

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

Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра системного аналізу та кібербезпеки

| | |
|-------------------------------------|-------------------------------------|
| ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА | «СИСТЕМНИЙ АНАЛІЗ» |
| Галузь знань | 12 «Інформаційні технології» |
| Спеціальність | 124 «Системний аналіз» |

Форма навчання: очна (денна)

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему «Розробка веб-сайту онлайн курсів »
здобувача Герасименка Олександра Олександровича

(підпис)

Науковий керівник: д.е.н., професор, Галіцин В.К.

(підпис)

Робота допущена до захисту перед екзаменаційною комісією з атестації здобувачів вищої освіти (ЕК)

Завідувач кафедри: д.фіз.-мат.н., проф.Джалладова Ірада Агаверді-кизи

(підпис)

Київ 2024

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

Навчально-науковий інститут «Інститут інформаційних технологій в економіці»

Кафедра системного аналізу та кібербезпеки

**ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
СПЕЦІАЛЬНІСТЬ**

**СИСТЕМНИЙ АНАЛІЗ
124 СИСТЕМНИЙ АНАЛІЗ**

ПОГОДЖЕНО

Керівник проектної групи (гарант)
освітньо-професійної програми

_____ Галіцин В.К.
(підпис)

_____ 2024 р.

ЗАТВЕРДЖУЮ

Завідувач кафедри
д.фіз.-мат.н., проф. Джалладова Ірада
Агаверді-кизи

(підпис)

_____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувачу вищої освіти Герасименку Олександрю Олександровичу
очної (денної) форми навчання
на підготовку кваліфікаційної бакалаврської роботи
на тему: «Розробка веб-сайту онлайн курсів»

Тему затверджено наказом ректора Університету від «30» квітня 2024 р. No 725-ст..

Кваліфікаційний бакалаврська робота виконується на матеріалах дослідження предметної області та літературних джерел.

План кваліфікаційної бакалаврської роботи

| | |
|-----------------|---|
| Розділ 1 | Характеристика та аналіз предметної області |
| (назва розділу) | |

| | |
|---|---|
| Розділ 2 | Розробка вимог і моделювання інформаційної системи |
| Розділ 3 | Проектування та реалізація комп'ютерної системи |
| Розділ 4 | Проектування системи |
| Об'єкт дослідження: | Процес розробки вебсайтів для онлайн-освіти. |
| Предмет дослідження: | Методологія системного аналізу, що застосовується для розробки вебсайту продажу онлайн курсів. |
| Мета кваліфікаційної бакалаврської роботи: | Розробити ефективний та функціональний вебсайт для продажу онлайн курсів, провести системний аналіз його компонентів та процесів, забезпечити зручність користування для користувачів і адміністраторів, а також дослідити методи оптимізації та підвищення продуктивності даної платформи. |

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

| | |
|--------------------|---|
| У розділі 1 | Провести аналіз існуючих рішень на ринку онлайн курсів, визначити основні функціональні та нефункціональні вимоги до системи, а також дослідити цільову аудиторію та її потреби. |
| У розділі 2 | Сформулювати детальні вимоги до вебсайту, створити UML-діаграми для візуалізації основних компонентів системи, таких як випадки використання, діаграми класів та діаграми послідовностей, а також провести моделювання бізнес-процесів за допомогою BPMN діаграм. |
| У розділі 3 | описується процес проектування та реалізації компонентів системи. Для цього використовуються різні методи та інструменти, такі як UML-діаграми, які допомагають візуалізувати структуру та взаємодію між компонентами системи |
| У розділі 4 | Архітектура системи визначає основні компоненти системи, їх взаємозв'язки та способи взаємодії. Вона включає серверну частину, клієнтську частину, базу даних та мережеву інфраструктуру. |

**Завдання підготував
науковий керівник**

В.К.Галіцин

(підпис)

(ініціали, прізвище)

« ____ » _____ 2024 р.

**Завдання одержав
здобувач**

О.О.Герасименко

(підпис)

(ініціали, прізвище)

« ____ » _____ 2024 р.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота містить 51 сторінку тексту, 3 таблиці, 7 рисунків, список використаних джерел з 30 найменувань, додатки.

«Розробка веб-сайту онлайн курсів»

Об'єктом дослідження: Процес розробки вебсайтів для онлайн-освіти.

Предметом дослідження: Методологія системного аналізу, що застосовується для розробки вебсайту продажу онлайн курсів.

Мета та завдання виконання дослідження: Розробити ефективний та функціональний вебсайт для продажу онлайн курсів, провести системний аналіз його компонентів та процесів, забезпечити зручність користування для користувачів і адміністраторів, а також дослідити методи оптимізації та підвищення продуктивності даної платформи.

Відповідно до поставленої мети були визначені такі завдання:

- Провести аналіз існуючих рішень на ринку онлайн курсів.
- Пизначити основні функціональні та нефункціональні вимоги до системи.
- Дослідити цільову аудиторію та її потреби.
- Сформулювати детальні вимоги до вебсайту.
- Провести тестування системи для виявлення та усунення можливих помилок.
- Реалізувати функціональність вебсайту згідно з вимогами, включаючи модулі для управління курсами, користувачами та платежами.
- Оптимізувати продуктивність системи та забезпечити її безпеку.

Методи дослідження: статистичний аналіз, прогнозне моделювання з використанням математичних моделей, поєднання економіко-математичних методів та методів кількісної оцінки прибутковості діяльності.

Теоретична, методична та практична значущість отриманих результатів: під час дослідження було розглянуто поняття ефективності в діяльності підприємства. Обрана методика оцінки показників підприємств є бути

інструментом для аналізу та оцінки фінансової стабільності підприємств у різних галузях. Побудована прогнозна модель формування прибутку від реалізації продукції дозволяє прогнозувати фінансові результати на майбутні періоди. Практичні результати дослідження полягають у можливості ефективніше використовувати фінансові ресурси, планувати подальші дії щодо підвищення конкурентоспроможності та стабільності на ринку.

Рік виконання кваліфікаційної бакалаврської роботи – 2024.

Рік захисту роботи – 2024. *Ключові слова: вебсайт, розробка, діяльність, архітектура системи, прогноз, прибуток.*

В і д г у к
про кваліфікаційну бакалаврську роботу
здобувача освітньо-професійної програми «Системний аналіз»
навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
Герасименка Олександра Олександровича
На тему: «Розробка веб-сайту онлайн курсів».

1. Актуальність теми: Кваліфікаційна бакалаврська робота присвячена застосуванню засобів і методів системного аналізу для розробки інформаційних систем у сфері онлайн-освіти.
2. Позитивні риси кваліфікаційної бакалаврської роботи: Кваліфікаційна бакалаврська робота є самостійним комплексним дослідженням.
3. Наявність самостійних розробок автора: Використання фронтенд та бекенд фреймворків, таких як React, Node.js та Django, дозволило створити модульну та легко підтримувану систему.
4. Цінність теоретичних висновків та практичних рекомендацій: Результати дослідження обумовлюють практичну значущість роботи, що визначається у можливості використання запропонованого автором проекту в різноманітних проектах досліджуваної галузі.
5. Наявність недоліків: Виконане дослідження доцільно доповнити моделлю оцінки ефективності запропонованого проекту.
6. Загальна оцінка кваліфікаційної бакалаврської роботи та її допущення до захисту перед ЕК: Тема роботи розкрита, основні положення та висновки достатньо обґрунтовані. Робота Герасименка Олександра Олександровича на тему «Розробка веб-сайту онлайн курсів» може бути допущена до захисту перед ЕК і заслуговує позитивної оцінки.

Науковий керівник професор кафедри системного аналізу та кібербезпеки д.е.н. професор

(посада, учене звання, науковий ступінь)

(підпис)

Галицин В.К.

(прізвище, ініціали)

Рецензія

на кваліфікаційну бакалаврську роботу
здобувача вищої освіти
Герасименка Олександра Олександровича

Тема «Розробка веб-сайту онлайн курсів»

Актуальність теми кваліфікаційної роботи і доцільність її розроблення обумовлені необхідністю забезпечення доступності якісної освіти через онлайн-платформи, що дозволяє зручно і ефективно управляти освітніми матеріалами, курсами, користувачами та фінансовими транзакціями. Використання сучасних методів системного аналізу та веб-розробки забезпечить високий рівень продуктивності та надійності розробленого вебсайту.

Якість проведеного дослідження. Методи системного аналізу, програмної інженерії, баз даних, а також сучасні технології веб-розробки, такі як HTML, CSS, JavaScript та різноманітні фреймворки, зумовили високу якість проведеного дослідження. Використання UML-діаграм та BPMN для моделювання системи підтверджує систематичний та структурований підхід автора.

Позитивні риси кваліфікаційної бакалаврської роботи. Робота виконана якісно. Дослідження проведені комплексно та об'єктивно, що дозволило детально охарактеризувати реальний стан проблеми та потреби користувачів. Усі аналітичні дані і висновки добре структуровані та ілюстровані. Для здійснення аналізу кваліфіковано використаний системний аналіз та методи веб-розробки на основі сучасних наукових методик та технологій.

Зауваження. Доцільно було б більш детально розглянути питання безпеки даних та захисту користувацької інформації, а також розглянути можливість інтеграції з іншими освітніми платформами для розширення функціональності системи.

Практична значимість висновків і рекомендацій. Запропонована автором архітектура та функціональна схема вебсайту для продажу онлайн курсів мають значне практичне значення для освітніх установ та приватних навчальних

платформ. Моделювання системи на основі сучасних технологій веб-розробки дозволяє створити ефективну та зручну для користувачів платформу, що сприятиме підвищенню якості освітніх послуг та доступу до них. Математична модель оптимізації продуктивності та безпеки вебсайту дозволяє забезпечити надійну роботу системи та захист даних користувачів.

