність, легке масштабування, глобальне охоплення, інтеграція з іншими сервісами.



Рисунок 3.2 - UML-діаграма розгортання

Джерело: розроблено автором самостійно

3 Програмне забезпечення ІС

Для розробки веб-сайту віртуальної піцерії з використанням фронтенд-технологій передбачено використання наступних компонентів:

Операційна система:

- windows Server (або Linux Server) (для серверної частини, якщо така передбачається для бекенду або бази даних);
- windows, macOS, Linux (для клієнтських машин розробників та адміністраторів).

Система управління базами даних (СУБД):

- PostgreSQL або MySQL (для зберігання даних про меню, замовлення, користувачів та іншої інформації).

Мережеве програмне забезпечення:

- сервер Apache або Nginx (для розміщення статичних файлів фронтенду та проксування запитів до бекенду, якщо він є);
- API для інтеграції зі сторонніми сервісами (наприклад, платіжними системами, службами доставки).

Основною мовою програмування для розробки клієнтської частини буде JavaScript. Адже її можливо використовувати в браузері для інтерактивності та динамічного відображення контенту. Для створення структури та стилізації сторінок використовуватимуться HTML та CSS відповідно.

Фреймворки та бібліотеки:

- react або Vue.js (для створення інтерактивного та компонентного клієнтського інтерфейсу, забезпечення швидкої розробки та підтримки).
redux або Vuex (для управління станом веб-додатку, якщо це необхідно для складної логіки);
- бібліотеки для роботи з HTTP-запитами (наприклад, Axios або Fetch API);
- бібліотеки для валідації форм, управління маршрутизацією, анімацій тощо (за потреби).

Інструменти розробки:

- середовища розробки (IDE): Visual Studio Code;
- системи контролю версій: Git;
- менеджери пакетів: npm;
- інструменти для збірки та мініфікації ассетів: Webpack;
- інструменти для тестування: Jest.

Прикладне програмне забезпечення (рисунок 3.3):

Система включає наступні модулі:

- Модуль перегляду меню:
 - призначення: надання користувачам можливості переглядати доступні піци та інші страви з їх описом, ціною та фотографіями;

- функції: відображення категорій страв, пошук за назвою, фільтрація за інгредієнтами, перегляд детальної інформації про страву;
- використання: клієнти для ознайомлення з асортиментом.
- Модуль формування замовлення:
 - призначення: забезпечення функціоналу додавання страв до кошика, редагування кількості, вибору розміру та додаткових опцій;
 - функції: додавання до кошика, видалення з кошика, зміна кількості, відображення підсумкової суми, застосування промокодів (за наявності);
 - використання: клієнти для складання замовлення.
- Модуль оформлення замовлення:
 - призначення: збір інформації про клієнта (ім'я, контактні дані, адреса доставки), вибір способу оплати та доставки, підтвердження замовлення;
 - функції: введення контактних даних, вибір адреси доставки (можливість збереження адрес), вибір способу оплати, вибір часу доставки, підтвердження замовлення та оплата (інтеграція з платіжними системами);
 - використання: клієнти для завершення процесу замовлення.
- Модуль особистого кабінету (за потреби):
 - призначення: надання зареєстрованим користувачам можливості переглядати історію замовлень, зберігати улюблені страви та адреси доставки;
 - функції: перегляд історії замовлень, повторне замовлення, редагування особистих даних, збереження адрес доставки;
 - використання: зареєстровані клієнти для зручнішого користування сервісом.
- Адміністративна панель (серверна частина, взаємодія через API):

- призначення: управління меню (додавання, редагування, видалення страв), категоріями, цінами, акціями, перегляд та управління замовленнями;
- функції: додавання/редагування/видалення страв та категорій, зміна цін, управління статусами замовлень, перегляд статистики (за потреби);
- використання: адміністратори та менеджери піцерії для управління контентом та замовленнями.



Рисунок 1.3 - UML-діаграм компонентів

Джерело: розроблено автором самостійно

Результати реалізації веб-сайту віртуальної піцерії

Для найкращої роботи всієї системи та сайту в цілому для мене досить важливим етапом є розроблення тест-кейсів для перевірки належної роботи всіх деталей. Далі буде наведено декілька прикладів тестів, які були проведені мною.

Тест 1.

Успішний тестовий випадок для функціональності додавання товару до кошика перевіряє, що після кліку на кнопку «Купити» відповідний товар додається до масиву `saveData`, а також зберігається у `localStorage`.

Перед виконанням тесту здійснюється початкове налаштування середовища: створюється макет DOM, мокується `localStorage`, а також підключаються необхідні залежності. Для перевірки виконується імітація кліку на кнопку, що тригерить додавання товару. Після цього перевіряється, що масив `saveData` містить один елемент з правильним `idDrink`, а також що метод `localStorage.setItem` був викликаний, підтверджуючи збереження інформації. Ще один тестовий випадок стосується функції `goodHtml`, яка відповідає за рендеринг кошика. Успішне виконання тесту означає, що при наявності товарів у кошику, вони правильно відображаються у таблиці (`.cartTable`), включаючи назву напою чи піци та його зображення.

Останній тестовий сценарій перевіряє поведінку `goodHtml`, коли у `localStorage` немає збережених товарів. У цьому випадку функція повинна вивести повідомлення «Ваш кошик порожній», що також перевіряється шляхом аналізу HTML-коду таблиці.

```
(node:8284) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative inst
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/utills/cart.test.js
  Кошик покупок - Тести
    ✓ addToCart додає товар до кошика при кліку на кнопку "Купити" (201 ms)
    ✓ goodHtml правильно відображає вміст кошика (52 ms)
    ✓ goodHtml показує повідомлення про порожній кошик, якщо немає даних (38 ms)

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 7.736 s
```

Рисунок 3.4 - Результат виконання першого тест-кейсу

Джерело: розроблено автором самостійно

Тест 2.

У даному тестовому випадку перевіряється функціональність відкриття та закриття модального вікна кошика. Основна мета тестування – переконатися, що модальне вікно відкривається при взаємодії користувача з іконкою кошика та закривається при натисканні відповідних кнопок, кліку на фон або натисканні клавіші Escape.

Модульний тест написано за допомогою Jest із використанням jsdom для емуляції DOM-структури. Оскільки при відкритті модального вікна має викликатися функція `goodHtml()`, що відповідає за рендеринг товарів у кошику, її було змоковано за допомогою `jest.mock()`. Перед кожним тестом створюється HTML-структура з усіма необхідними елементами, серед яких `.cart-container`, що виконує роль фону модального вікна, `.cart-icon`, що відповідає за відкриття вікна, `.cartModal`, яка містить вміст кошика, `[data-action=«close-modal»]`, що є кнопкою закриття, та `.makeOrder`, яка імітує оформлення замовлення.

Тестування включає кілька сценаріїв. Спершу перевіряється відкриття модального вікна при кліку на іконку кошика. Для цього імітується подія кліку на `.cart-icon`, після чого перевіряється, чи доданий клас `show-modal` до `.cartModal` та `.cart-container`, а також чи викликається функція `goodHtml()`. Далі тестується закриття вікна при натисканні кнопки закриття, що реалізується аналогічним чином – після імітації кліку перевіряється, чи видаляється клас `show-modal`.

Окрім цього, тестуємо сценарій закриття модального вікна при кліку на фон. У цьому випадку імітується подія `click` на `.cart-container`, і очікується, що модальне вікно закриється, тобто відповідні класи буде видалено. Аналогічно перевіряється закриття модального вікна при натисканні клавіші Escape – імітується подія `keydown`, а тест перевіряє, чи зникне модальне вікно. Останній тест перевіряє, чи закриється модальне вікно при натисканні на кнопку «Оформити замовлення».

За результатами тестування встановлено, що всі перевірені сценарії виконуються коректно, що підтверджує правильну інтеграцію даного модуля в систему.

```
(node:17460) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/utils/try.test.js
  Функціональність відкриття/закриття модального вікна кошика
    ✓ Клік на іконку кошика повинен відкривати модальне вікно (78 ms)
    ✓ Клік на кнопку закриття повинен закривати модальне вікно (22 ms)
    ✓ Клік на фон повинен закривати модальне вікно (30 ms)
    ✓ Натискання клавіші Escape повинно закривати модальне вікно (20 ms)
    ✓ Клік на кнопку "Оформити замовлення" повинен закривати модальне вікно (21 ms)

Test Suites: 1 passed, 1 total
Tests: 5 passed, 5 total
Snapshots: 0 total
Time: 6.82 s
```

Рисунок 3.5 - Результат виконання другого тест-кейсу

Джерело: розроблено автором самостійно

Тест 3.

У цьому тестовому випадку перевіряється функціональність отримання даних про піци з JSON-файлу та їх відображення у вигляді карток товарів на сторінці. Для тестування використовується Jest із середовищем jsdom, що дозволяє імітувати роботу DOM у тестовому середовищі.

Спочатку JSON-файл `pizza.json`, який містить дані про піци, мокується за допомогою `jest.mock()`, щоб уникнути реальних запитів під час тестування. Він містить два тестових об'єкти з інформацією про піци «Маргарита» та «Пепероні». Перед кожним тестом DOM очищається та створюється контейнер `.goods_container`, у який будуть вставлятися картки товарів.

Перша перевірка стосується завантаження даних із JSON-файлу функцією `getdataPizza`. Вона зчитує мокуємі дані, додає їх у глобальний масив `goodsDataPizza`, після чого викликає функцію `renderCardPizza()`, яка відображає отримані товари. Після виклику функції тест перевіряє, чи масив `goodsDataPizza` дійсно містить очікувані дані.

Друга перевірка спрямована на функцію `renderCardPizza()`, яка створює картки товарів у DOM. Тест викликає функцію з мокованими даними та перевіряє, чи в контейнері `.goods_container` з'явилися два елементи `.card`, що відповідають очікуваним піцам. Крім того, перевіряється коректність наповнення карток: заголовки, зображення, ціни та атрибути кнопок «Замовити».

Третій тест фокусується на перевірці кнопок «Замовити». Після виклику `renderCardPizza()` тест перевіряє, чи всі кнопки мають правильні атрибути `data-id` і `data-price`, які відповідають інформації про піци з JSON-файлу.

За результатами тестування встановлено, що всі перевірені сценарії виконуються коректно: дані про піци успішно завантажуються та відображаються у вигляді карток, а кнопки «Замовити» мають правильні атрибути.

```
(node:13740) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
PASS src/utils/get-menu.test.js
  Функціональність отримання даних меню
    ✓ Функція getDataPizza повинна завантажувати дані з JSON файлу (19 ms)
    ✓ Функція renderCardPizza повинна створювати картки товарів в DOM (7 ms)
    ✓ Кнопка "Замовити" повинна мати правильні атрибути data-id та data-price (5 ms)

Test Suites: 1 passed, 1 total
Tests: 3 passed, 3 total
Snapshots: 0 total
Time: 1.921 s
Ran all test suites matching /get-menu.test.js/i.
```

Рисунок 3.6 - Результат виконання третього тест-кейсу

Джерело: розроблено автором самостійно

Результати тестування показали, що всі перевірені функції працюють відповідно до вимог, дані коректно обробляються, а взаємодія користувача з інтерфейсом проходить без помилок. Це свідчить про стабільність та відповідність розроблених компонентів заявленим функціональним вимогам.

Проведено детальний аналіз відповідності кожної функціональної вимоги до реалізованих компонентів веб-сайту віртуальної піцерії. Трасування вимог є важливим етапом верифікації, оскільки воно дозволяє перевірити, чи виконуються всі вимоги, сформульовані на етапі проектування системи, та чи всі компоненти системи правильно інтегровані з іншими елементами.

Протягом цього етапу було здійснено зв'язування функціональних вимог з конкретними частинами коду веб-сайту, що забезпечують їх виконання. Для цього було використано метод трасування, де кожна вимога, яка була визначена на етапі аналізу. Особливу увагу було приділено таким вимогам, як функціональність перегляду меню піцерії, додавання товарів у кошик, оформлення замовлення, обробка взаємодії з модальним вікном, а також перевірки, чи правильно працюють всі кнопки й функції сайту, зокрема наявність необхідних атрибутів в HTML-кодi та коректність взаємодії з користувачем через Java Script. Всі ці вимоги були покриті тестами, що підтвердили їх реалізацію.