Директор ТОВ «Бюро рекламних
технологій»

_____ Вілена КАСЬЯНОВА

ЗМІСТ

| | |
|--|-----------|
| ВСТУП | 4 |
| РОЗДІЛ 1. | 6 |
| ТЕОРИТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБСАЙТІВ ТА СИСТЕМНОГО АНАЛІЗУ | 6 |
| 1.1 ПОНЯТТЯ ТА КЛАСИФІКАЦІЯ ВЕБСАЙТІВ..... | 6 |
| 1.2 ОСНОВИ СИСТЕМНОГО АНАЛІЗУ | 9 |
| 1.2.1 <i>Модель системного аналізу</i> | 11 |
| 1.2.2 <i>Важливість системного аналізу в розробці інформаційних систем</i> | 12 |
| РОЗДІЛ 2. | 15 |
| ВИЗНАЧЕННЯ В СИСТЕМИ | 15 |
| 2.1 МЕТОДОЛОГІЯ ЗБОРУ ВИМОГ | 15 |
| 2.1.1 <i>Функціональні вимоги</i> | 16 |
| 2.1.2 <i>Нефункціональні вимоги</i> | 17 |
| 2.2 БІЗНЕС-ВИМОГИ | 20 |
| 2.3 ЦІЛЬОВА АУДИТОРІЯ..... | 22 |
| 2.3.1 <i>Конкурентні переваги</i> | 25 |
| 2.3.2 <i>Фінансові показники</i> | 26 |
| 2.3.3 <i>Доступ до куплених курсів та додаткових матеріалів</i> | 28 |
| РОЗДІЛ 3. | 31 |
| ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ | 31 |
| КОМПОНЕНТІВ СИСТЕМИ | 31 |
| 3.1 ВИКОРИСТАННЯ UML-ДІАГРАМ..... | 31 |
| 3.2 ДІАГРАМА КЛАСІВ..... | 32 |
| 3.3 ДІАГРАМА ПОСЛІДОВНОСТІ..... | 34 |
| РОЗДІЛ 4 | 36 |
| ПРОЕКТУВАННЯ СИСТЕМИ | 36 |
| 4.1 АРХІТЕКТУРА СИСТЕМИ | 36 |
| 4.2 ДИЗАЙН ІНТЕРФЕЙСІ ВИКЛАДАЧА | 37 |
| ВИСНОВКИ | 48 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 50 |
| ДОДАТКИ | 53 |

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

| | |
|------|--|
| БД | База Даних |
| ІС | Інформаційна система |
| КС | Комп'ютерна система |
| BSC | Binance Smart Chain |
| ОС | Операційна система |
| API | Application Programming Interface |
| СКБД | Система керування базами даних |
| UML | Unified Modeling Language |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |
| SSMS | Microsoft SQL Server Management Studio |
| BDD | Block definition diagrams |
| IBD | Internal Block Diagram |
| MVC | Model View Controller |
| RAD | Rapid Application Development |
| SQL | Structured Query Language |

ВСТУП

Актуальність теми: В сучасних умовах розвитку інформаційних технологій і глобалізації знань онлайн-освіта набуває все більшого значення. Інтернет надає можливість доступу до різноманітних навчальних ресурсів з будь-якої точки світу, що робить процес навчання більш гнучким і доступним. Вебсайти для продажу онлайн курсів стали ефективним інструментом для надання освітніх послуг, дозволяючи користувачам отримувати нові знання та навички без відриву від основної діяльності. Це особливо актуально в умовах пандемії, коли дистанційне навчання стало необхідністю.

Аналіз останніх досліджень і публікацій: Аналіз існуючих рішень на ринку онлайн курсів показує, що найбільш успішні платформи поєднують в собі зручний інтерфейс, високу якість контенту та гнучкі умови навчання. Успішні освітні платформи, такі як Coursera, Udemy та edX, пропонують широкий вибір курсів з різних галузей знань, інтерактивні методики навчання та можливість отримання сертифікатів. Сучасні підходи до розробки таких платформ включають використання системного аналізу для оптимізації функціональних і нефункціональних вимог, що забезпечує високу продуктивність і зручність використання.

Мета і завдання дослідження: Метою даної дипломної роботи є розробка вебсайту для продажу онлайн курсів з використанням системного аналізу для забезпечення його ефективності та відповідності потребам користувачів. Для досягнення цієї мети необхідно вирішити наступні завдання: провести аналіз існуючих рішень та визначити ключові вимоги до системи; здійснити системний аналіз для моделювання архітектури вебсайту; розробити дизайн інтерфейсу користувача та структуру бази даних; реалізувати вебсайт з використанням сучасних технологій та інструментів розробки; провести тестування функціональності, продуктивності та безпеки системи; оцінити результати розробки та визначити перспективи подальшого розвитку системи.

Об'єкт дослідження: Процес розробки вебсайтів для онлайн-освіти.

Предмет дослідження: Методологія системного аналізу, що застосовується для розробки вебсайту продажу онлайн курсів.

Методи дослідження: методи системного аналізу соціально-економічних об'єктів; методи економіко-математичного моделювання; логіко-ймовірнісні методи; методи ризик-менеджменту; методи теорії нечітких множин; методи штучного інтелекту та теорії прийняття рішень в умовах невизначеності.

Теоретична, методична та практична значущість отриманих результатів: Теоретична значущість дослідження полягає у розвитку методології системного аналізу для розробки інформаційних систем у сфері онлайн-освіти. Методична значущість роботи полягає у створенні практичних рекомендацій щодо розробки вебсайтів для продажу онлайн курсів, що можуть бути використані в подальших дослідженнях та розробках. Практична значущість роботи полягає у створенні функціонального вебсайту, який забезпечує зручність користування для користувачів і адміністраторів, а також підвищує ефективність процесу навчання.

Інформаційна база дослідження: Інформаційною базою дослідження стали літературні джерела, наукові статті, інтернет-ресурси та матеріали дослідження предметної області. Використовувалися матеріали провідних освітніх платформ, а також рекомендації щодо системного аналізу та веб-розробки.

РОЗДІЛ 1.

ТЕОРИТИЧНІ ОСНОВИ РОЗРОБКИ ВЕБСАЙТІВ ТА СИСТЕМНОГО АНАЛІЗУ

1.1 Поняття та класифікація вебсайтів

Вебсайт (англ. website) — це сукупність взаємопов'язаних вебсторінок, доступних через інтернет за унікальною URL-адресою, яка зазвичай починається з доменного імені. Вебсайти можуть містити текстову, графічну, аудіо- та відеоінформацію і використовуються для різних цілей..

Класифікація вебсайтів:

Інформаційні вебсайти: Призначені для надання користувачам різноманітної інформації (новини, статті, блоги).

Корпоративні вебсайти: Сайти компаній і організацій, що містять інформацію про діяльність, продукти та послуги.

Електронна комерція (E-commerce): Вебсайти, що дозволяють здійснювати онлайн покупки товарів і послуг (інтернет-магазини).

Сайти соціальних мереж: Платформи для спілкування та обміну інформацією між користувачами (Facebook, Twitter).

Вебсайти освітніх установ: Сайти шкіл, університетів, онлайн курсів, що надають освітні матеріали та ресурси.

Розважальні вебсайти: Сайти, що пропонують розважальний контент, такий як ігри, відео, музика.

Форуми та спільноти: Платформи для обговорення різних тем і обміну досвідом серед користувачів.

Особливості вебсайтів для продажу онлайн курсів

Вебсайти для продажу онлайн курсів мають специфічні особливості, що визначають їх функціональність та структуру. Вони повинні забезпечувати

зручний доступ до навчальних матеріалів, зручність навігації та безпеку особистих даних користувачів.

Основні особливості:

Реєстрація та авторизація: Надання можливості користувачам реєструватися на сайті, створювати облікові записи та входити до системи.

Каталог курсів: Відображення списку доступних курсів з можливістю їх фільтрації та пошуку.

Деталі курсу: Сторінки з описом курсів, навчальною програмою, відгуками користувачів та демо-матеріалами.

Онлайн оплата: Інтеграція з платіжними системами для зручного проведення оплат.

Доступ до контенту: Забезпечення доступу до навчальних матеріалів після оплати курсу.

Система зворотного зв'язку: Надання можливості користувачам залишати відгуки та оцінювати курси.

Адміністративна панель: Інструменти для управління курсами, користувачами та фінансовими операціями.

Безпека: Захист особистих даних користувачів та забезпечення безпеки фінансових транзакцій.

Перший крок у класифікації вебсайтів за допомогою системного аналізу - визначення його типу. Вебсайти можуть бути інформаційними, комерційними, соціальними, розважальними, освітніми, блогами, портфоліо, форумами та іншими. Кожен тип вебсайту має свої специфічні особливості, функціональність та цільову аудиторію. Наприклад, інформаційний вебсайт надає користувачам інформацію про певну тему, комерційний вебсайт продає товари або послуги, соціальний вебсайт дозволяє користувачам спілкуватися і ділитися інформацією.

Після визначення типу вебсайту, системний аналіз досліджує його структуру, навігацію та функціональність. Це включає в себе аналіз меню, контенту, форм, пошукових систем, способу відображення інформації, інтерактивних елементів та інших важливих компонентів. Системний аналіз

допомагає зрозуміти, як вебсайт організований, як користувачі переміщуються по ньому і як він працює в цілому.

Важливою складовою класифікації є аналіз цільової аудиторії. Системний аналіз дозволяє визначити характеристики користувачів, їх потреби, мотивацію, поведінку та очікування. Ця інформація допомагає оцінити, чи вебсайт відповідає потребам своєї цільової аудиторії, чи він легкий у використанні і чи задовольняє очікування користувачів.

Наступним етапом є аналіз технологій, що використовуються на вебсайті. Системний аналіз дозволяє визначити мову програмування, фреймворки, бази даних, сервери і інші технології, що використовуються для створення та обслуговування вебсайту. Ця інформація допомагає оцінити технічну сторону вебсайту і визначити його надійність, продуктивність, безпеку та масштабованість.

На завершення системного аналізу оцінюються ризики і можливості, пов'язані з вебсайтом. Системний аналіз допомагає визначити потенційні проблеми, пов'язані з безпекою, продуктивністю, конкуренцією та іншими факторами. Також оцінюються можливості для розвитку вебсайту, наприклад, впровадження нових функцій, збільшення трафіку або розширення цільової аудиторії.

В результаті системного аналізу вебсайту формується чітке розуміння його структури, функціональності, цільової аудиторії, технологій і ризиків. Ця інформація допомагає оцінити ефективність вебсайту, визначити можливі проблеми та розробити стратегію його розвитку.

1.2 Основи системного аналізу

Системний аналіз — це методологія дослідження складних систем, що дозволяє визначити їх складові частини, структуру, взаємозв'язки та функціонування. Системний аналіз використовується для проектування, розробки, впровадження та оцінки інформаційних систем.

Основні особливості:

Реєстрація та авторизація: Надання можливості користувачам реєструватися на сайті, створювати облікові записи та входити до системи.

Каталог курсів: Відображення списку доступних курсів з можливістю їх фільтрації та пошуку.

Деталі курсу: Сторінки з описом курсів, навчальною програмою, відгуками користувачів та демо-матеріалами.

Онлайн оплата: Інтеграція з платіжними системами для зручного проведення оплат.

Доступ до контенту: Забезпечення доступу до навчальних матеріалів після оплати курсу.

Система зворотного зв'язку: Надання можливості користувачам залишати відгуки та оцінювати курси.

Адміністративна панель: Інструменти для управління курсами, користувачами та фінансовими операціями.

Безпека: Захист особистих даних користувачів та забезпечення безпеки фінансових транзакцій.