Процес трасування включав кілька етапів. Спочатку кожна вимога була зіставлена з конкретним компонентом інтерфейсу користувача або частиною коду. Наприклад, вимога про можливість перегляду меню була перевірена через тестування функції завантаження даних з JSON-файлу та перевірки відображення піц чи напоїв на сторінці. Вимога про можливість додавання товарів до кошика була перевірена через функціональність кнопки «Замовити» та її атрибути.

Крім того, у процесі трасування враховувалися різні рівні тестування, такі як модульні, інтеграційні, системні та приймальні тести. Тести також включали перевірку відповідності інтерфейсу користувача і очікуваного результату.

Таким чином, у результаті трасування вимог було підтверджено, що усі основні функціональні вимоги, визначені на етапі проектування веб-сайту віртуальної піцерії, були коректно реалізовані в системі. Крім того, за допомогою тестів було забезпечено повну перевірку кожної вимоги через тестування основних сценаріїв взаємодії користувача з веб-сайтом. Це дозволило досягти високої надійності та якості розробленої системи, що відповідає вимогам замовника.

Веб-сайт для віртуальної піцерії був розроблений для того, щоб продемонструвати основні функціональні можливості системи, такі як перегляд меню, додавання товарів до кошика, оформлення замовлення та взаємодія з користувачем через інтерфейс. Для цього було обрано використання технологій HTML, CSS та Java Script, що дозволяє створити інтерактивний і зручний у використанні веб-сайт, який відповідає вимогам користувачів і замовника.

Для розробки прототипу була вибрана система, яка відповідає вимогам простоти використання та зручності для кінцевого користувача. Використання Java Script дозволяє реалізувати динамічні функції, такі як обробка кліків на кнопки, відкриття модальних вікон для кошика та автоматичне оновлення інформації на сторінці без необхідності перезавантажувати її.

Для створення сайту були використані такі технології як HTML для створення структури сторінок, CSS для стилізації та створення адаптивного

дизайну, а також Java Script для реалізації динамічних функцій. Враховуючи вимог, було також використано інструменти для створення графічних прототипів, такі як Figma, яка допомогла візуалізувати інтерфейс та макети екранних форм.

В проекті було реалізовано наступні сторінки:

- Головна сторінка;
- Про нас
- Меню;
- Контакти
- Кошик;
- Сторінка авторизації;

Всі сторінки розроблено в одному стилі, аби загальна картинка сайту була гармонійна та приваблювала користувача.

Головна сторінка (рисунок 3.7) містить в собі інформацію, яку користувач вперше бачить, зайшовши на сайт. На цій сторінці присутня «шапка» сайту, яка є на кожній сторінці і навіть при прокручуванні сайту до низу вона не зникає, в ній ми бачимо перелік всіх сторінок, логотип який при натисканні на будь якій сторінці перекидає нас на сторінку з меню, іконку корзини (рисунок 3.8), при натисканні якої нам відкривається модальне вікно з обраними нами товарами, та кнопка увійти/вийти, яка відкриває нам вікно для авторизації через гугл акаунт (рисунок 3.9), при успішній авторизації в нас буде підтягуватися фото з нашого акаунту.

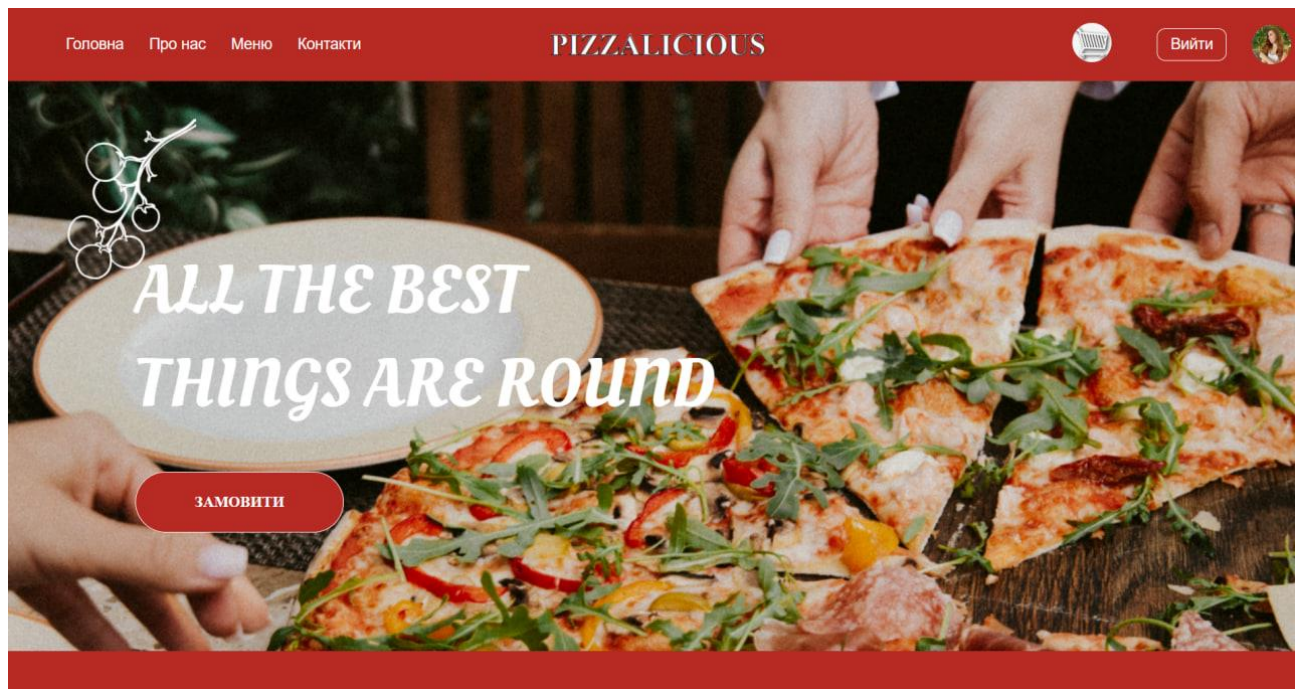


Рисунок 3.7 - Головна сторінка

Джерело: розроблено автором самостійно



Рисунок 3.8 - Іконка корзини

Джерело: розроблено автором самостійно

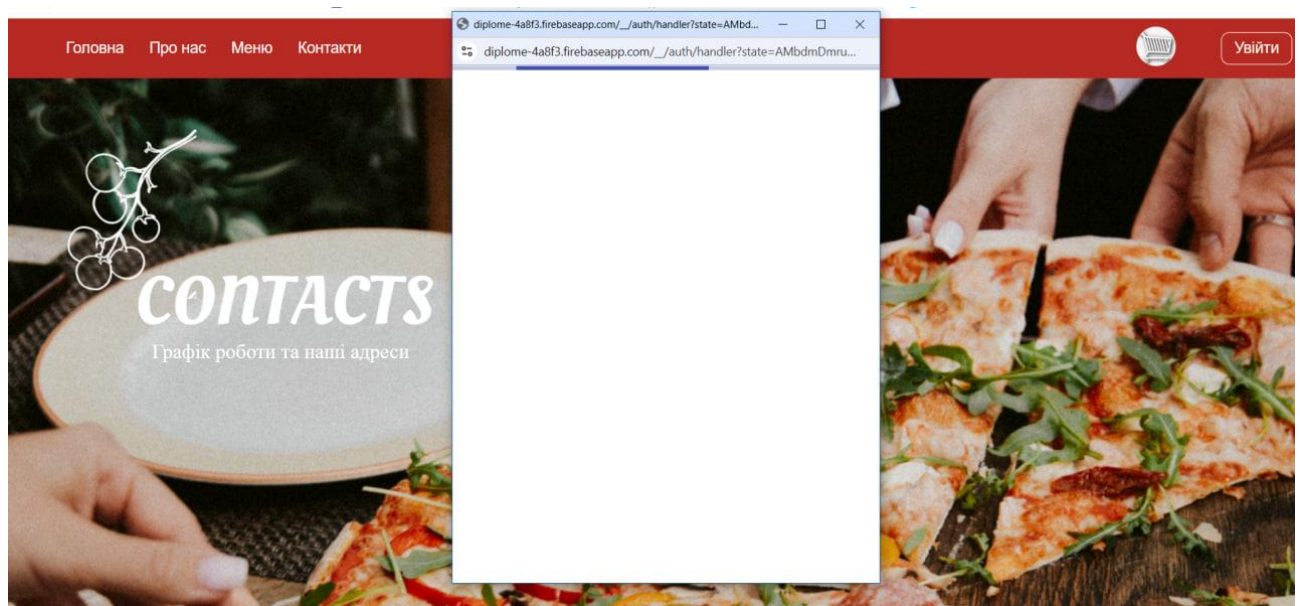


Рисунок 3.9 - Вікно авторизації

Джерело: розроблено автором самостійно

Також на головній сторінці представлена цікава інформація про саму піцерію (рисунок 3.10), наведено багато анімації та фото (рисунок 3.11) для цікавішого перегляду сайту, але в той самий час сторінка не перезавантажена інформацією, аби користувачу було зручно переглядати. Наведено найбільш популярні позиції з меню (рисунок 3.12) та в нижній частині сторінки є контакти, графік роботи і адреса для зручності (рисунок 3.13), які також присутні на кожній сторінці внизу.