Основи Системного аналізу: На шляху до ефективних рішень

Системний аналіз - це метод дослідження та проектування складних систем, який допомагає зрозуміти, як вони функціонують, визначити їхні складові частини, взаємозв'язки та взаємодію з навколишнім середовищем. Цей метод широко застосовується в різних галузях, таких як інформаційні технології, інженерія, бізнес, управління та інші, щоб знайти оптимальні рішення для складних проблем.

Основна мета системного аналізу полягає в тому, щоб розбити складну систему на більш прості компоненти, дослідити їхню взаємодію та взаємозв'язки, а потім використовувати цю інформацію для розробки ефективних рішень. Він фокусується на розумінні цілісної картини, а не тільки на окремих елементах системи.

Ключовим елементом системного аналізу є збір та аналіз інформації. Це може включати інтерв'ю з користувачами, дослідження існуючих документів, спостереження за функціонуванням системи, аналіз статистичних даних та ін.

Після збору інформації її потрібно структурувати та проаналізувати, щоб зрозуміти взаємозв'язки між різними елементами системи. Для цього використовуються різноманітні методи і інструменти, такі як діаграми потоків даних, діаграми класів, діаграми Use Case, матриці вимог та інші.

Важливим етапом системного аналізу є виявлення та аналіз ризиків. Це можуть бути технічні проблеми, бізнес-проблеми, проблеми з безпекою або проблеми з доступністю системи.

Виходячи з проведеного аналізу, формулюються висновки та рекомендації щодо покращення системи. Це може включати розробку нових функцій, перегляд існуючих процесів, впровадження нових технологій або зміну організаційної структури.

Системний аналіз не є статичним процесом. Він постійно розвивається і вдосконалюється, враховуючи нові технології та вимоги користувачів.

Ключові принципи системного аналізу: системний підхід: розглядає проблему в контексті всієї системи, а не тільки окремих її частин.

Орієнтація на користувача: враховує потреби та вимоги користувачів системи.

Покроковий аналіз: розбиває складні завдання на менші, більш керовані кроки.

Ітеративність: дозволяє коригувати процес аналізу та розробки на основі отриманих результатів.

Документація: фіксує все, що було зроблено, щоб забезпечити прозорість і можливість повторного використання отриманої інформації.

Перевага системного аналізу: збільшення ефективності: дозволяє оптимізувати роботу системи та зменшити витрати.

Покращення якості: дозволяє створити систему, яка відповідає потребам користувачів та вимогам бізнесу.

Зменшення ризиків: дозволяє виявити та усунути потенційні проблеми на ранніх етапах розробки.

Забезпечення прозорості: дозволяє зрозуміти, як працює система, і надає можливість внести зміни в її роботу.

Системний аналіз є цінним інструментом для розробки та вдосконалення складних систем. Він допомагає знайти оптимальні рішення для різних проблем, підвищити ефективність та якість роботи системи, а також зменшити ризики.

1.2.1 Модель системного аналізу

Системний аналіз — це методологія дослідження складних систем, що дозволяє визначити їх складові частини, структуру, взаємозв'язки та функціонування. Системний аналіз використовується для проектування, розробки, впровадження та оцінки інформаційних систем.

Основні етапи системного аналізу:

Визначення вимог: Збір і аналіз вимог користувачів до системи, включаючи функціональні та нефункціональні вимоги.

Моделювання системи: Створення моделей, що відображають структуру та функціонування системи (UML-діаграми, DFD, ER-діаграми).

Проектування: Розробка архітектури системи, включаючи базу даних, інтерфейс користувача та логіку бізнес-процесів.

Реалізація: Розробка програмного коду відповідно до проектної документації.

Тестування: Перевірка функціональності, продуктивності та безпеки системи.

Впровадження: Розгортання системи в робочому середовищі та навчання користувачів.

Оцінка та оптимізація: Аналіз роботи системи, внесення необхідних змін і вдосконалень.

1.2.2 Важливість системного аналізу в розробці інформаційних систем

Системний аналіз відіграє ключову роль у розробці інформаційних систем, оскільки забезпечує:

Чітке розуміння вимог: Допомагає виявити і задовольнити потреби користувачів, що є основою для успішної розробки системи.

Оптимізацію ресурсів: Забезпечує ефективне використання ресурсів (час, гроші, персонал) завдяки чіткому плануванню і прогнозуванню витрат.

Зменшення ризиків: Ідентифікує можливі проблеми та ризики на ранніх етапах розробки, що дозволяє їх уникнути або мінімізувати.

Підвищення якості: Забезпечує високу якість кінцевого продукту завдяки ретельному тестуванню та перевірці всіх компонентів системи.

Гнучкість та масштабованість: Дозволяє створити систему, що легко адаптується до змін і може бути розширена з урахуванням майбутніх потреб.

Ефективну комунікацію: Полегшує взаємодію між замовниками, розробниками та іншими зацікавленими сторонами через створення зрозумілої документації та моделей.

Системний аналіз є критично важливим інструментом у розробці вебсайтів для продажу онлайн курсів, оскільки дозволяє створити високоякісну, надійну та

зручну систему, що відповідає потребам користувачів та забезпечує ефективну реалізацію бізнес-цілей.

Важливість системного аналізу в розробці інформаційних систем: Глибший погляд

Системний аналіз - це не просто набір технічних інструментів, а скоріше глибокий підхід до розробки інформаційних систем, що дозволяє створити продукт, який не просто відповідає функціональним вимогам, а й є дійсно корисним, ефективним та призначеним для конкретної цільової аудиторії.

Глибше розуміння потреб:

Системний аналіз виходить за межі простого збирання вимог до системи. Він фокусується на розумінні контексту, в якому буде функціонувати система, на аналізі поведінки користувачів, їх мотивації і очікувань. Це дозволяє створити систему, яка не тільки працює, а і справді допомагає користувачам вирішувати їхні проблеми.

Виявлення прихованих ризиків:

Системний аналіз допомагає виявити потенційні ризики, які можуть бути пропущені на початкових етапах розробки. Це можуть бути технічні проблеми, проблеми з інтеграцією з іншими системами, проблеми з масштабованістю, проблеми з безпекою, а також проблеми з правовими нормативами та етичними аспектами. Вчасне виявлення та аналіз ризиків дозволяє розробити заходи для їх усунення та знизити ймовірність виникнення проблем в майбутньому.

Оптимізація ресурсів:

Системний аналіз допомагає оптимізувати використання ресурсів та зменшити витрати на розробку та експлуатацію системи. Він дозволяє визначити необхідні ресурси, вибрати оптимальні технології та забезпечити ефективну організацію роботи розробників.

Збільшення ефективності та гнучкості:

Системний аналіз допомагає створити систему, яка буде більш ефективною та гнучкою у використанні. Він дозволяє легше внести зміни в систему, додати нові функції або адаптувати систему до зміни вимог користувачів.

Підвищення задоволеності користувачів:

Завдяки ретельному системному аналізу система, що розроблена, краще відповідає потребам користувачів, що збільшує задоволеність та мотивацію користувачів до її використання.

Захист інвестицій:

Системний аналіз допомагає зменшити ризик невдалого впровадження системи та захистити інвестиції в розробку. Він дозволяє уникнути поширених помилок та створити систему, яка буде успішно функціонувати в довгостроковій перспективі.

Системний аналіз - це стратегічна інвестиція в успішне впровадження інформаційної системи. Він допомагає уникнути поширених помилок, зменшити витрати, підвищити ефективність та забезпечити успішне функціонування системи.

РОЗДІЛ 2.

ВИЗНАЧЕННЯ В СИСТЕМИ

2.1 Методологія збору вимог

Визначення вимог до системи є одним із ключових етапів у розробці інформаційних систем, включаючи вебсайти для продажу онлайн курсів. Для ефективного збору вимог використовується ряд методів та інструментів, що дозволяють максимально точно визначити потреби користувачів і бізнесу.

Основні методи збору вимог:

Опитування та анкетування: Збір інформації від потенційних користувачів шляхом проведення опитувань і анкетувань. Це дозволяє отримати велику кількість даних про очікування і потреби користувачів.

Інтерв'ю: Поглиблені бесіди з ключовими зацікавленими сторонами (стейкхолдерами), що дозволяють детально зрозуміти їхні вимоги до системи.

Аналіз існуючих систем: Вивчення та аналіз конкурентних вебсайтів та аналогічних рішень для виявлення кращих практик і унікальних функцій.

Спостереження: Наблюдання за тим, як користувачі взаємодіють з поточними системами або аналогічними продуктами, щоб виявити проблеми та потреби.

Мозковий штурм (brainstorming): Проведення сесій мозкового штурму з командою розробників, дизайнерами та іншими зацікавленими сторонами для генерації ідей і вимог.

Аналіз документів: Вивчення наявних бізнес-процесів, політик, інструкцій та іншої документації для виявлення вимог до нової системи.

Важливі аспекти методології збору вимог:

Чітке визначення цілей проекту: Що має досягти система? Які проблеми вона має вирішити?

Ідентифікація ключових користувачів: Хто буде використовувати систему? Які їхні ролі та очікування?

Визначення контексту використання: Де і як буде використовуватися система? Які зовнішні фактори впливають на її роботу?

Виявлення всіх необхідних функцій та характеристик: Які функції має мати система? Які дані вона має обробляти? Які інтерфейси вона має мати?

Забезпечення чіткого та зрозумілого документування: Вимоги має бути задокументовані в зрозумілій формі і доступні для всіх зацікавлених сторін.

Ефективна методологія збору вимог є запорукою успішної розробки інформаційної системи. Вона забезпечує чітке розуміння потреб користувачів та бізнес-вимог, що дозволяє розробити систему, яка відповідатиме очікуванням і буде ефективною в роботі.

2.1.1 Функціональні вимоги

Функціональні вимоги визначають конкретні функції та можливості системи, які повинні бути реалізовані для забезпечення її роботи. В контексті вебсайту для продажу онлайн курсів функціональні вимоги можуть включати

Реєстрація та авторизація користувачів:

- Можливість створення облікових записів.
- Вхід до системи з використанням електронної пошти та пароля.
- Відновлення пароля.

Каталог курсів:

- Перегляд списку доступних курсів.
- Пошук курсів за різними критеріями (категорія, ціна, рейтинг).
- Фільтрація та сортування курсів.

Деталі курсу:

- Відображення детальної інформації про курс (опис, програма, відгуки).

- Доступ до демо-матеріалів.

Онлайн оплата:

- Інтеграція з платіжними системами (PayPal, Stripe).
- Можливість оплати за курс.
- Генерація електронних квитанцій.

Доступ до контенту:

- Надання доступу до матеріалів курсу після оплати.
- Можливість завантаження або перегляду матеріалів онлайн.

Система зворотного зв'язку:

- Надання можливості залишати відгуки та оцінки курсів.
- Модерація відгуків.

Адміністративна панель:

- Управління курсами (додавання, редагування, видалення).
- Управління користувачами.
- Перегляд статистики продажів і користувацької активності.

2.1.2 Нефункціональні вимоги

Нефункціональні вимоги визначають критерії функціонування системи, які не стосуються безпосередньо її основних функцій, але є критично важливими для задоволення потреб користувачів та досягнення бізнес-цілей проекту.

Продуктивність:

- Швидкість завантаження сторінок не більше 3 секунд.
- Підтримка одночасного доступу не менш ніж 1000 користувачів.

Безпека:

- Захист особистих даних користувачів (використання SSL сертифікатів).

- Захист від атак типу XSS, SQL-ін'єкцій, CSRF.

Шифрування паролів та платіжної інформації:

- Зручність використання (Usability).
- Інтуїтивно зрозумілий інтерфейс.
- Адаптивний дизайн для різних пристроїв (десктоп, планшети, смартфони).

Доступність для користувачів з обмеженими можливостями.

Масштабованість:

Можливість розширення функціональності без значних змін у системі.

Підтримка додавання нових курсів і користувачів без зниження продуктивності.

Надійність:

- Висока доступність системи (99.9% uptime).
- Можливість автоматичного резервного копіювання даних.

на рис. 2.1-2.5 наведено приклади.

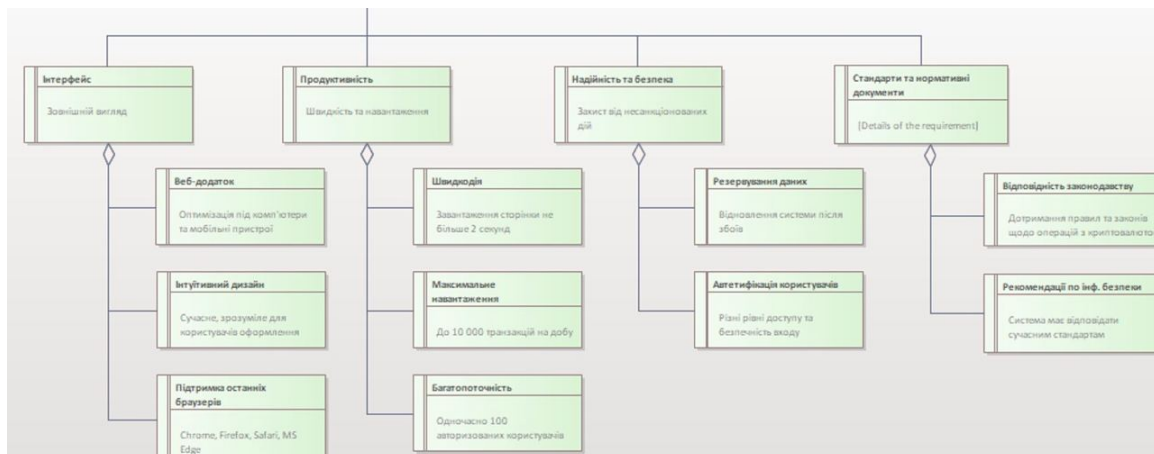


Рисунок 2.1 – Діаграма функціональних вимог.

Джерело: розроблено автором на основі [6]

| | | | | |
|---|---------------|-------------|------|--------|
| <input checked="" type="checkbox"/> Інтерфейс Зовнішній вигляд | NonFunctional | Approved | High | High |
| <input checked="" type="checkbox"/> Веб-додаток Оптимізація під комп'ютери та мобільні пристрої | NonFunctional | Implemented | High | Medium |
| <input checked="" type="checkbox"/> Інтуїтивний дизайн Сучасне, зрозуміле для користувачів оформлення | NonFunctional | Implemented | Low | Medium |
| <input checked="" type="checkbox"/> Підтримка останніх браузерів Chrome, Firefox, Safari, MS Edge | NonFunctional | Implemented | Low | Medium |

Рисунок 2.2 – Специфікація нефункціональних вимог «Інтерфейс»

Джерело: розроблено автором на основі [7]

| | | | | |
|--|---------------|-------------|--------|------|
| <input checked="" type="checkbox"/> Надійність та безпека Захист від несанкціонованих дій | NonFunctional | Approved | High | High |
| <input checked="" type="checkbox"/> Автентифікація користувачів Різні рівні доступу та безпечність входу | NonFunctional | Implemented | Medium | High |
| <input checked="" type="checkbox"/> Резервування даних Відновлення системи після збоїв | NonFunctional | Implemented | Medium | High |

Рисунок 2.3 – Специфікація нефункціональних вимог «Надійність та безпека»

Джерело: розроблено автором на основі [7]

| | | | | |
|---|---------------|-----------|--------|--------|
| <input checked="" type="checkbox"/> Продуктивність Швидкість та навантаження | NonFunctional | Validated | High | Medium |
| <input checked="" type="checkbox"/> Багатопоточність Одночасно 100 авторизованих користувачів | NonFunctional | Validated | Medium | Medium |
| <input checked="" type="checkbox"/> Максимальне навантаження До 10 000 транзакцій на добу | NonFunctional | Validated | Medium | Low |
| <input checked="" type="checkbox"/> Швидкодія Завантаження сторінки не більше 2 секунд | NonFunctional | Validated | Medium | Medium |

Рисунок 2.4 – Специфікація нефункціональних вимог «Продуктивність»

Джерело: розроблено автором на основі [7]

| | | | | |
|--|---------------|-------------|--------|------|
| <input checked="" type="checkbox"/> Стандарти та нормативні документи Відповідність до міжнародних законів та норм | NonFunctional | Proposed | Medium | High |
| <input checked="" type="checkbox"/> Відповідність законодавству Дотримання правил та законів щодо операцій з криптовалютою | NonFunctional | Implemented | Medium | High |
| <input checked="" type="checkbox"/> Рекомендації по інф. безпеки Система має відповідати сучасним стандартам | NonFunctional | Implemented | Medium | High |

Рисунок 2.5 – Специфікація нефункціональних вимог «Стандарти та нормативні документи»

Джерело: розроблено автором на основі [7]

Одним з ключових аспектів нефункціональних вимог є забезпечення належної продуктивності та масштабованості системи. Ці вимоги гарантують, що система зможе ефективно обробляти велику кількість запитів та операцій

2.2 Бізнес-вимоги

Бізнес-вимоги - це дорожня карта для успішної онлайн-платформи, яка визначає її цілі, цільову аудиторію, конкурентні переваги і бізнес-модель.

Бізнес-модель - це серце платформи, яке визначає, як вона генерує дохід і приносить прибуток. Існує декілька популярних бізнес-моделей:

- Прямий продаж курсів: Найпростіша модель, коли студенти сплачують фіксовану суму за окремі курси.
- Підписка: Студенти сплачують фіксовану щомісячну плату за доступ до всіх курсів на платформі. Це може бути вигідним для студентів, які хочуть навчитися різним речам, і для платформи, яка отримує стабільний дохід.
- Корпоративні пакети: Платформа продає пакети курсів компаніям для навчання їхніх співробітників. Це дозволяє платформі залучити більших клієнтів і отримати більший дохід.
- Партнерські програми: Платформа співпрацює з іншими освітніми платформами або організаціями, щоб розширити свою аудиторію і залучити нових

студентів. Партнери можуть отримувати відсоток від продажу курсів, які вони рекомендують.

Окрім бізнес-моделі, бізнес-вимоги визначають також цільову аудиторію і конкурентні переваги. Розуміння потреб і бажань цільової аудиторії допомагає розробити платформу, яка буде цікавою і корисною для студентів. Конкурентні переваги допомагають визначити, чим платформа відрізняється від інших і чому студенти повинні вибрати саме її на Рисунок 2.6-2.7 наведено приклади.

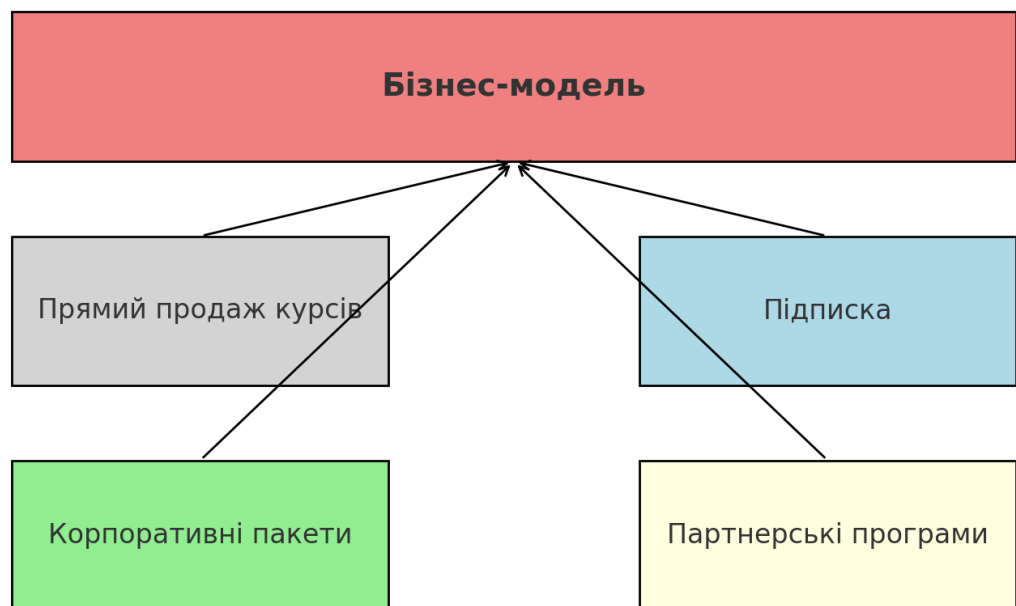


Рисунок 2.6. – Бізнес-модель

Джерело: розроблено автором на основі [10]

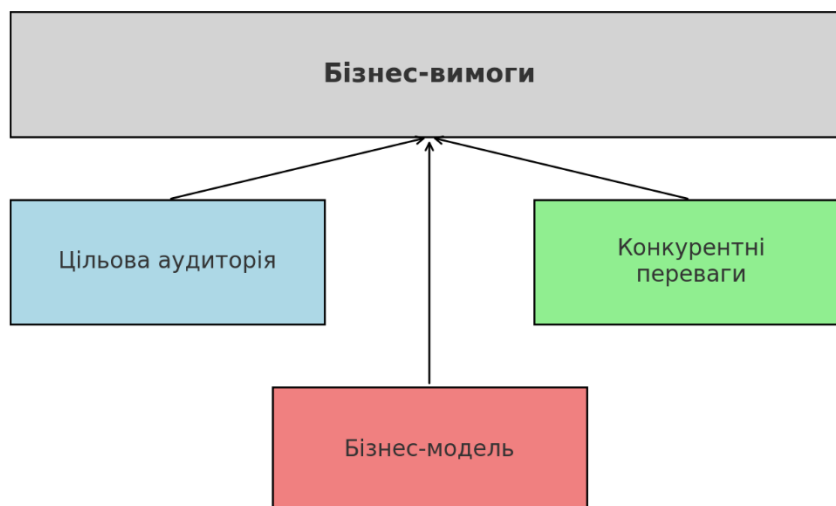


Рисунок 2.7 – Бізнес-вимоги

Джерело: розроблено автором на основі [10]

Чітко визначені бізнес-вимоги - це основа для успішного розвитку платформи онлайн-навчання. Вони дозволяють створити бізнес-план, розробити стратегію маркетингу і зробити платформу конкурентноздатною на ринку.