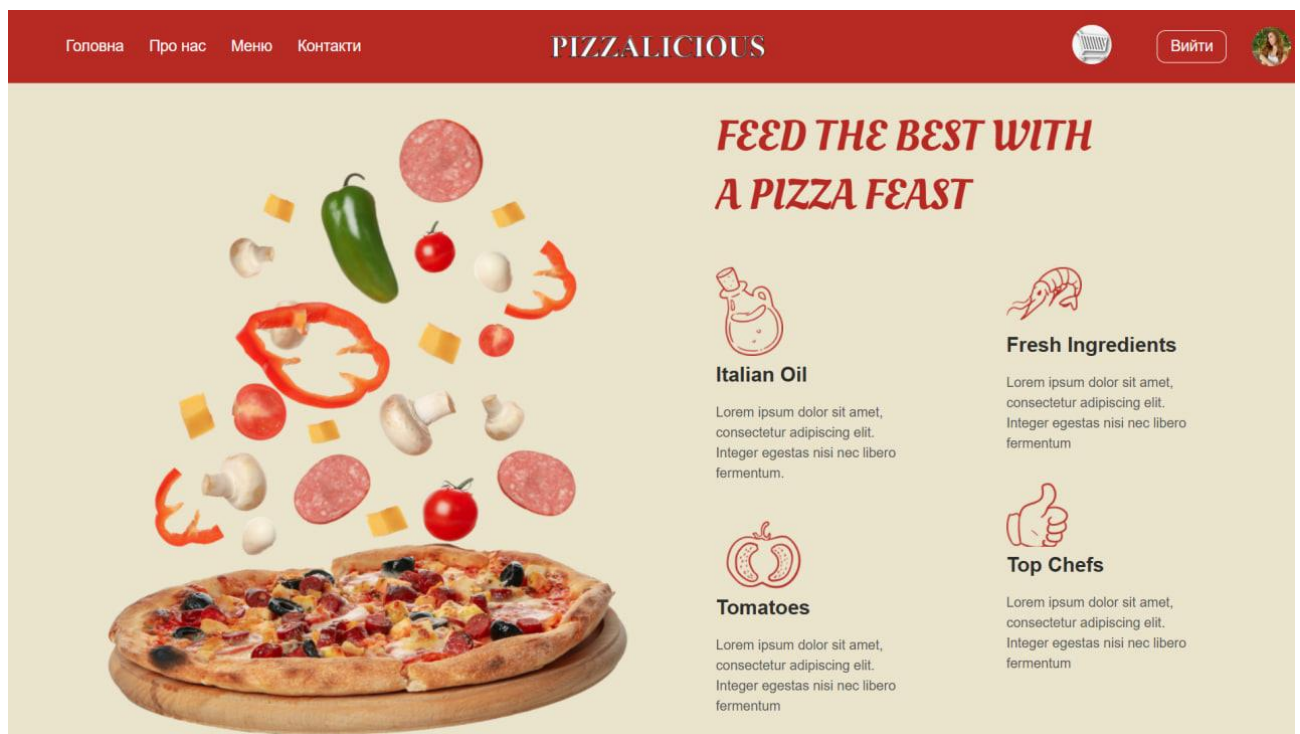


Рисунок 3.10 - Інформація про саму піцерії

Джерело: розроблено автором самостійно

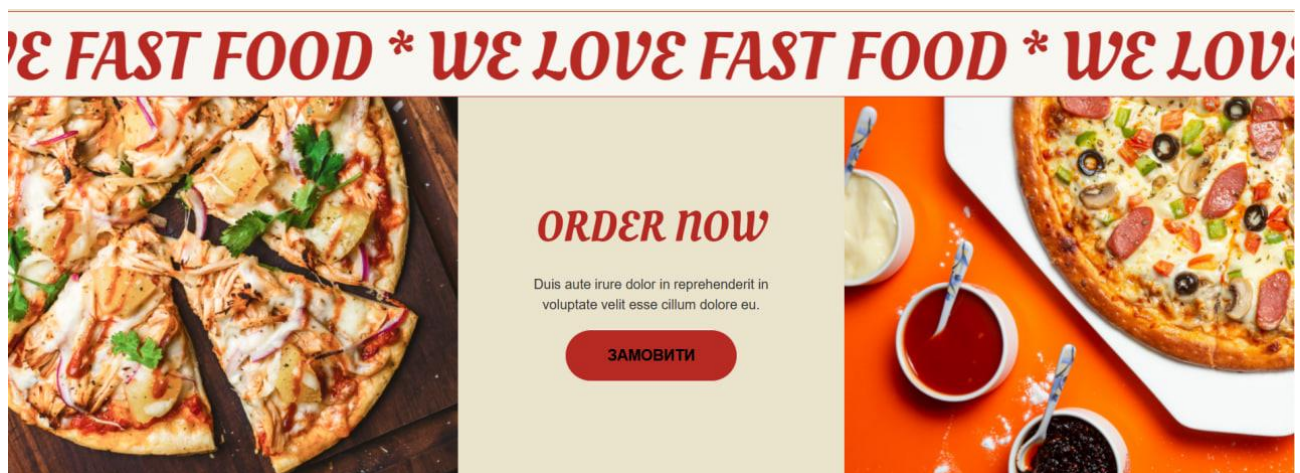


Рисунок 3.11 - Анімації на головній сторінці

Джерело: розроблено автором самостійно

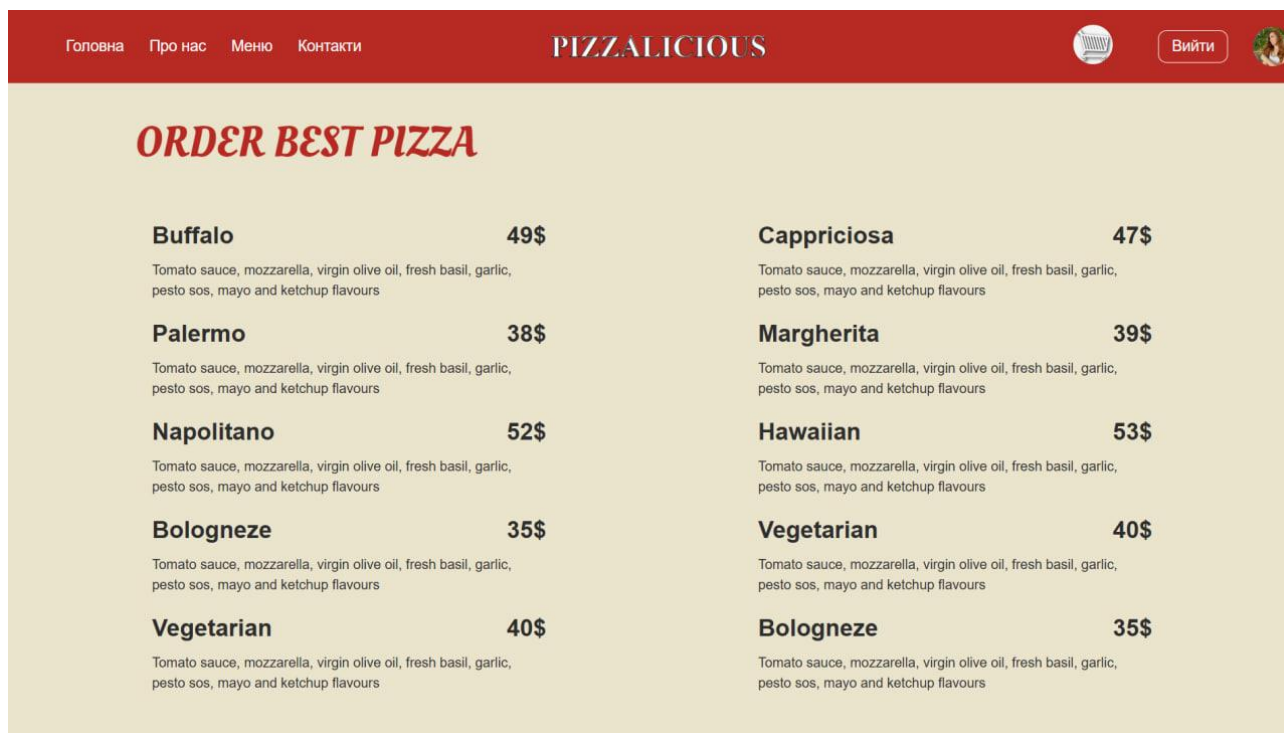


Рисунок 3.12 - Найбільш популярні позиції з меню

Джерело: розроблено автором самостійно



Рисунок 3.13 - Нижня частина головної сторінки

Джерело: розроблено автором самостійно

Сторінка «Про нас» містить зверху заголовок в схожому стилі з головною сторінкою (рисунок 3.14), історію створення піцерії (рисунок 3.15), так само наведено фото (рисунок 3.16), в даному випадку готових піц, які пропонує заклад та так само контактну інформацію.

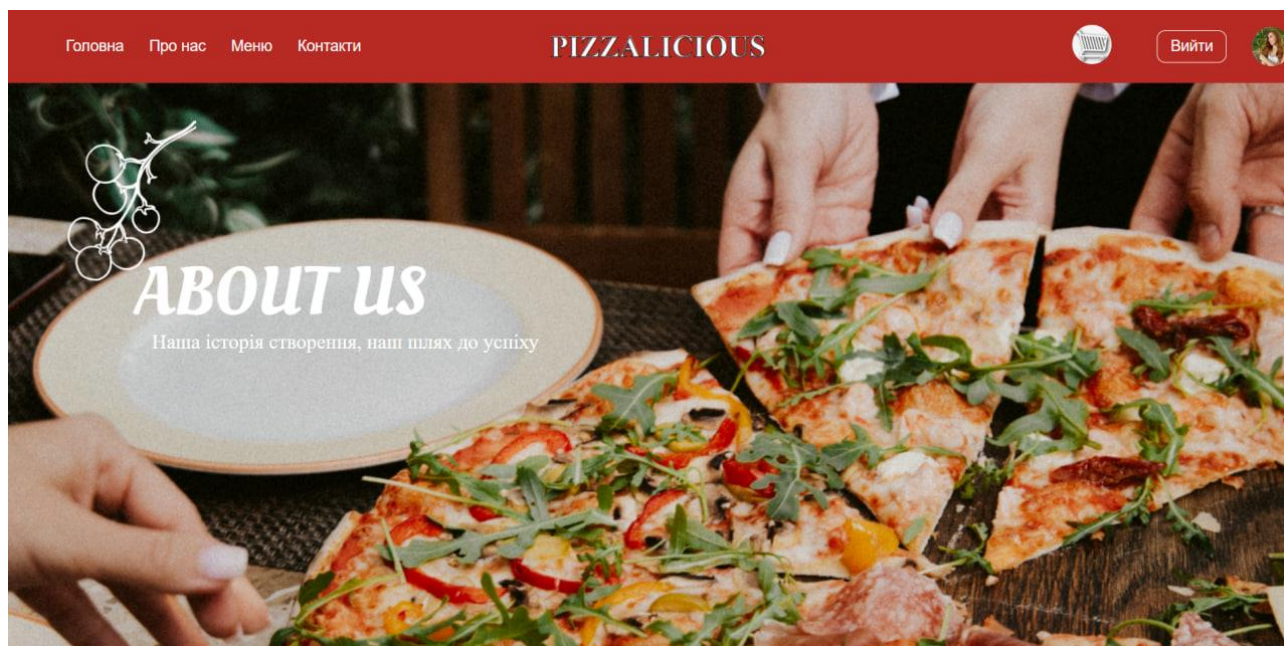


Рисунок 3.14 - Перший екран сторінки «Про нас»

Джерело: розроблено автором самостійно



Рисунок 3.15 - Історія створення піцерії

Джерело: розроблено автором самостійно

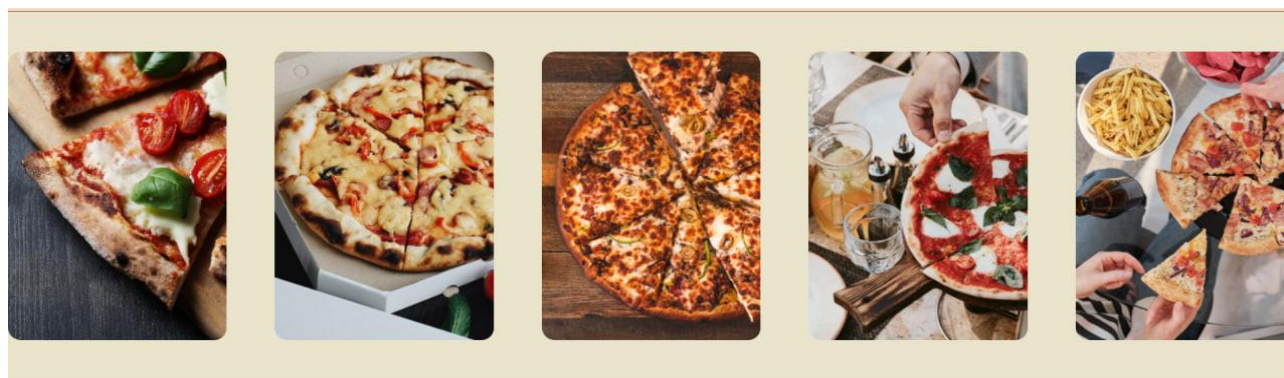


Рисунок 3.16 - Фото готових піц

Джерело: розроблено автором самостійно

Сторінка «Меню» зустрічає нас заголовком (рисунок 3.17) так само як і інші, тут представлено різноманітний вибір піц та напоїв доступних до замовлення (рисунки 3.18 – 3.21), під кожною карточкою міститься фото товару, назва та його вартість.

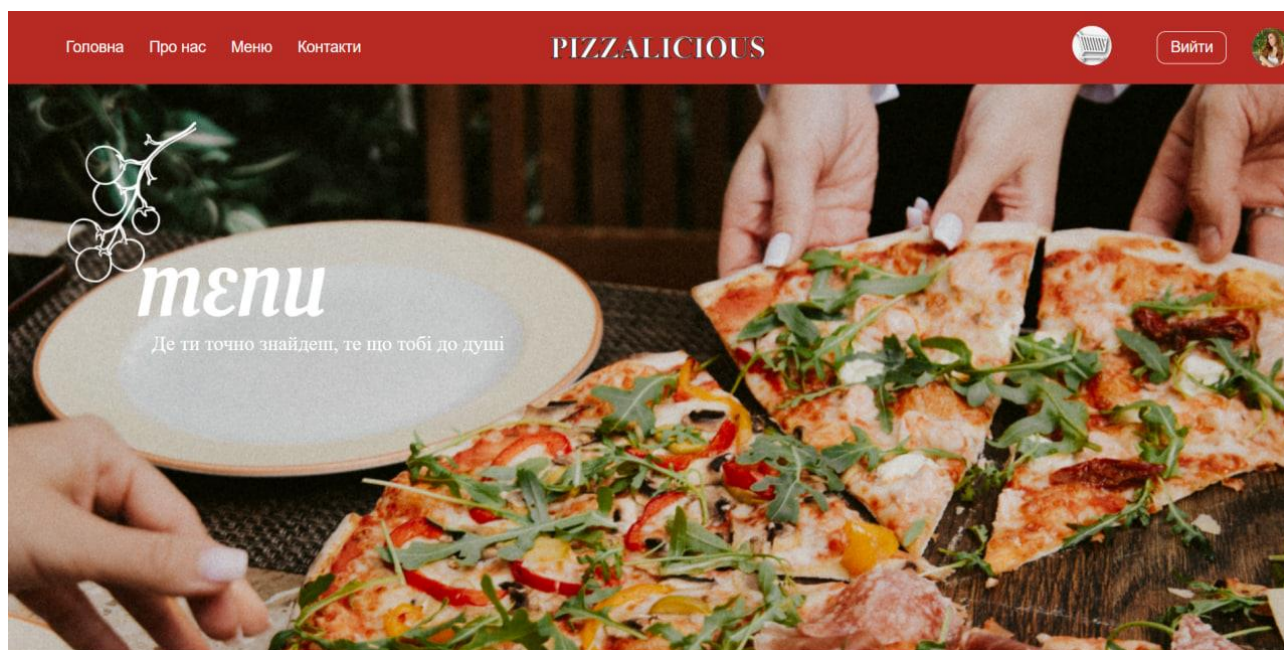


Рисунок 3.17 - Перши екран сторінки «Меню»

Джерело: розроблено автором самостійно

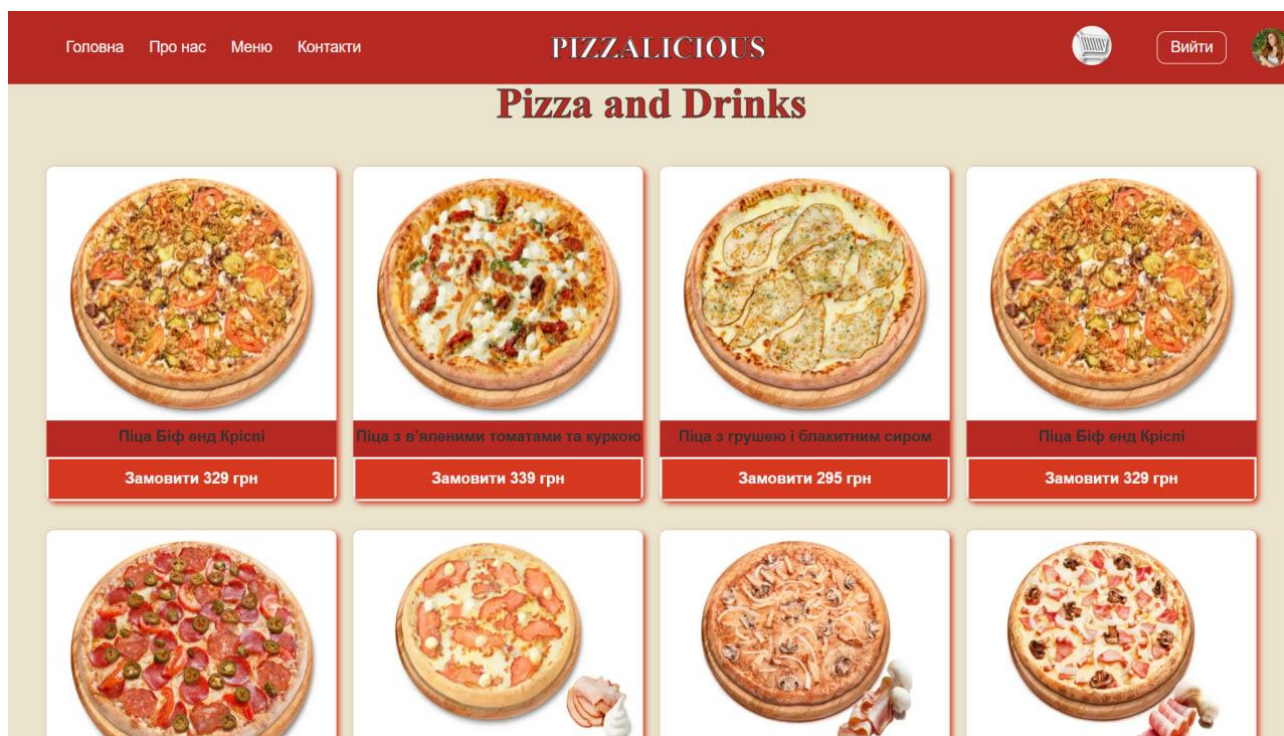


Рисунок 3.18 - Вибір піц

Джерело: розроблено автором самостійно

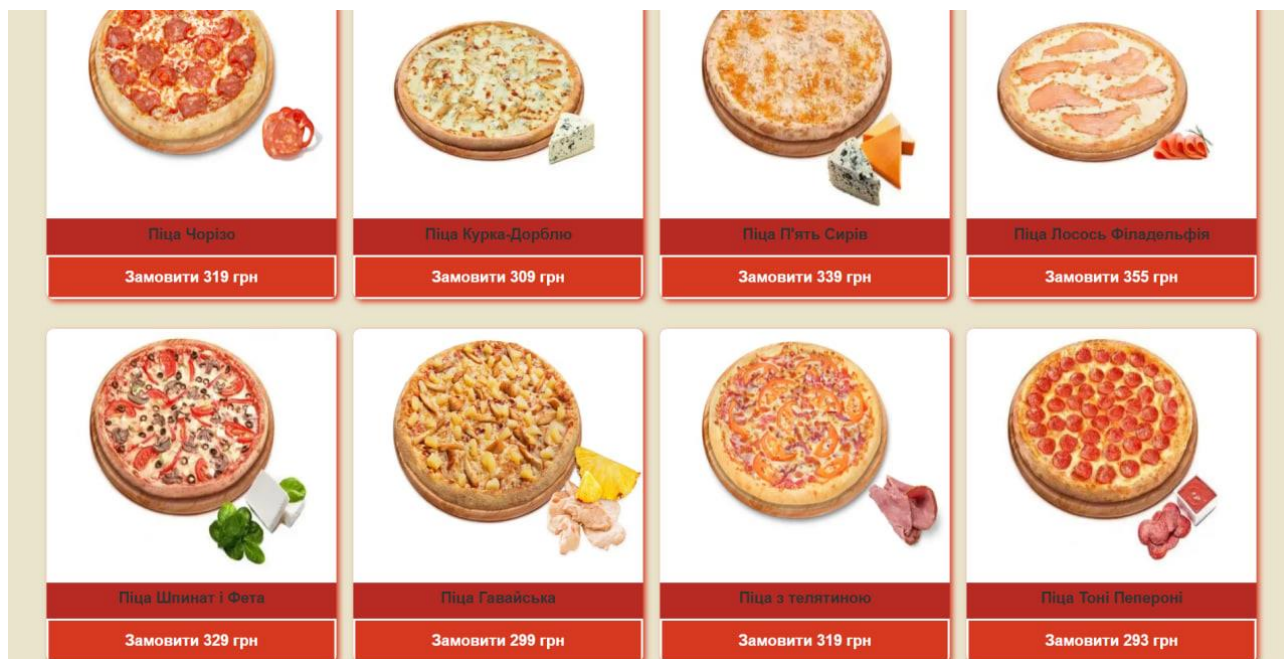


Рисунок 3.19 - Вибір піц

Джерело: розроблено автором самостійно

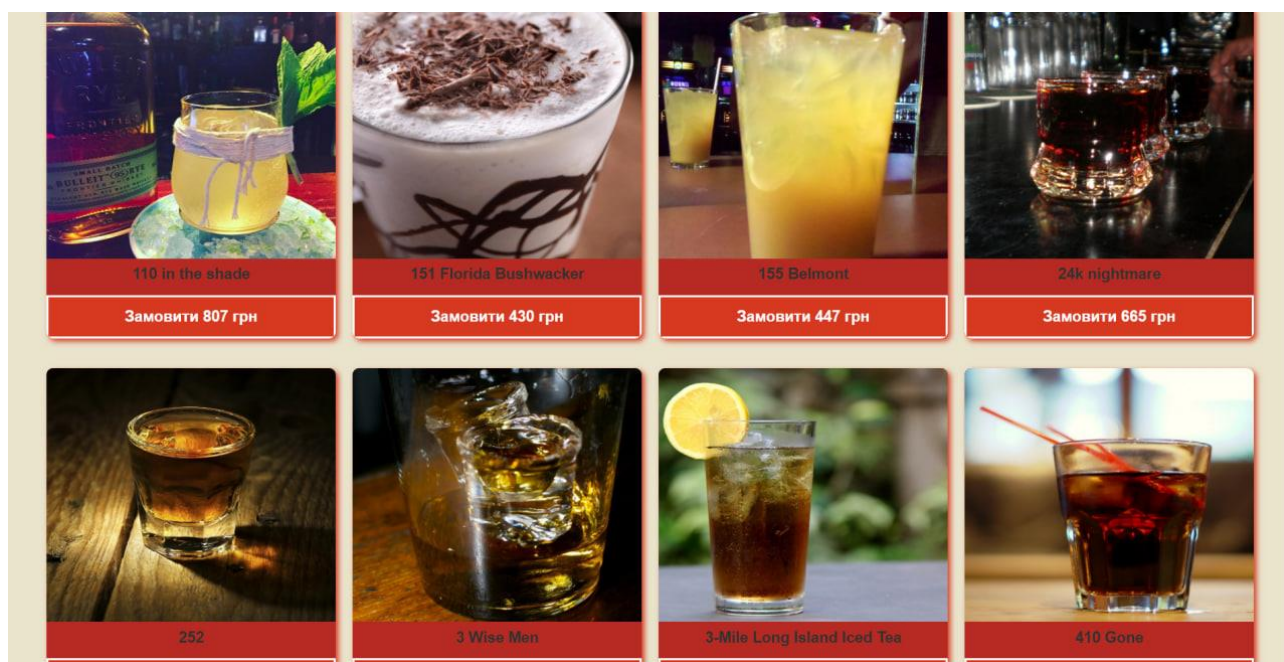


Рисунок 3.20 - Вибір напоїв

Джерело: розроблено автором самостійно

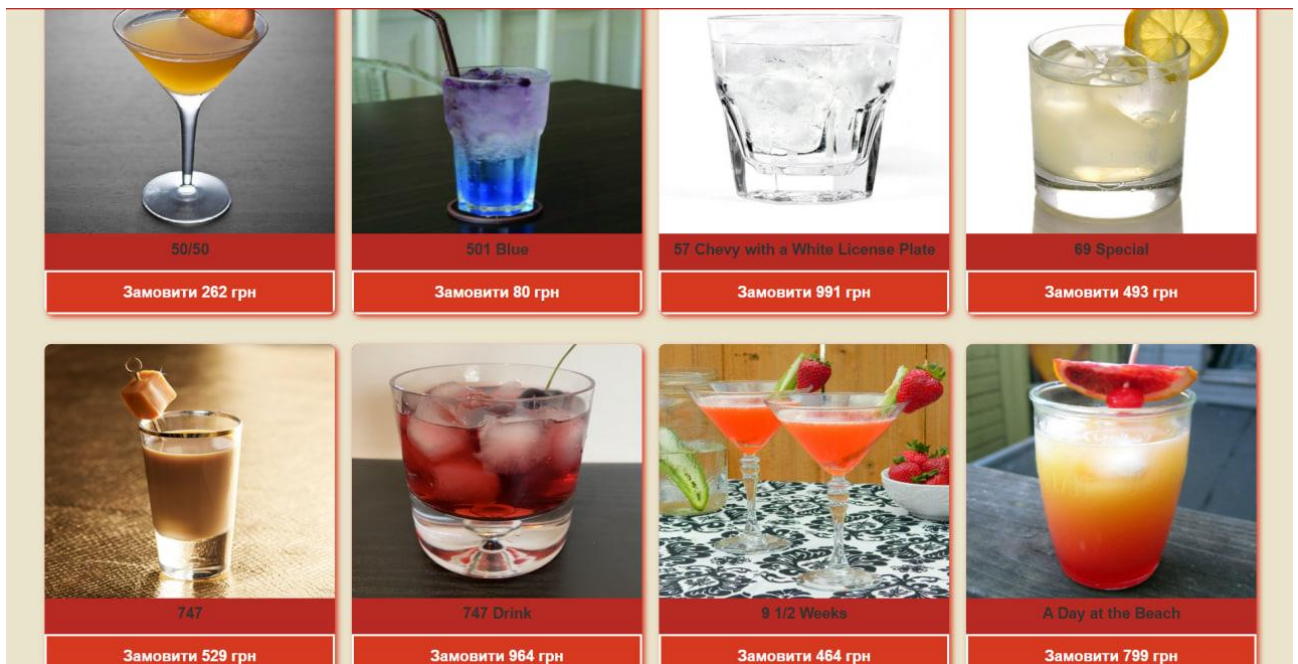


Рисунок 3.21 - Вибір напоїв

Джерело: розроблено автором самостійно

І остання сторінка містить в собі контактну інформацію з локацією, номерами телефонів та графіком роботи (рисунки 3.22 - 3.24)