2.3 Цільова аудиторія

Визначення цільової аудиторії є критично важливим кроком для розробки ефективного вебсайту з продажу онлайн курсів. Знання потреб і особливостей різних груп користувачів дозволяє створити релевантний контент, адаптувати функціональність та забезпечити зручний користувацький досвід.

Основні групи цільової аудиторії:

1) Студенти та учні:

Потреби: Отримання додаткових знань та навичок, підготовка до іспитів, покращення академічних результатів.

Особливості: Висока мотивація до навчання, активне використання цифрових технологій, обмежений бюджет на навчання.

Пропозиції: Курси з академічних дисциплін, підготовка до ЗНО, SAT, АСТ, доступні ціни, гнучкі графіки навчання.

2) Професіонали:

Потреби: Підвищення кваліфікації, освоєння нових компетенцій, сертифікація, зміна професії.

Особливості: Високий рівень зайнятості, потреба в гнучких графіках навчання, орієнтація на практичні знання.

Пропозиції: Курси з професійного розвитку, сертифікаційні програми, тренінги з управління проектами, лідерства, цифрових навичок.

3) Освітні установи:

Потреби: Інтеграція онлайн курсів в навчальні програми, доступ до додаткових навчальних матеріалів.

Особливості: Зосередження на якості контенту, відповідність академічним стандартам, необхідність в адаптації матеріалів до навчальних програм.

Пропозиції: Курси, розроблені відповідно до академічних стандартів, пакети ліцензій для шкіл та університетів, інтеграція з існуючими системами управління навчанням (LMS).

4) Компанії та організації:

Потреби: Підвищення кваліфікації працівників, адаптація нових співробітників, розвиток лідерських навичок.

Особливості: Фокус на ефективності навчання, необхідність аналітики та звітності, бюджетування на корпоративне навчання.

Пропозиції: Корпоративні пакети, індивідуальні навчальні програми, аналітичні інструменти для відстеження прогресу навчання, консультації та підтримка.

Детальний розгляд цільових груп:

– Студенти та учні:

Віковий діапазон: 15-25 років.

Цілі: Підготовка до вступу до ВНЗ, покращення академічних результатів, здобуття додаткових знань поза шкільною програмою.

Переваги: Гнучкість у виборі часу навчання, доступність матеріалів на різних пристроях, можливість взаємодії з викладачами та іншими студентами через форуми та чати.

Приклади курсів: Математика, фізика, хімія, підготовка до ЗНО/ЄДІ, іноземні мови, програмування для початківців.

– Професіонали:

Віковий діапазон: 25-45 років.

Цілі: Кар'єрний ріст, зміна професії, отримання сертифікацій, розвиток нових навичок.

Переваги: Курси, які можна проходити у вільний час, матеріали, що відповідають актуальним стандартам індустрії, можливість отримати сертифікати після завершення курсу.

Приклади курсів: Управління проектами, лідерство, цифровий маркетинг, програмування, аналіз даних, фінанси та бухгалтерія.

– Освітні установи:

Типи установ: Школи, коледжі, університети, освітні центри.

Цілі: Інтеграція онлайн курсів в навчальні програми, надання додаткових ресурсів для викладачів та студентів, забезпечення доступу до сучасних навчальних матеріалів.

Переваги: Ліцензування на групу студентів, доступ до навчальних матеріалів для викладачів, підтримка інтеграції з LMS.

Приклади курсів: Загальноосвітні предмети, підготовка до міжнародних іспитів, спеціалізовані курси для підвищення кваліфікації викладачів.

4) Компанії та організації:

Типи організацій: Малі та середні підприємства, великі корпорації, державні установи.

Цілі: Підвищення ефективності та продуктивності працівників, розвиток лідерських навичок, адаптація нових співробітників, підтримка безперервного навчання.

Переваги: Індивідуальні програми навчання, знижки на великі корпоративні пакети, аналітика та звітність про успішність навчання, підтримка та консультації.

Приклади курсів: Лідерство і управління, управління проектами, комунікативні навички, технічні курси (ІТ, інженерія), тренінги з безпеки.

2.3.1 Конкурентні переваги

Конкурентні переваги - це те, що робить ваш онлайн-сервіс з продажу курсів кращим за інших. Щоб виділитися, важливо розробити стратегію, що дозволить привабити користувачів і завоювати їх довіру.

Унікальний контент:

- Ексклюзивні курси: Надавайте курси, розроблені експертами, які недоступні на інших платформах. Також розробіть спеціалізовані курси, що відповідають сучасним тенденціям і вимогам ринку.

- Авторські методики: Використовуйте інноваційні методики навчання, розроблені провідними фахівцями. Включайте інтерактивні завдання, кейси і практичні проекти, щоб навчання було цікавим і ефективним.

Висока якість навчання:

- Інтерактивні методики: Використовуйте відео-лекції, тести, практичні завдання, симуляції, щоб студенти краще засвоювали матеріал. Надайте систему зворотного зв'язку, де студенти отримують оцінки і рекомендації від викладачів.

- Актуальність контенту: Регулярно оновлюйте навчальні матеріали відповідно до нових тенденцій і стандартів галузі. Застосовуйте реальні кейси і приклади з практики, щоб навчання було релевантним.

Доступні ціни:

- Гнучка цінова політика: Надайте різні варіанти цін, щоб задовольнити потреби різних категорій користувачів. Пропонуйте пакети курсів за вигідною ціною, що включають знижки на комплексні програми навчання.

– Знижки і акції: Проводьте регулярні акції і знижки для нових і постійних клієнтів. Розробіть програми лояльності, що надають бонуси за тривале використання платформи.

Гнучкість навчання:

– Навчання у зручний час: Дозвольте студентам навчатися в будь-який час, що особливо актуально для зайнятих професіоналів і студентів. Надайте доступ до матеріалів курсу 24/7.

– Мобільний доступ: Забезпечте адаптивний дизайн, що дозволяє користуватися платформою на різних пристроях (смартфонах, планшетах, ноутбуках). Розробіть спеціальні мобільні додатки для зручного доступу до курсів.

Застосовуючи ці принципи, ви зможете створити конкурентноздатну платформу онлайн-навчання, яка привертатиме увагу студентів і забезпечить успіх вашого бізнесу.

2.3.2 Фінансові показники

Фінансові показники - це компас, який допомагає оцінити успіх і ефективність бізнесу, зокрема платформи онлайн-навчання. Для вебсайту з продажу онлайн курсів ключовими показниками є:

Дохід: Це гроші, які платформа отримує від своїх послуг. Дохід може походити з різних джерел:

Продаж курсів: Це основний джерело доходу для платформи онлайн-навчання.

Підписки: Це регулярні платежі за доступ до певних курсів або функцій платформи.

Корпоративні пакети: Це угоди з компаніями, які придбають доступ до курсів для своїх співробітників.

Партнерські програми: Це співпраця з іншими компаніями або особами, які розповсюджують курси платформи і отримують відсоток від продажу.

Витрати: Це гроші, які платформа витрачає на свою діяльність:

Розробка і підтримка платформи: Це витрати на розробку та обслуговування технічної частини платформи.

Маркетинг і реклама: Це витрати на просування платформи і приваблення нових користувачів.

Оплата праці викладачів і співробітників: Це зарплата викладачів, які створюють і проводять курси, та інших співробітників платформи.

Технічна підтримка і обслуговування: Це витрати на забезпечення безперебійної роботи платформи і надання технічної допомоги користувачам.

Рентабельність: Це показник ефективності платформи, який відображає співвідношення доходу і витрат. Ключовими показниками рентабельності є:

Рентабельність продажів: Відображає прибуток від продажу курсів у відсотках від доходу від продажу.

Рентабельність інвестицій (ROI): Відображає ефективність інвестицій у платформу і показує, скільки грошей приносить кожна інвестована гривня.

Операційна рентабельність: Відображає прибуток платформи від її операційної діяльності.

Коефіцієнти прибутковості: Це показники, які відображають прибутковість платформи і її здатність генерувати прибуток:

Маржа валового прибутку: Відображає різницю між доходом від продажу курсів і собівартістю продажу курсів.

Чистий прибуток: Відображає прибуток платформи після вирахування всіх витрат.

Показники росту: Ці показники допомагають оцінити темпи розвитку платформи і її здатність збільшувати свій бізнес:

Зростання доходів: Відображає збільшення доходу платформи у порівнянні з попередніми періодами.

Збільшення кількості користувачів: Відображає збільшення кількості користувачів, які користуються платформою.

Збільшення кількості підписок і корпоративних клієнтів: Відображає збільшення кількості користувачів, які підписалися на платформу або придбали корпоративні пакети.

Інвестиції: Цей розділ дозволяє оцінити фінансові ресурси, які використовуються для розвитку платформи

Розмір залучених інвестицій: Відображає суму грошей, які були інвестовані в платформу

Витрати на розвиток нових функцій і розширення платформи: Відображає витрати на впровадження нових функцій і збільшення функціональності платформи.

Аналізуючи ці фінансові показники, власники платформи онлайн-навчання можуть оцінити ефективність бізнесу, визначити сильні і слабкі сторони та розробити стратегію для збільшення прибутковості і розвитку платформи.

2.3.3 Доступ до куплених курсів та додаткових матеріалів

Забезпечення зручного доступу до куплених курсів та додаткових матеріалів є критично важливим аспектом функціональності платформи онлайн-навчання, оскільки це безпосередньо впливає на досвід навчання користувачів. Ефективна система доступу до контенту сприяє задоволенню студентів і збільшує імовірність успішного завершення курсів.

Кожен користувач повинен мати особистий кабінет, що слугує централізованим пунктом доступу до всіх їхніх курсів. Кабінет повинен наочно відображати список куплених курсів, рівень прогресу в навчанні, а також доступні матеріали для кожного курсу. Важливо забезпечити можливість відновлення доступу до курсів, які були придбані раніше. Кабінет повинен мати систему для

організації матеріалів, наприклад, можливість створювати теки або списки "ознайомитись пізніше", що дозволить студентам легко орієнтуватися в навчальних матеріалах.