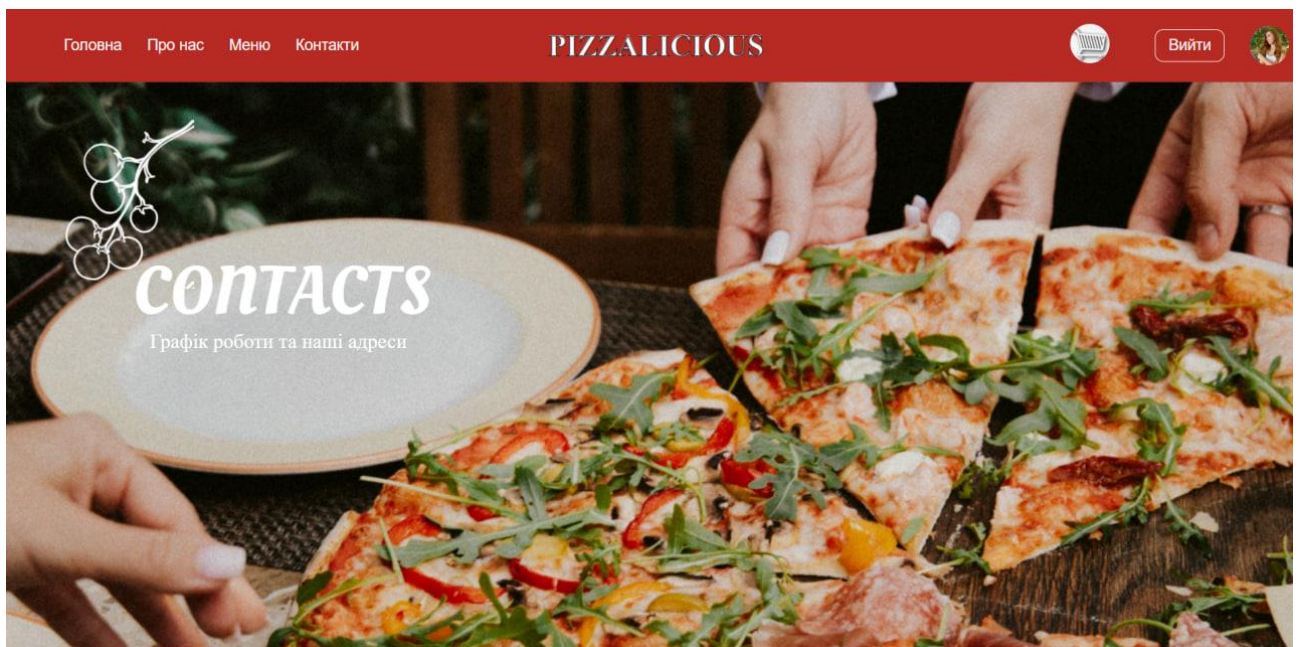


Рисунок 3.22 -Головний екран сторінки «Контакти»

Джерело: розроблено автором самостійно

OUR LOCATION AND CONTACTS

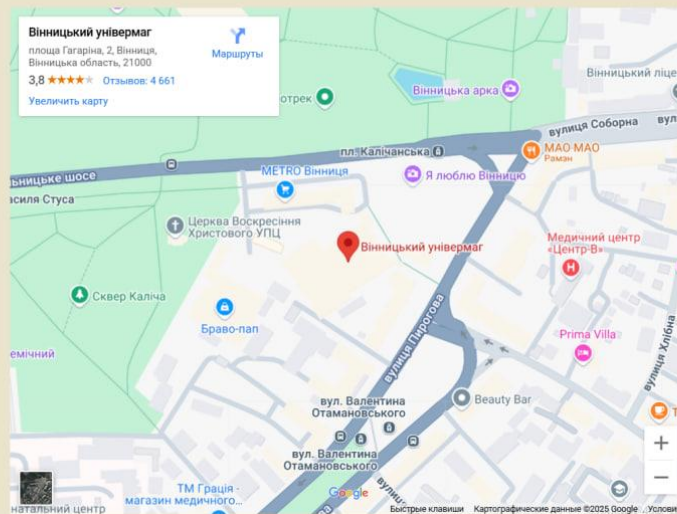


Рисунок 3.23 - Локація піцерії

Джерело: розроблено автором самостійно

Затишний ресторан в центрі міста

площа Гагаріна, 2, Вінниця, Вінницька область

+012-345-6789 Pizzalicious@contact.com

Opening Hours

Monday/Friday 9:00-23:00

Saturday 10:00-21:00

Weekend Closed

T FOOD * WE LOVE FAST FOOD * WE LOVE FAST

Рисунок 3.24 - Адреси піцерій та графік роботи

Джерело: розроблено автором самостійно

Відкриваючи кошик, в нас з'являється модальне вікно яке представлено

Н
И
Ж
Ч
е

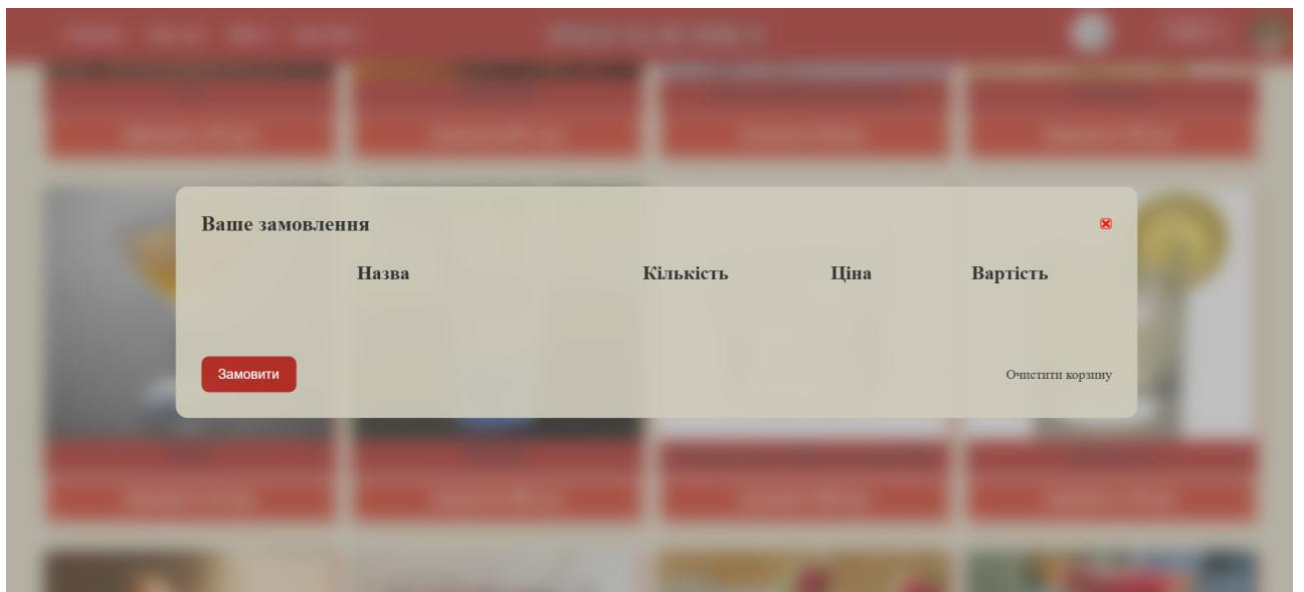


Рисунок 3.25 - Пустий кошик

Джерело: розроблено автором самостійно

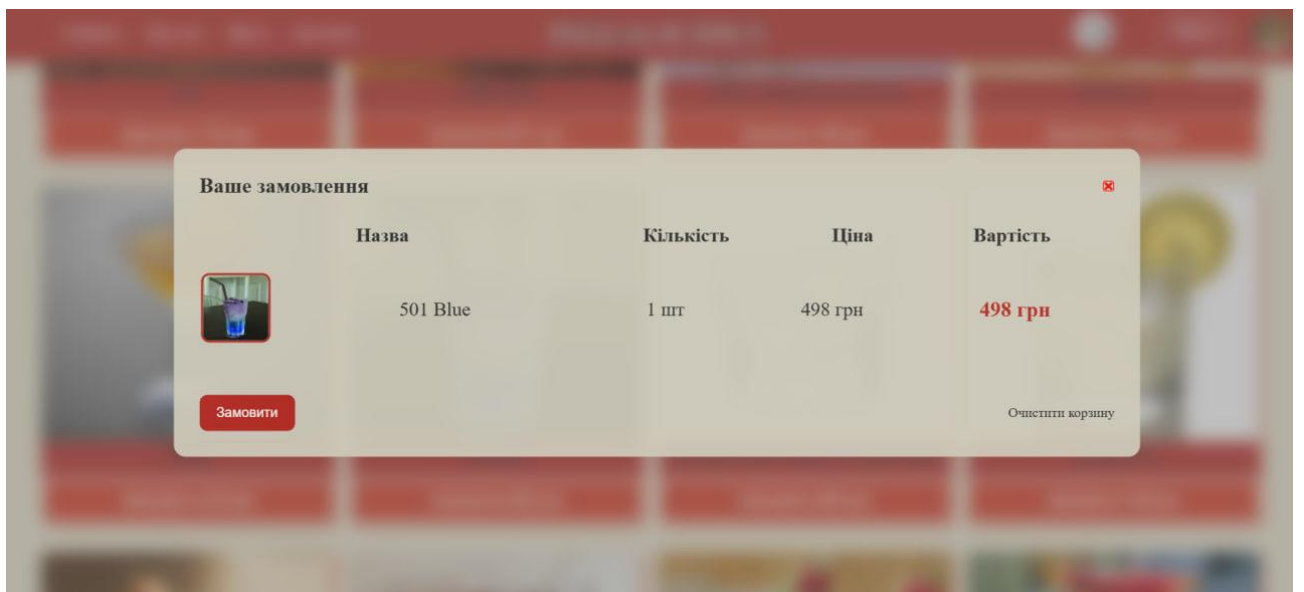


Рисунок 3.26 - Кошик із доданим товаром

Джерело: розроблено автором самостійно

ВИСНОВКИ

У ході виконання роботи була реалізована система, яка включає всі основні елементи для створення функціонального веб-сайту для онлайн-продажу піци. У роботі було розглянуто всі етапи розробки, починаючи від моделювання предметної області до створення прототипу веб-сайту, тестування та верифікації проектних рішень.

Першим етапом було проведення детального аналізу предметної області, що дозволило сформулювати вимоги до системи. Це включало як бізнес-вимоги, так і функціональні та нефункціональні вимоги, що є основою для подальшого проектування та реалізації системи. Визначено, що основними функціями веб-сайту є відображення меню піц, взаємодія з кошиком, а також оформлення замовлень.

Проектування системи було здійснено за допомогою методів моделювання поведінки, структури та користувацького інтерфейсу. Моделювання дозволило створити чітке уявлення про роботу системи, її компоненти та їх взаємодію. Зокрема, особливу увагу було приділено проектуванню зручного та інтуїтивно зрозумілого інтерфейсу, що забезпечує ефективну взаємодію з користувачем.

Тестування та верифікація проектних рішень дозволили переконатися у правильності реалізації системи. Були розроблені тести різних типів (модульні, інтеграційні, системні та приймальні), що допомогло виявити та усунути потенційні проблеми на ранніх етапах розробки. Трасування вимог до системи продемонструвало правильність співвіднесення вимог із проектними компонентами та функціональними можливостями сайту.

У результаті був створений веб-сайт для віртуальної піцерії, що містить усі основні функції для перегляду меню, додавання товарів у кошик, оформлення замовлень та взаємодії з користувачем. Сайт був успішно протестований на предмет зручності користування та відповідності вимогам.

Однак, незважаючи на досягнуті успіхи, проект ще потребує значних вдосконалень для повної відповідності сучасним вимогам ринку. Зокрема,

важливо впровадити систему оплати, яка дозволить користувачам здійснювати платежі безпосередньо через платформу. Така функціональність значно покращить користувацький досвід та підвищить зручність використання системи, дозволяючи користувачам швидко та безпечно здійснювати фінансові транзакції. Впровадження системи оплати також відкриє нові можливості для монетизації платформи, що є важливим аспектом для забезпечення її сталого розвитку та фінансової стійкості.

Додатково, рекомендовано розширити адміністративний функціонал, додавши інструменти для аналітики та моніторингу системи в режимі реального часу. Це дасть змогу адміністраторам оперативно реагувати на будь-які виникаючі проблеми та ефективніше управляти ресурсами системи. Аналітичні інструменти дозволять відслідковувати ключові показники ефективності та виявляти області, що потребують покращення, забезпечуючи тим самим більш ефективне управління платформою.

Вдосконалення проекту передбачає збільшення функціональних можливостей, що дозволить підприємству ефективніше використовувати інформаційні технології у своїй діяльності. Особливу увагу слід звернути на інтеграцію системи оплати та вдосконалення адміністративного інтерфейсу для більш зручного управління системою. Реалізація цих заходів дозволить значно підвищити загальну ефективність та конкурентоспроможність впроваджуваного рішення, що сприятиме успішному розвитку підприємства.

Також варто зазначити, що подальший розвиток проекту може включати впровадження додаткових функцій, таких як автоматизація процесів обробки даних, інтеграція з іншими системами та сервісами, а також розробка мобільних додатків для зручного доступу до платформи з будь-яких пристроїв. Ці кроки дозволять забезпечити більш гнучке та ефективне управління бізнес-процесами, підвищити задоволеність користувачів та сприяти зростанню доходів підприємства.

У підсумку, виконана робота продемонструвала можливість реалізації віртуальної піцерії на базі сучасних веб-технологій, таких як Java Script, HTML та CSS.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

енісова О. О., Сендзюк М. А. Інформаційні системи в економіці [Електронний

ресурс] : підручник. — Київ : КНЕУ, 2023. — 311, [1] с. (дата звернення:

oodle. URL: <https://moodle.org/?lang=uk> (дата звернення: 01.05.2025).

lassroom. URL: <https://sites.google.com/view/classroom-workspace/?pli=1> (дата звернення: 02.06.2025).

latonus. URL: <https://asu-edu.kz/> (дата звернення: 17.05.2025).

ikipedia: Student information system. URL:
https://en.wikipedia.org/wiki/Student_information_system (дата звернення:

ікіпедія: Інформаційна система. URL:
https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0 (дата звернення:

ikipedia: Unified Modeling Language. URL:
https://uk.wikipedia.org/wiki/Unified_Modeling_Language (дата звернення:

rawio. URL: <https://www.drawio.com/blog/uml-class-diagrams> (дата звернення:

ucid: The world's first work acceleration platform. URL:
https://lucid.co/?_gl=1%2A11bgivy%2A_ga%2AOTUzMjcxOTQ0LjE3MzMxNDI1NzQ.%2A_ga_MPV5H3XMB5%2AczE3NDg5MzcyNDYkbzYkZzAkdDE3NDg5MzcyNDYkajYwJGwwJGgw (дата звернення: 22.05.2025).

ікіпедія: Прикладний програмний інтерфейс. URL:
https://uk.wikipedia.org/wiki/%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%B8%D0%B9_%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81%D1%82%D0%B5%D0%BC%D0%B0
(дата звернення: 15.05.2025).

ентр підготовки IT фахівців: HTTP-запит (HTTP request). URL:

<https://qalight.ua/baza-znaniy/http-zapyt-http-request/> (дата звернення:

ikipedia: JSON. URL: <https://en.wikipedia.org/wiki/JSON> (дата звернення:

ostgreSQL: The World's Most Advanced Open Source Relational Database. URL: <https://www.postgresql.org/> (дата звернення: 25.05.2025).

ікіпедія: Скрам. URL: <https://uk.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B0%D0%BC> (дата звернення: 18.05.2025).

ікіпедія: BPMN. URL: <https://uk.wikipedia.org/wiki/BPMN> (дата звернення:

ікіпедія: Система управління навчанням. URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F%D0%BC (дата звернення: 24.05.2025).

ікіпедія: EOM. URL: [https://uk.wikipedia.org/wiki/%D0%95%D0%9E%D0%9C_\(%D1%82%D0%B5%D1%85%D0%BD%D1%96%D0%BA%D0%B0\)](https://uk.wikipedia.org/wiki/%D0%95%D0%9E%D0%9C_(%D1%82%D0%B5%D1%85%D0%BD%D1%96%D0%BA%D0%B0)) (дата звернення: 17.05.2025).

ikipedia: Unified Modeling Language. URL: https://uk.wikipedia.org/wiki/Unified_Modeling_Language (дата звернення:

ode JS. URL: <https://nodejs.org/en> (дата звернення: 10.05.2025).

east: The library for web and native user interfaces. URL: <https://react.dev/> (дата звернення: 10.05.2025).

урінний С. Розробка вебсайтів для початківців: HTML – CSS – JavaScript. 2022. 104 с. (дата звернення: 07.05.2025).

улеса О.Ю. Основи HTML та CSS. Лабораторний практикум. Ужгород, 2019. 53 с. (дата звернення: 07.05.2025).

enterprise Architect User Guide. URL:
<https://sparxsystems.com/bin/EAUserGuide.pdf> (дата звернення: 01.06.2025) .
ML modeling tools for Business, Software, Systems. URL:
<https://sparxsystems.com/> (дата звернення: 28.05.2025).
pizza Day. URL: <https://pizzaday.ua/> (дата звернення: 26.05.2025).
dominos Pizza. URL: <https://dominos.ua/> (дата звернення: 26.05.2025).
terraMare Pizza. URL: <https://terramare.vn.ua/> (дата звернення: 26.05.2025).

s

елюбов В. О., Білак Ю. Ю., Реблян А. М. Проектування реляційних баз даних.
Ужгород : ДВНЗ «УжНУ», 2023. 60 с. (дата звернення: 10.05.2025).
Єгеда І. Проектування та використання баз даних. Київ : КПІ ім. Ігоря
Єікорського, 2021. 49 с. (дата звернення: 10.05.2025).
ятник Н. В. Проектування баз і сховищ даних. Київ : КНЕУ, 2004. 348 с. (дата
звернення: 10.05.2025).

D

i

a

g

r

a

m

s

[

E

л

е

к

ДОДАТКИ

Додаток А

Фрагмент коду створення БД:

-- Table: Користувачі (Users)

р

о

Користувачі (

-- Table: Категорії Страв (Dish Categories)

CREATE TABLE Категорії_Страв (

-- Table: Страви (Dishes)

CREATE TABLE Страви (

goryID INTEGER REFERENCES Категорії_Страв(ID), -- Foreign key to Категорії_Страв table,

-- Table: Замовлення (Orders)

CREATE TABLE Замовлення (

UserID INTEGER REFERENCES Користувачі(ID) NOT NULL, -- Foreign key to Користувачі

-- Table: Позиції Замовлення (Order Items/Order Details) - this table is implicitly required to link Замовлення and Страви

-- The image shows 'Позиції Замовлення' with attributes 'ID', 'OrderID', 'DishID', 'Quantity',

CREATE TABLE Позиції_Замовлення (

OrderID INTEGER REFERENCES Замовлення(ID) ON DELETE CASCADE NOT NULL, --
Foreign key to Замовлення table, ON DELETE CASCADE means if an order is deleted, its items

DishID INTEGER REFERENCES Страви(ID) ON DELETE RESTRICT NOT NULL, --
Foreign key to Страви table, ON DELETE RESTRICT means cannot delete a dish if it's part of an