Платформа повинна мати інтегрований відеоплеєр з зручним інтерфейсом і функціями керування відтворенням (пауза, перемотування тощо), що зробить перегляд відеоуроків більш комфортним. Платформа повинна надавати можливість завантаження додаткових матеріалів, таких як PDF-файли, презентації і завдання, що дозволить студентам ознайомитися з ними в офлайн-режимі. Можливість робити відмітки і нотатки під час перегляду відеоуроків сприяє кращому засвоєнню інформації і збільшує ефективність навчання.

Доступ до додаткових ресурсів, таких як статті, книги, посилання на корисні вебсайти, розширює знання і допомагає студентам глибше ознайомитися з темою курсу. Вебінари та онлайн-семінари додають до курсів інтерактивність і надають можливість взаємодіяти з викладачем і іншими студентами в реальному часі. Форум для обговорень дозволяє студентам спілкуватися між собою і з викладачем, обмінюватися досвідом, ставити питання і отримувати відповіді.

Система оповіщень інформує студентів про новий контент і оновлення курсу, що дозволяє їм бути в курсі всіх змін і не пропустити важливу інформацію. Нагадування допомагають студентам завжди пам'ятати про завдання і тести і вчасно їх виконувати. Система оповіщень інформує студентів про вебінари і онлайн-семінари, що дозволяє їм взяти участь в цих заходах.

Доступ до курсів з мобільних пристроїв збільшує гнучкість навчання і дозволяє студентам вчитися в будь-якому місці і в будь-який час. Спеціальні мобільні додатки з оптимізованим інтерфейсом і функціями, призначеними для мобільних пристроїв, зроблять навчання на смартфонах і планшетах більш комфортним і ефективним.

Система має забезпечувати захист контенту від несанкціонованого доступу і копіювання. Платформа повинна збирати дані про активність користувачів, що дозволить визначити ефективність курсів і зробити необхідні коригування. Студенти повинні мати можливість звернутися за технічною підтримкою в разі

виникнення проблем з доступом до курсів або матеріалів наведено приклад на таб. 2.1 .

Таблиця 2.1 – Конкурентні переваги

| Конкурентні переваги | Опис |
|------------------------|--|
| Унікальний контент | Ексклюзивні курси, авторські методики |
| Висока якість навчання | Інтерактивні методики, актуальність контенту |
| Доступні ціни | Гнучка цінова політика, знижки та акції |
| Гнучкість навчання | Навчання у зручний час, мобільний доступ |
| Підтримка користувачів | Технічна підтримка, консультації |
| Сертифікація | Офіційні сертифікати, співпраця з професійними організаціями |

Джерело: розроблено автором на основі [10]

Ретельна розробка системи доступу до куплених курсів і додаткових матеріалів є ключовим фактором успіху платформи онлайн-навчання. Зручна та ефективна система доступу до контенту сприяє задоволенню студентів, збільшує ефективність навчання і підвищує імовірність успішного завершення курсів.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ СИСТЕМИ

3.1 Використання UML-діаграм

Unified Modeling Language (UML) — це стандартна мова для візуального моделювання систем, що широко використовується в програмній інженерії. UML-діаграми дозволяють описати різні аспекти системи, такі як її структуру, поведінку та взаємодію між компонентами.

1. Основні типи UML-діаграм:

Діаграми випадків використання (Use Case Diagrams):

Відображають взаємодію користувачів із системою через різні сценарії використання (випадки використання).

Показують, які функції системи доступні користувачам.

Діаграми класів (Class Diagrams):

Відображають структуру системи у вигляді класів та їх взаємозв'язків.

Описують атрибути та методи класів, а також зв'язки між ними (асоціації, наслідування, агрегації).

– Діаграми послідовності (Sequence Diagrams):

Показують порядок взаємодії між об'єктами у часі.

Відображають обмін повідомленнями між об'єктами для виконання конкретного сценарію.

– Діаграми станів (State Diagrams):

Описують можливі стани об'єкта та переходи між ними залежно від подій.

Використовуються для моделювання поведінки об'єктів у відповідь на зовнішні та внутрішні події.

– Діаграми активностей (Activity Diagrams):

Показують послідовність виконання дій та процесів у системі.

Відображають потоки даних та логіку бізнес-процесів.

1. Діаграма випадків використання (Use Case Diagram)

Мета: Визначити, які функціональні можливості має система і як користувачі взаємодіють із системою.

Приклад:

Актори: Користувач, Інструктор, Адміністратор

Випадки використання: Реєстрація, Авторизація, Перегляд курсу, Купівля курсу, Додавання курсу, Редагування курсу, Видалення курсу.

Інструкція для створення:

Вставте прямокутник для системи і назвіть його "Система продажу онлайн курсів".

Додайте акторів зліва від прямокутника (наприклад, "Користувач" та "Адміністратор").

Додайте овали для кожного випадку використання всередині прямокутника і з'єднайте їх з відповідними акторами.

3.2 Діаграма класів

Діаграма класів - це графічний інструмент, який використовується в об'єктно-орієнтованому програмуванні для візуалізації структури та взаємозв'язків між класами в системі. Вона є важливим елементом системного аналізу і проектування, оскільки допомагає розробити структуру системи, визначити її компоненти та їхні взаємодії.

В об'єктно-орієнтованому програмуванні клас - це шаблон, що визначає властивості і поведінку об'єктів. Він визначає дані (атрибути) і функції (методи), які будуть доступні для кожного об'єкта, створеного на основі цього класу. Наприклад, клас "Автомобіль" може мати атрибути, такі як "Марка", "Модель", "Колір", і методи, такі як "Їхати", "Гальмувати", "Змінити швидкість".

Діаграма класів відображає класи системи і їхні взаємозв'язки. Кожен клас зображається у вигляді прямокутника, всередині якого розміщуються назви атрибутів і методів. Взаємозв'язки між класами зображаються у вигляді ліній, які з'єднують прямокутники.

Найбільш загальний тип взаємозв'язку - асоціація, що вказує на те, що класи пов'язані між собою. Узагальнення вказує на ієрархічне співвідношення між класами, коли один клас є узагальненням інших класів (наприклад, клас "Транспортний засіб" є узагальненням класів "Автомобіль", "Літак" і "Потяг"). Агрегація вказує на те, що один клас складається з інших класів (наприклад, клас "Комп'ютер" складається з класів "Процесор", "Оперативна пам'ять", "Жорсткий диск"). Композиція - більш міцний тип агрегації, що вказує на те, що один клас повністю контролює життя інших класів (наприклад, клас "Кімната" складається з класів "Стіни", "Підлога", "Стеля", і без "Кімнати" ці класи не існують).

Діаграми класів допомагають зрозуміти структуру системи і її компоненти, візуалізувати взаємозв'язки між класами і зрозуміти, як вони взаємодіють між собою, створити документацію системи і зберегти її структуру, та допомагають розробникам спілкуватися між собою і зрозуміти структуру системи на рис. 3.1 наведено приклад.

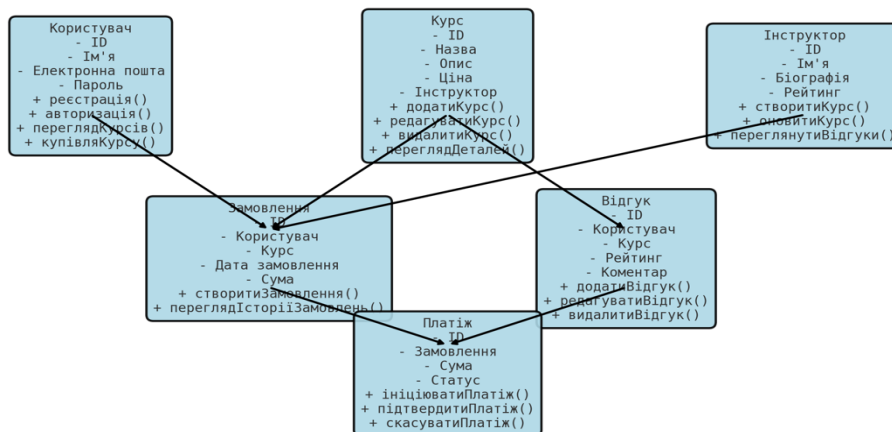


Рисунок 3.1 – Діаграма класів

Джерело: розроблено автором на основі [6]

- User. Ініціатор процесу, який хоче вивести токени.
- Token. Представляє конкретну криптовалюту, яку користувач вибирає для виведення.
 - WithdrawalTokens. Це початкова точка сценарію взаємодії, яка може бути активована користувачем для початка процесу виведення.
 - ChooseToken. Етап, на якому користувач вибирає специфічний токен для виведення.
 - GetBalance. Компонент системи, який відповідає за отримання інформації про баланс токенів на рахунку користувача.
 - Withdrawal. Компонент, який обробляє сам процес виведення, включаючи перевірку можливості виведення та виконання транзакції.
 - PostReq. Обробляє запити від користувача на виведення коштів та комунікацію з внутрішніми або зовнішніми службами для реалізації цього запиту.
 - SaveRes. Компонент, призначений для збереження результатів операції у базі даних системи.
 - Amount. Величина, яку користувач хоче вивести, відіграє ключову роль у процесі виведення.
 - SendTokens. Кінцева дія у сценарії, де токени відправляються користувачу або на зовнішній гаманець.

РОЗДІЛ 4

ПРОЕКТУВАННЯ СИСТЕМИ

4.1 Архітектура системи

Архітектура системи визначає основні компоненти системи, їх взаємозв'язки та способи взаємодії. Вона включає серверну частину, клієнтську частину, базу даних та мережеву інфраструктуру.

Основні компоненти архітектури:

Клієнтська частина (Frontend):

Відповідає за взаємодію з користувачем.

Використовує HTML, CSS, JavaScript та фреймворки (наприклад, React, Angular, Vue.js).

Забезпечує рендеринг сторінок, обробку подій користувача та взаємодію з сервером через API.

Серверна частина (Backend):

Відповідає за логіку роботи додатку та обробку запитів від клієнтської частини.

Використовує серверні мови програмування (наприклад, Node.js, Python, Java, PHP) та фреймворки (наприклад, Express, Django, Spring).

Виконує обробку даних, аутентифікацію, авторизацію, обробку платежів та управління користувачами.

База даних (Database):

Зберігає дані додатку, такі як інформація про користувачів, курси, замовлення та платежі.

Використовує СУБД (наприклад, MySQL, PostgreSQL, MongoDB).

Забезпечує надійне зберігання та швидкий доступ до даних.

Мережева інфраструктура:

Забезпечує зв'язок між клієнтською та серверною частинами.