Додаток Б

Екранні форми створеного сайту

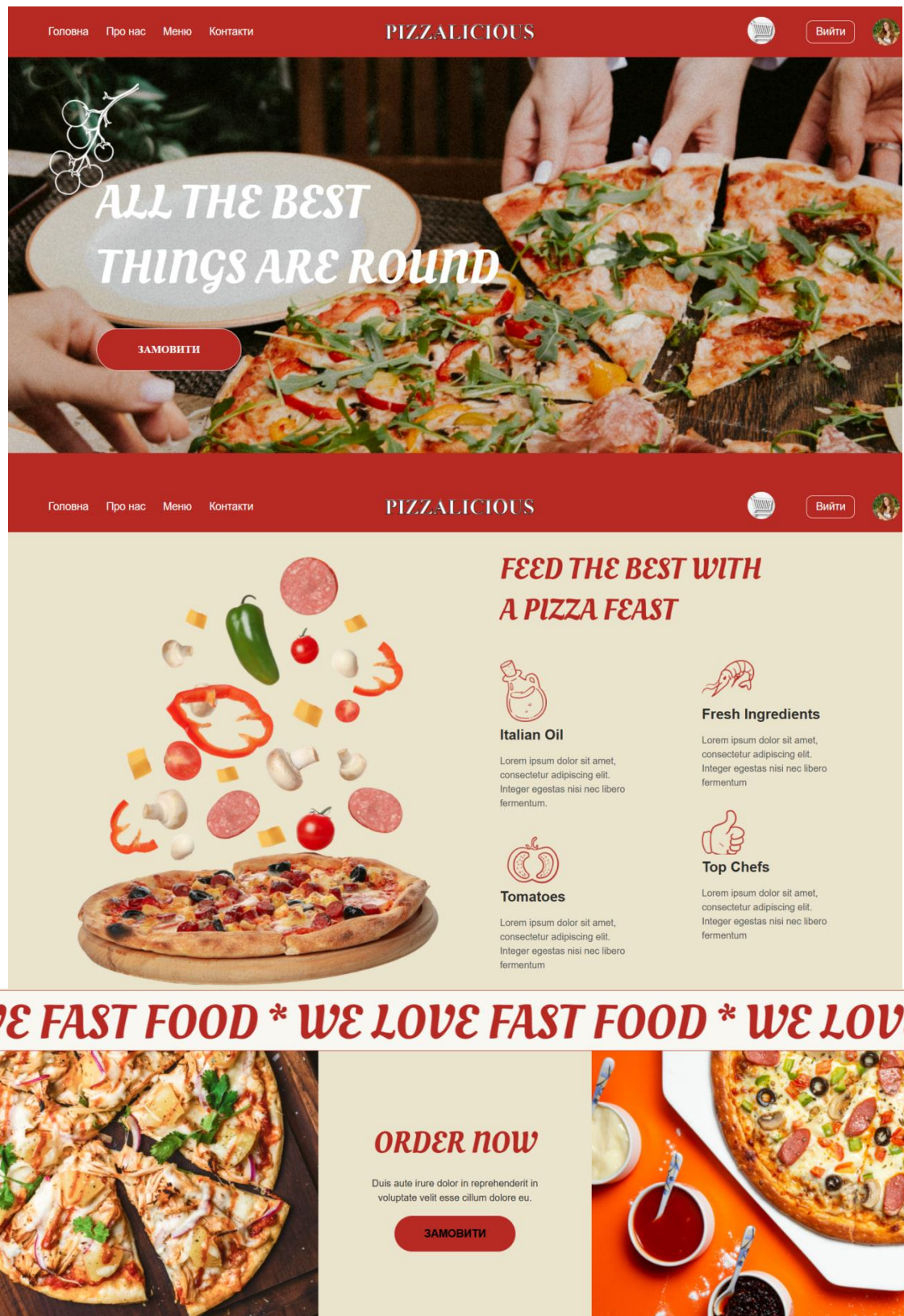


Рисунок Б.1 – Повне представлення головної сторінки

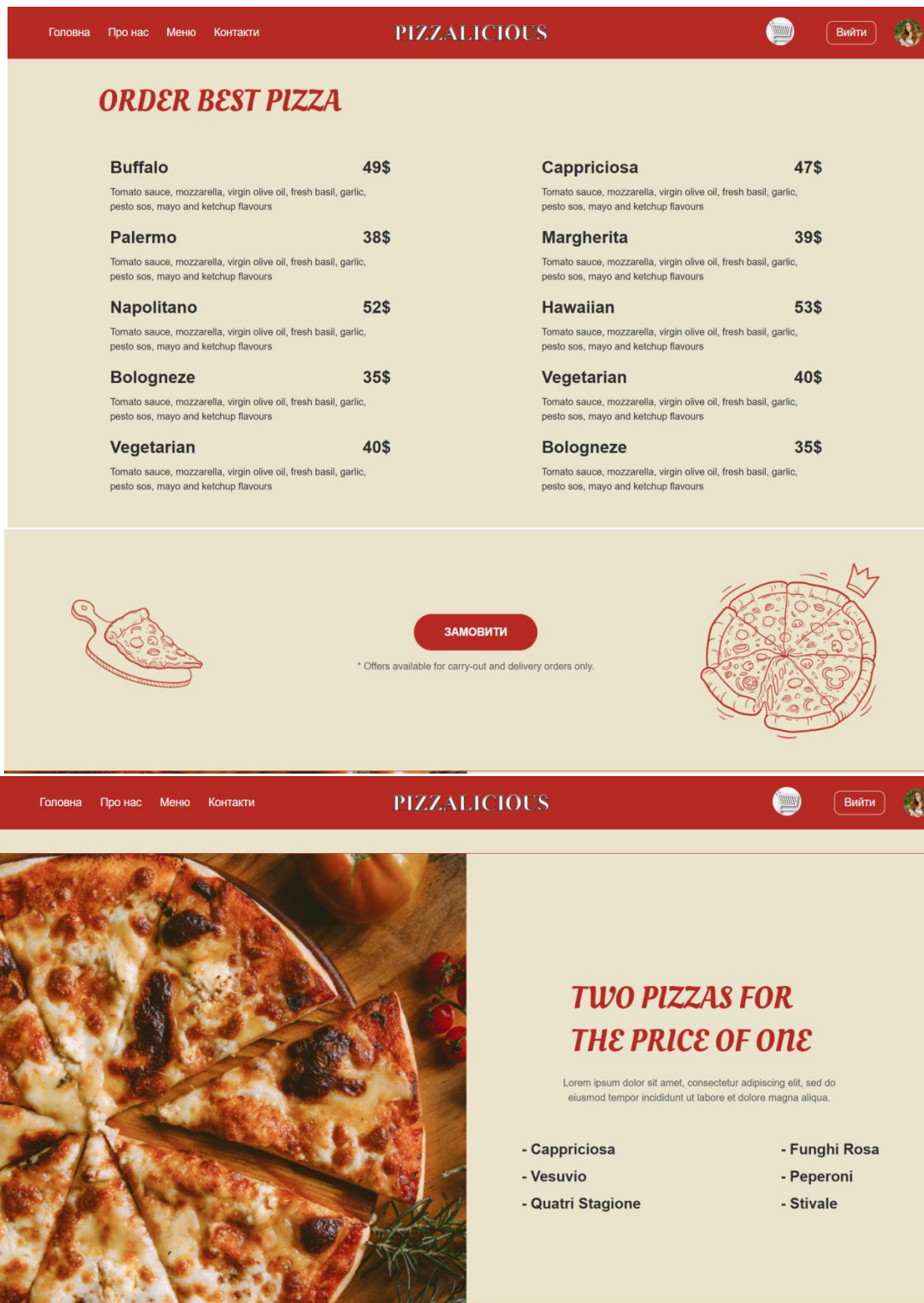


Рисунок Б.2 – Повне представлення головної сторінки (продовження)

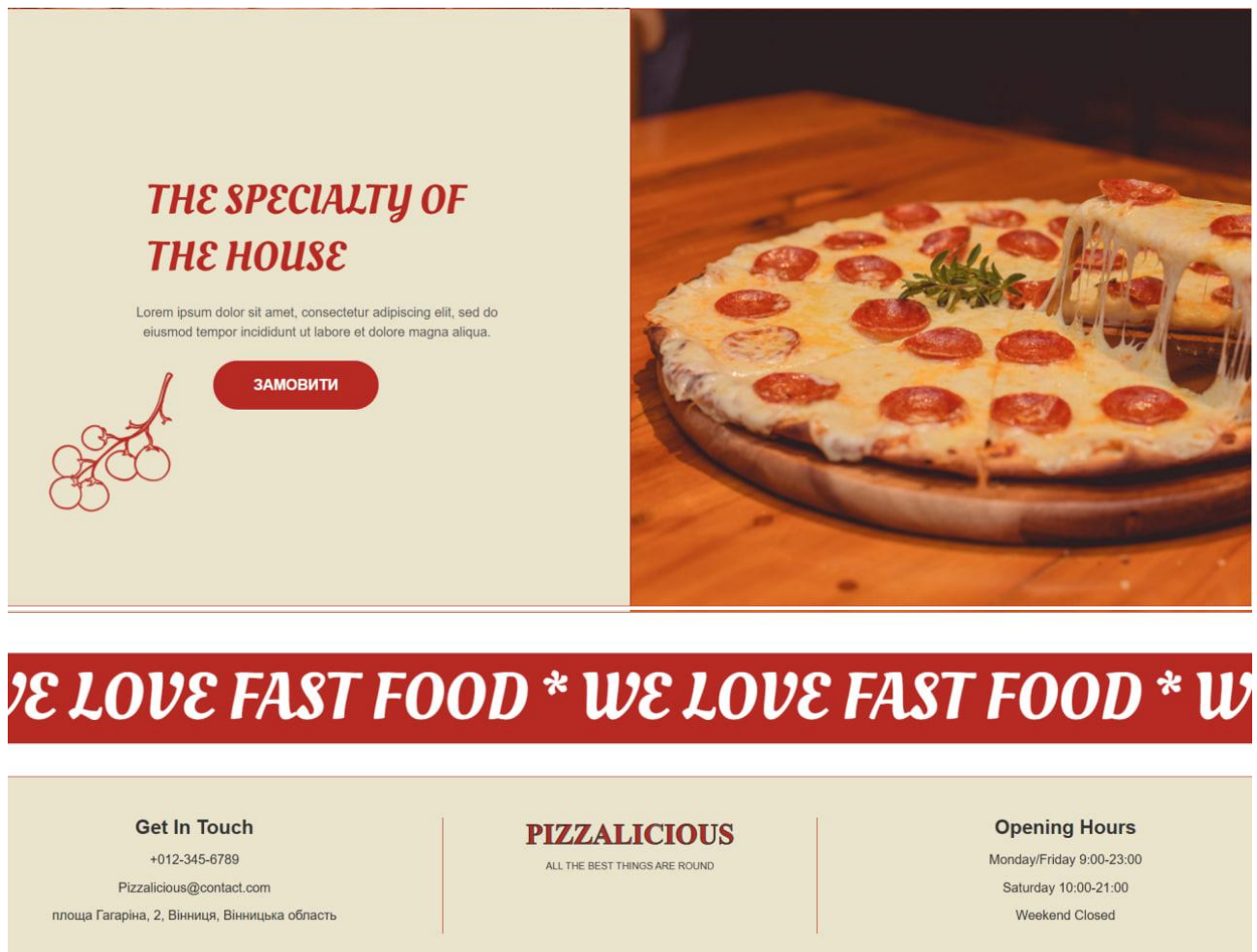


Рисунок Б.3 - Повне представлення головної сторінки (продовження)

Екранні форми створеного сайту

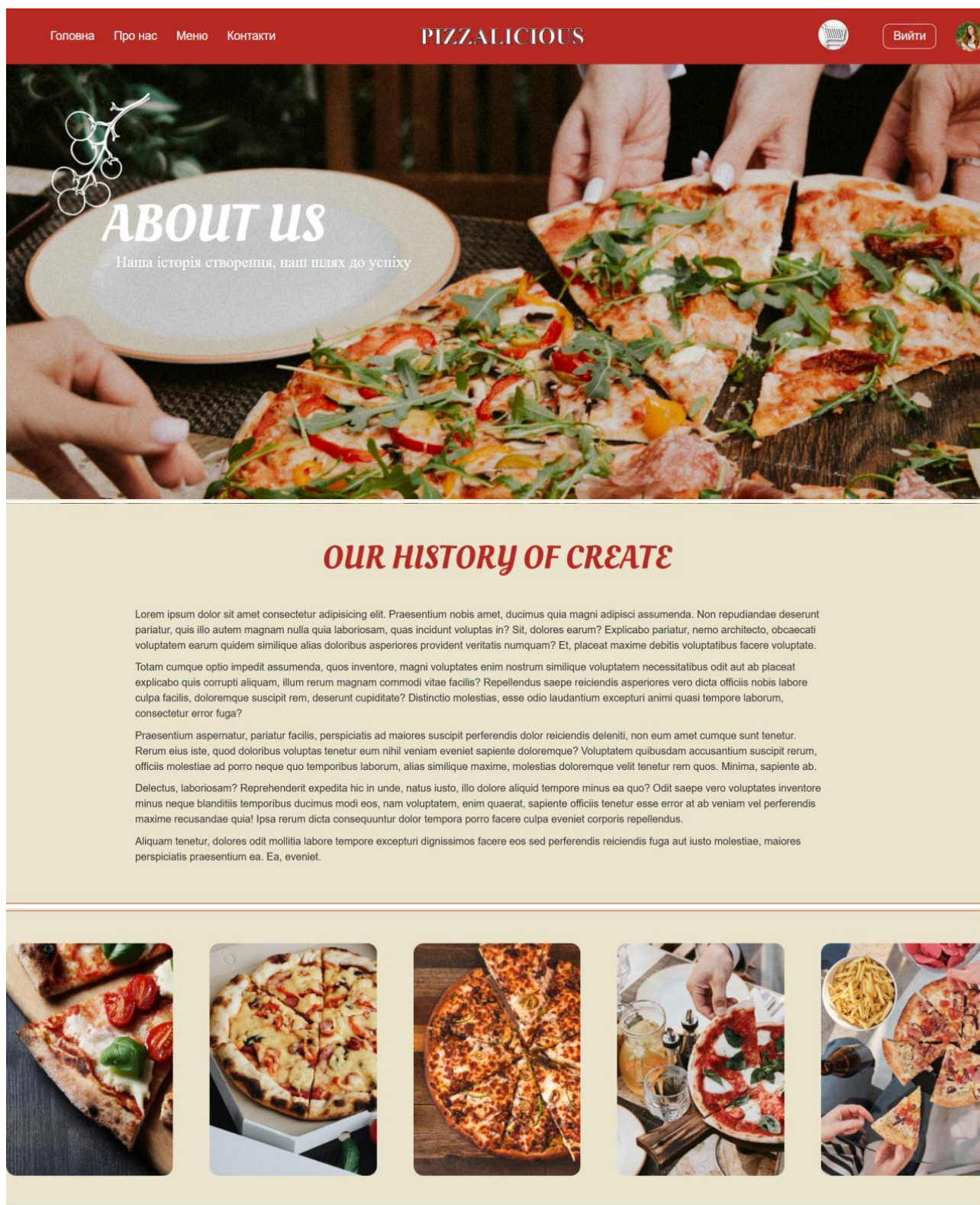


Рисунок В.1 - Повне представлення сторінки «Про нас»

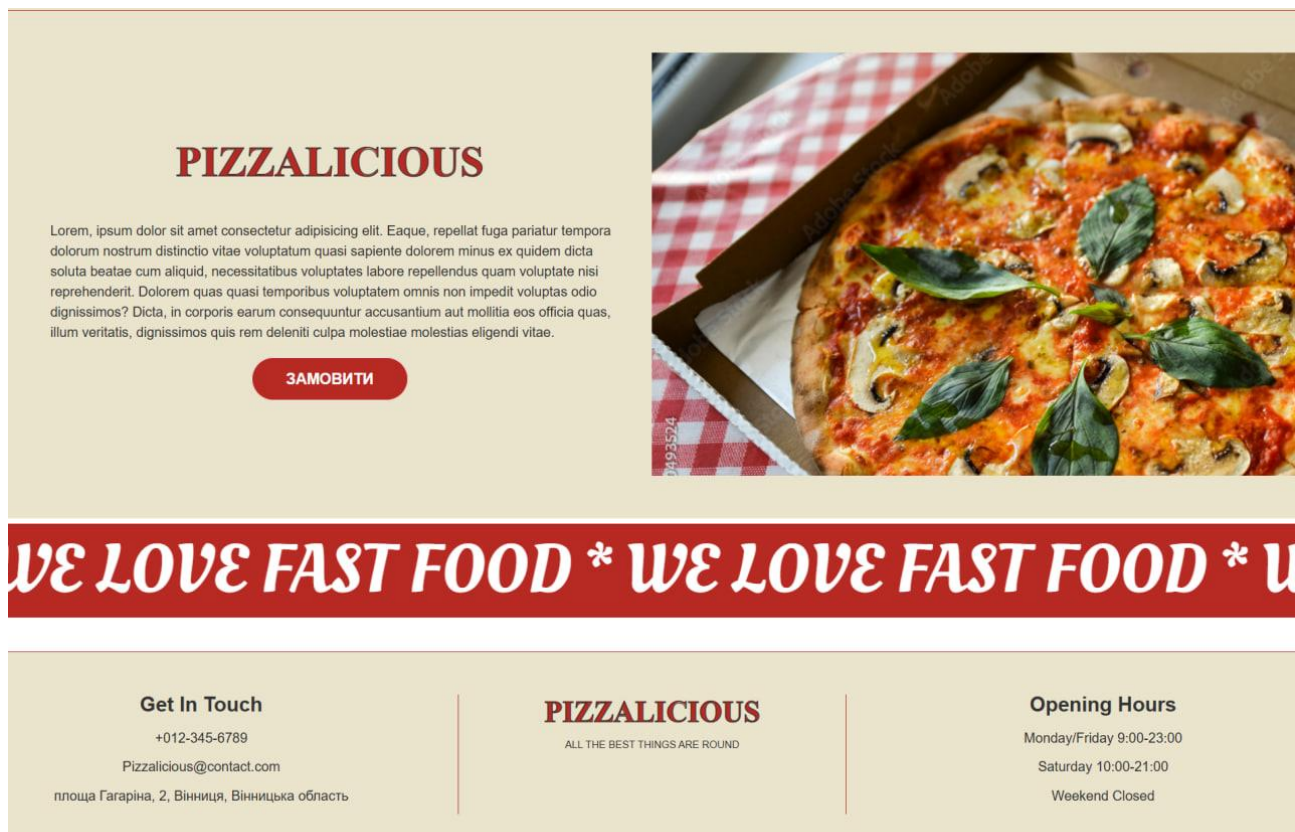


Рисунок В.2 - Повне представлення сторінки «Про нас»(продовження)