Використовує HTTP/HTTPS протоколи для передачі даних.

Включає балансувальники навантаження, проксі-сервери та мережеві брандмауери.

4.2 Дизайн інтерфейсі викладача

Успішний UI-дизайн платформи онлайн-навчання - це мистецтво створення інтерфейсу, який не тільки візуально привабливий, але й зручний, доступний та ефективний для користувачів.

Простота та зручність: Ключовими принципами є простота та інтуїтивність. Інтерфейс повинен бути зрозумілим і легким у використанні, без надмірного ускладнення та зайвих елементів. Користувачі повинні швидко знаходити необхідну інформацію та виконувати потрібні дії без зусиль.

Послідовність: Важливою умовою є послідовність у використанні елементів дизайну та стилю на всіх сторінках платформи. Елементи інтерфейсу (кнопки, меню, шрифти тощо) повинні мати однакове візуальне оформлення та розташовуватися на сторінках в одному і тому ж місці. Це дозволяє користувачам швидко звикати до платформи і легко навігувати між сторінками.

Доступність: Інтерфейс має бути доступним для всіх користувачів, включаючи людей з обмеженими можливостями. Це означає використання відповідних кольорів, шрифтів та розмірів тексту, а також застосування спеціальних функцій та інструментів, які допоможуть людям з особливими потребами користуватися платформою.

Реагуючий дизайн (Responsive Design): У світі, де користувачі доступють до інтернету з різних пристроїв, важливо забезпечити зручність використання платформи на комп'ютерах, планшетах і смартфонах. Це досягається за допомогою адаптивних макетів, які змінюються відповідно до розміру екрану пристрою.

Зворотний зв'язок: Важливо надавати користувачам інформацію про результат їхніх дій. Це може бути у вигляді повідомлень, сповіщень та анімацій, що дозволяють користувачам зрозуміти, що їхні дії були успішними.

Основні компоненти UI:

Головна сторінка: Це перше, що бачать користувачі, тому важливо створити приємне і інформативне враження. На головній сторінці має бути вітання, навігаційне меню, список доступних курсів та рекомендації.

Сторінка курсу: На цій сторінці надається детальна інформація про курс (назва, опис, інструктор, ціна), відгуки користувачів та кнопка для реєстрації на курс.

Особистий кабінет: Тут користувачі можуть переглянути інформацію про свій профіль, список куплених курсів та здійснити редагування налаштувань.

Сторінка кошика: Тут користувачі можуть переглянути список вибраних курсів для покупки, підсумкову суму і перейти до оформлення замовлення.

Сторінка оплати: На цій сторінці користувачі вводять платіжну інформацію для оплати курсів.

Дотримання цих принципів і уважне продумування основних компонентів UI дозволить створити платформу онлайн-навчання, яка буде зручною і ефективною для користувачів, що сприятиме їхньому успішному навчанню.

Сторінки прототипу системи

Розглянемо прототипи сторінок інформаційної системи «Криптоінвестиції» що представлені на рис. 4.1-4.12.

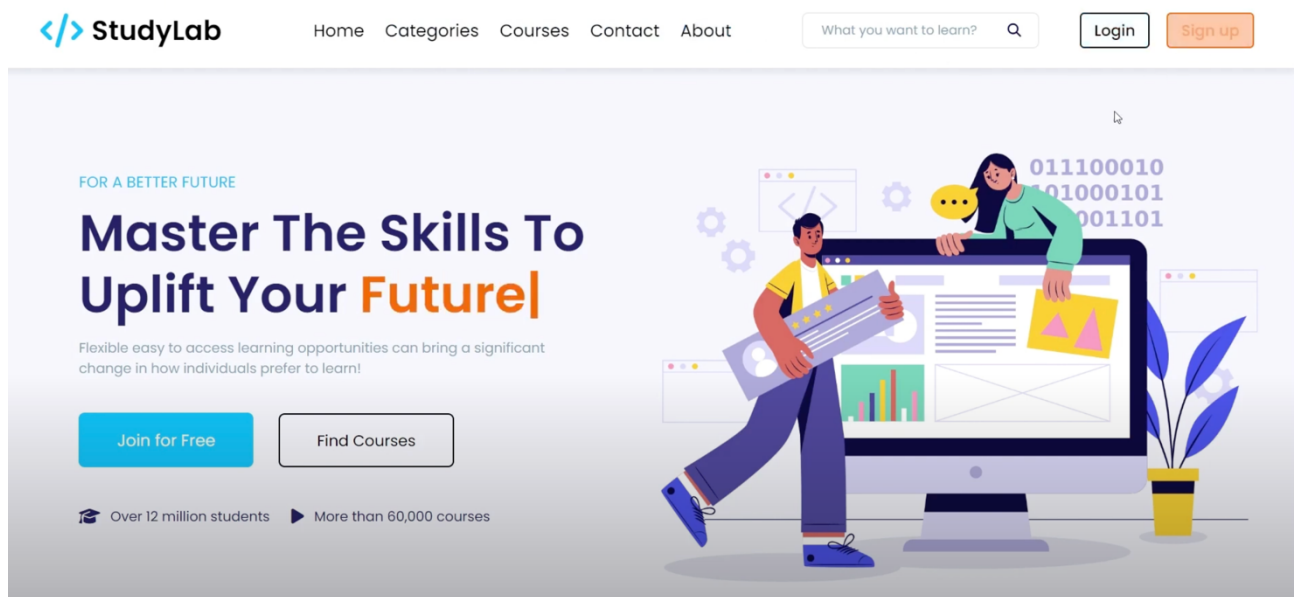


Рисунок 4.1 – Головна сторінка

Джерело: розроблено автором на основі [10]

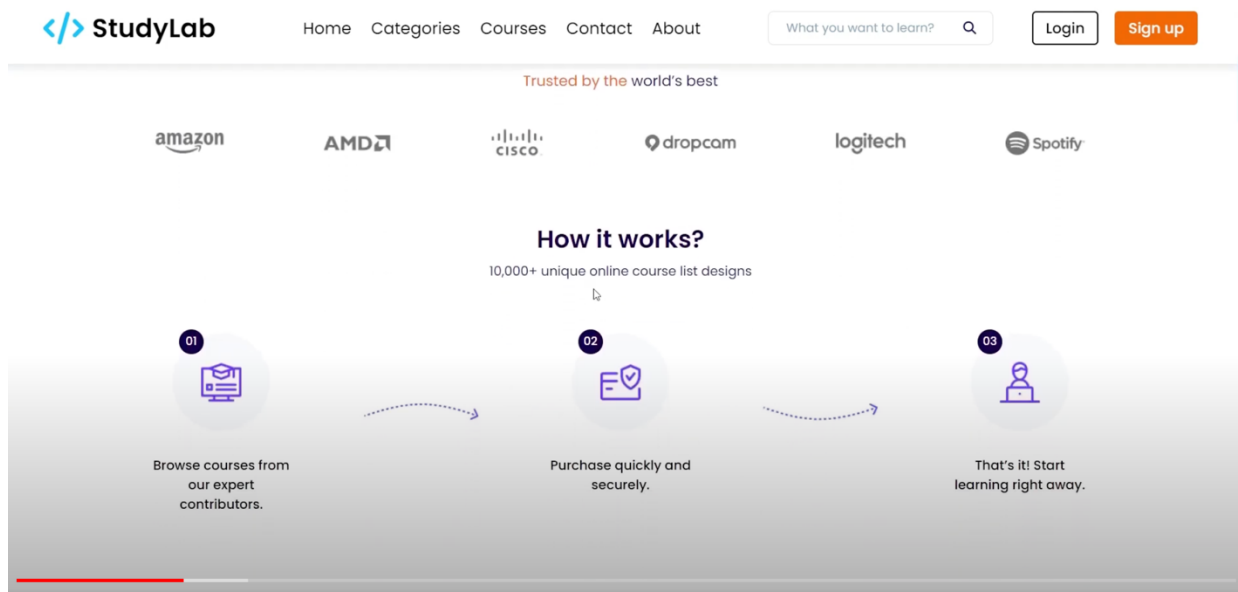


Рисунок 4.2 – Сторінка про нас

Джерело: розроблено автором на основі [10]

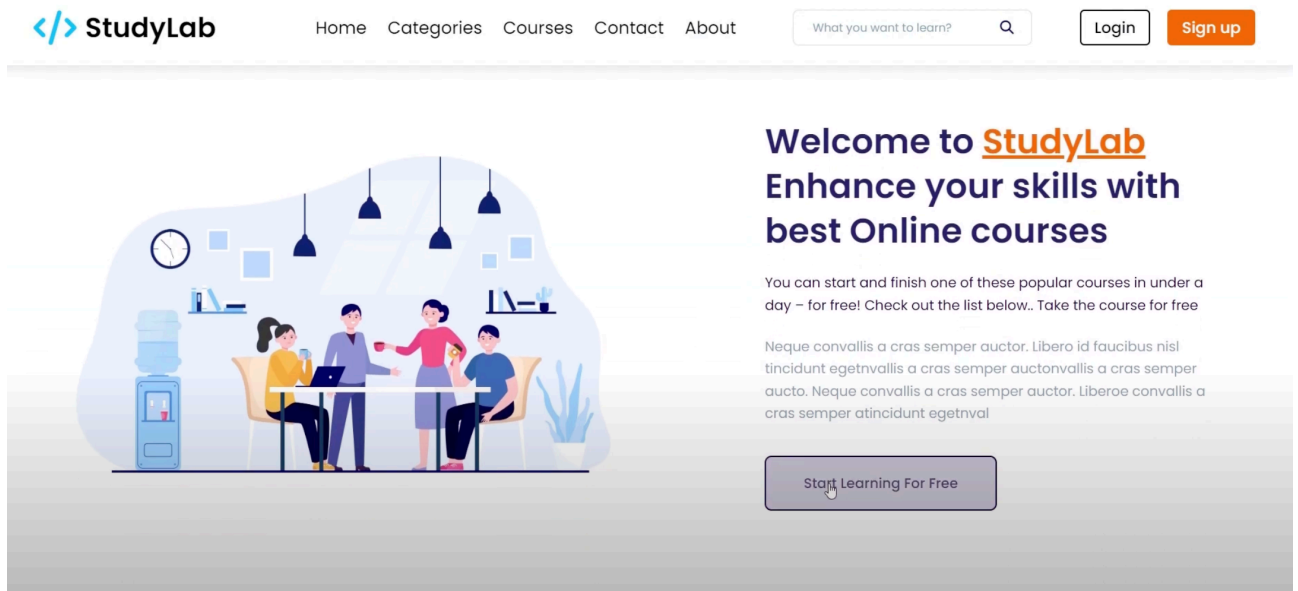


Рисунок 4.3 – Сторінка «About Us»

Джерело: розроблено автором на основі [10]