Екранні форми створеного сайту



Рисунок Г.1 - Більш повне представлення сторінки меню

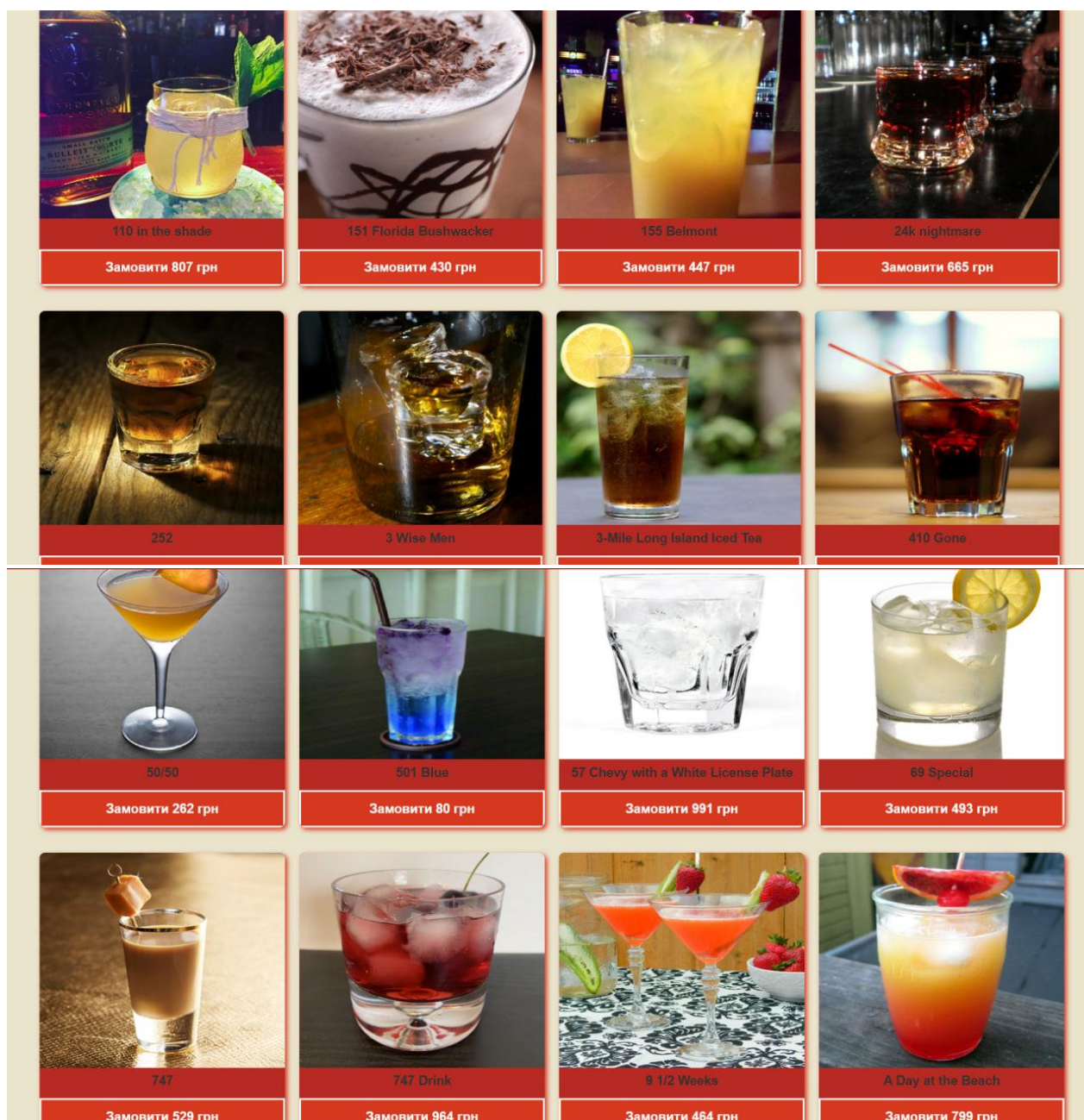


Рисунок Г.2 - Більш повне представлення сторінки меню (продовження)

Екранні форми створеного сайту

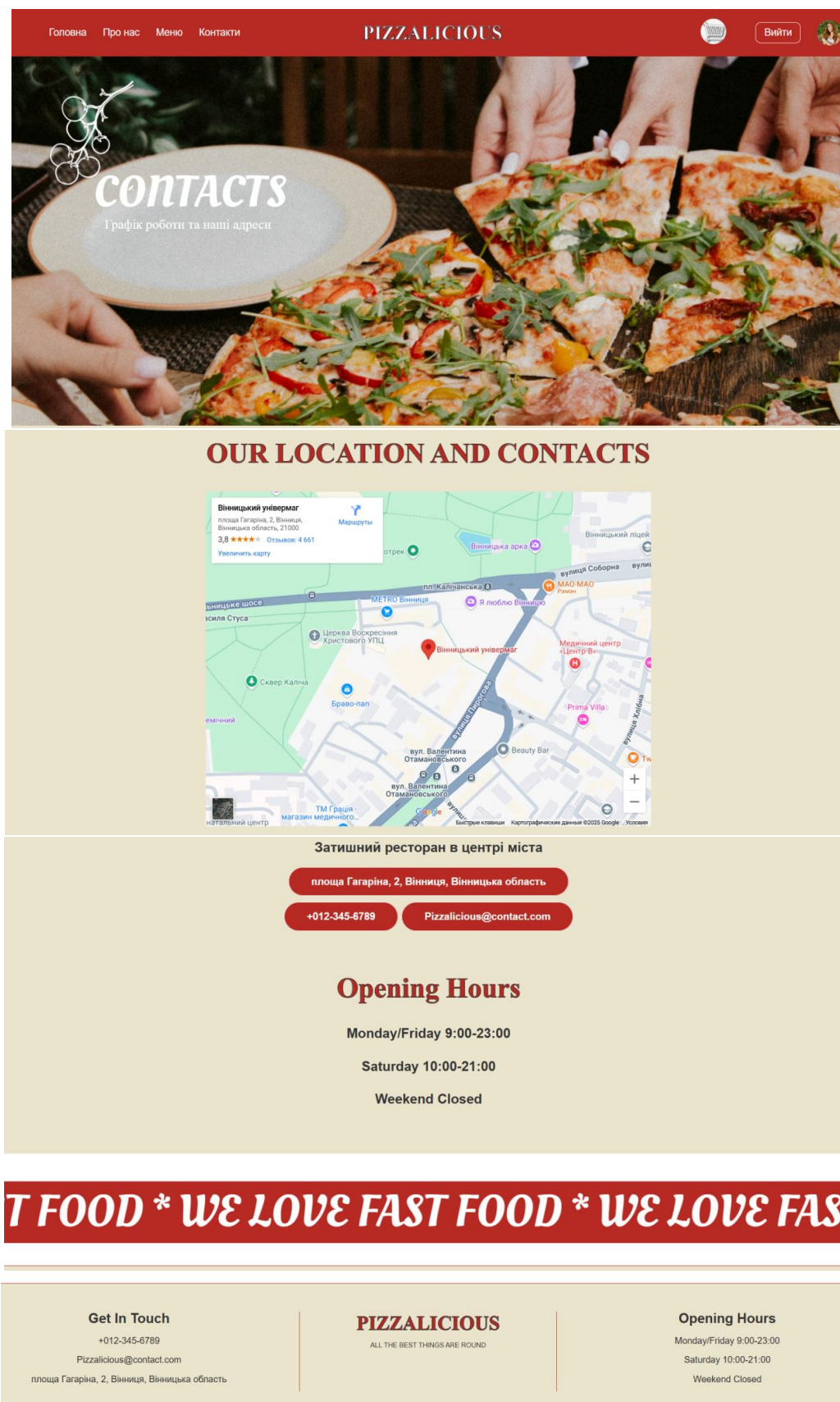


Рисунок Д.1 - Повне представлення сторінки «Контакти»

Фрагмент коду додавання товарів в корзину

```
import { goodsData } from "../getData.js";
export let saveData = [];
document.addEventListener('click', (e)=>{
    if(e.target.classList.contains('buyBtn')){
        addToCart(e);
    }
})
function addToCart(e){
    const priceTarget = e.target.dataset.price;
    const idTarget = e.target.dataset.id;
    const [drinks] = goodsData;
    const currentCocktail = drinks.find((data) => data.idDrink === idTarget);
    saveData.push(currentCocktail);
    addDataToLocalStorage();
    goodHtml();
}
function addDataToLocalStorage (data){
    localStorage.setItem('cartData', JSON.stringify(saveData));
}
function getDataFromLocalStorage(){
    return JSON.parse(localStorage.getItem('cartData'));
}
export function goodHtml (){
    const renderData = getDataFromLocalStorage();
    const table = document.querySelector('.cartTable');
    console.log(table);
    if(renderData){
        const renderGoods = renderData?.map(({ price, count, strDrinkThumb, strDrink }) => {
            let strSum = price * count;
            return`<tr class="goodList">
                <td></td>
                <td class="title">${strDrink} <span> </span></td>
                <td class="count"><span class="minus"></span>${count} шт <span class="plus"></span></td>
                <td class="price"> ${price} грн</td>
                <td class="sum"> ${strSum} грн</td>
            </tr>`
        }).join("")
        table.innerHTML = renderGoods;
    } else{
        table.innerHTML = " Ваш кошик порожній, перейдіть до меню щоб обрати.";
    }
}
```

Фрагмент коду відкривання та закривання корзини як модального вікна

```
import { goodHtml } from "./addGood.js";
document.addEventListener('DOMContentLoaded', (e)=>{
  const backdrop = document.querySelector('.cart-container');
  const openCart = document.querySelector('.cart-icon');
  const closeCart = document.querySelector('[data-action="close-modal"]');
  const cartModal = document.querySelector('.cartModal');
  const makeOrder = document.querySelector('.makeOrder');
  openCart.addEventListener('click', onOpenModal);
  closeCart.addEventListener('click', onCloseModal);
  backdrop.addEventListener('click', onBackdropClick);
  makeOrder.addEventListener('click', onCloseModal);
  function onOpenModal(){
    window.addEventListener('keydown', onEscKeyPress);
    cartModal.classList.toggle('show-modal');
    backdrop.classList.toggle('show-modal');
    goodHtml();
  }
  function onCloseModal(){
    window.removeEventListener('keydown', onEscKeyPress);
    cartModal.classList.remove('show-modal');
    backdrop.classList.remove('show-modal');
    makeOrder.classList.remove('show-modal');
  }
  function onBackdropClick(e){
    if(e.currentTarget === e.target){
      onCloseModal();
    }
  }
  function onEscKeyPress(e){
    const ESC_KEY_CODE = "Escape";
    if(ESC_KEY_CODE === e.code){
      onCloseModal();
    }
  }
});
```

Звіт подібності роботи