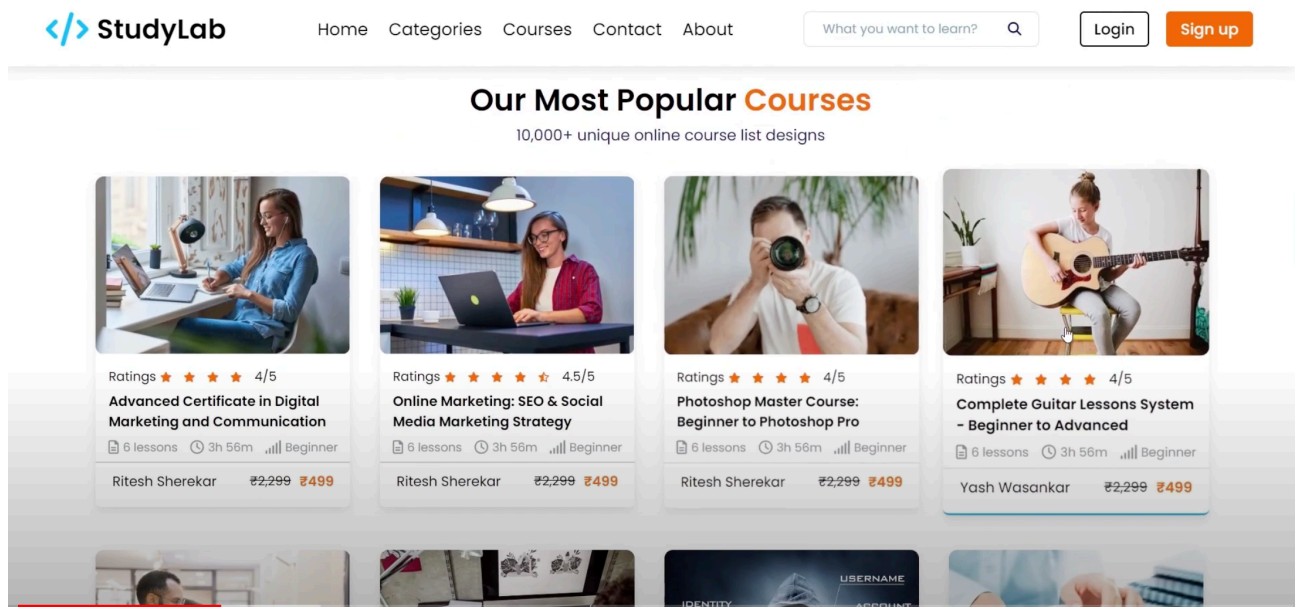


Рисунок 4.4 – Випадаючий список «Courses»

Джерело: розроблено автором на основі [10]

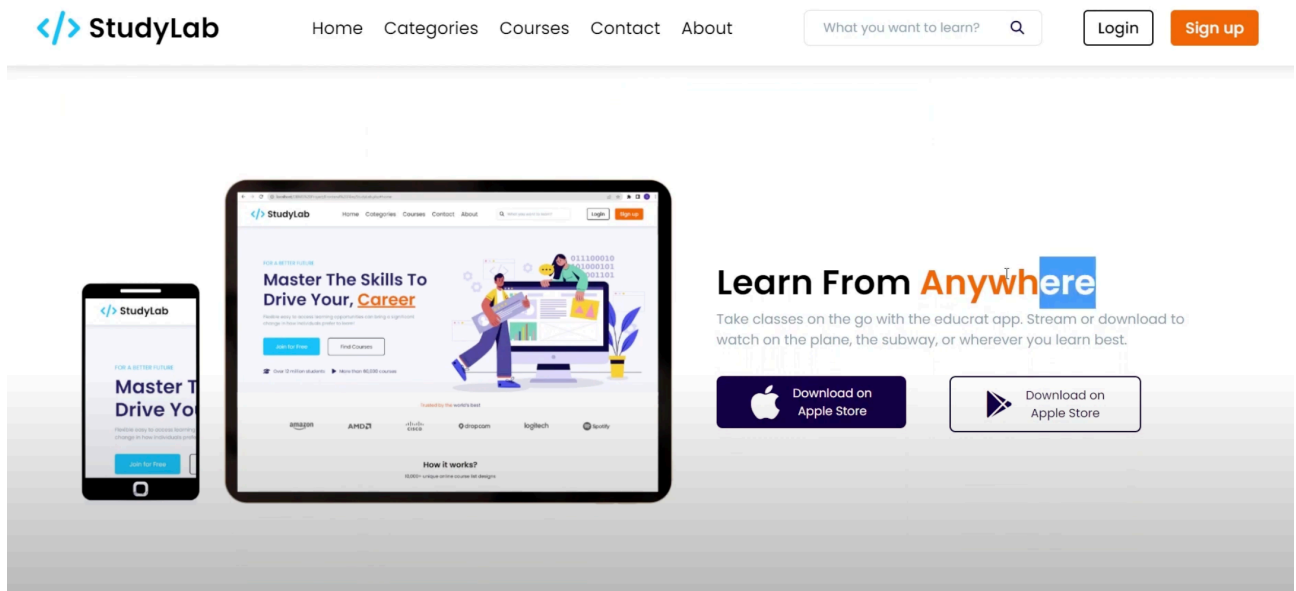


Рисунок 4.5 – Завантажити додаток на телефон

Джерело: розроблено автором на основі [10]

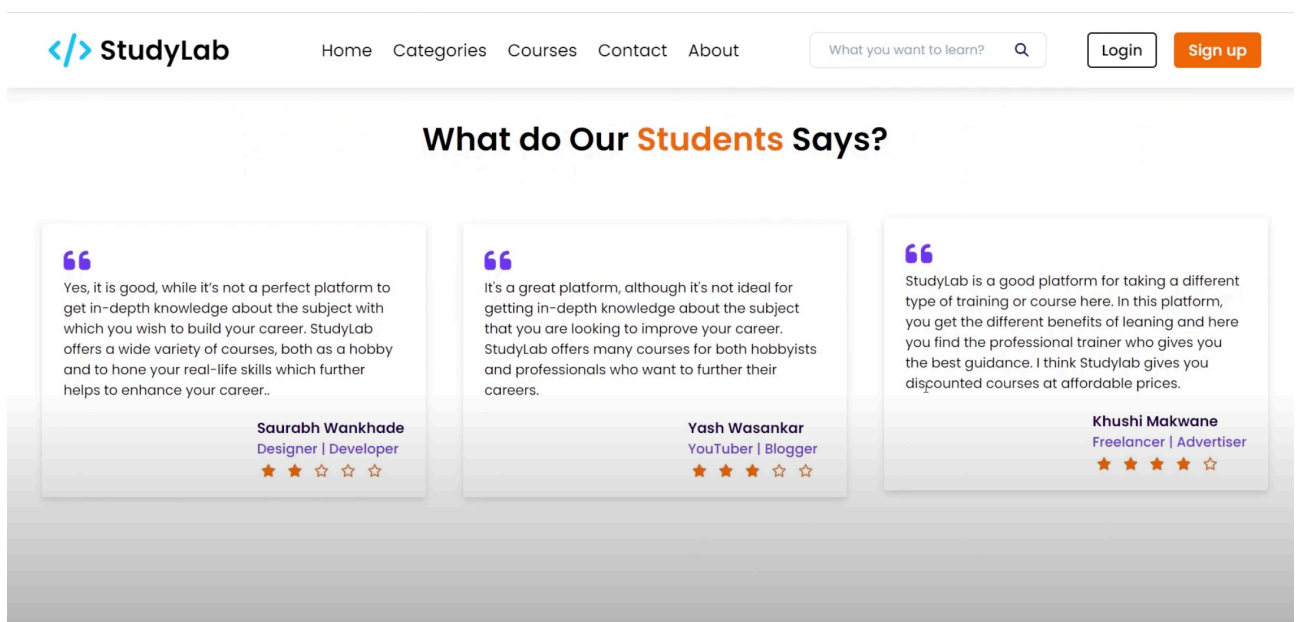


Рисунок 4.6 – Сервіс «Відгуки»

Джерело: розроблено автором на основі [10]

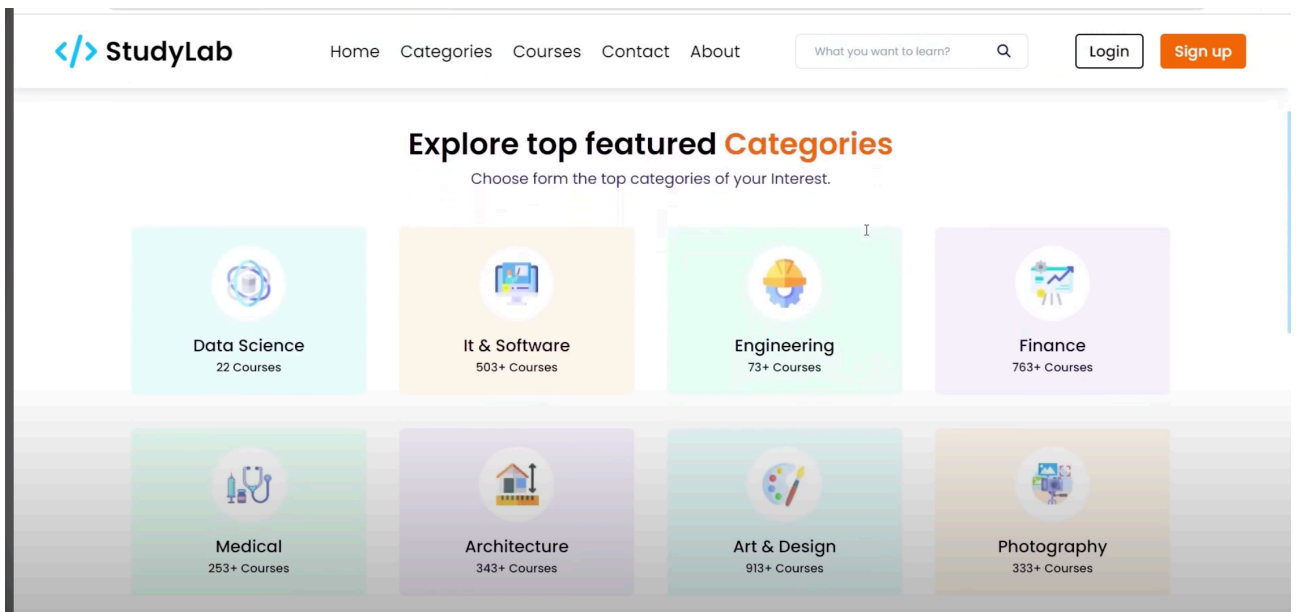


Рисунок 4.7 – Сервіс «Categories»

Джерело: розроблено автором на основі [10]

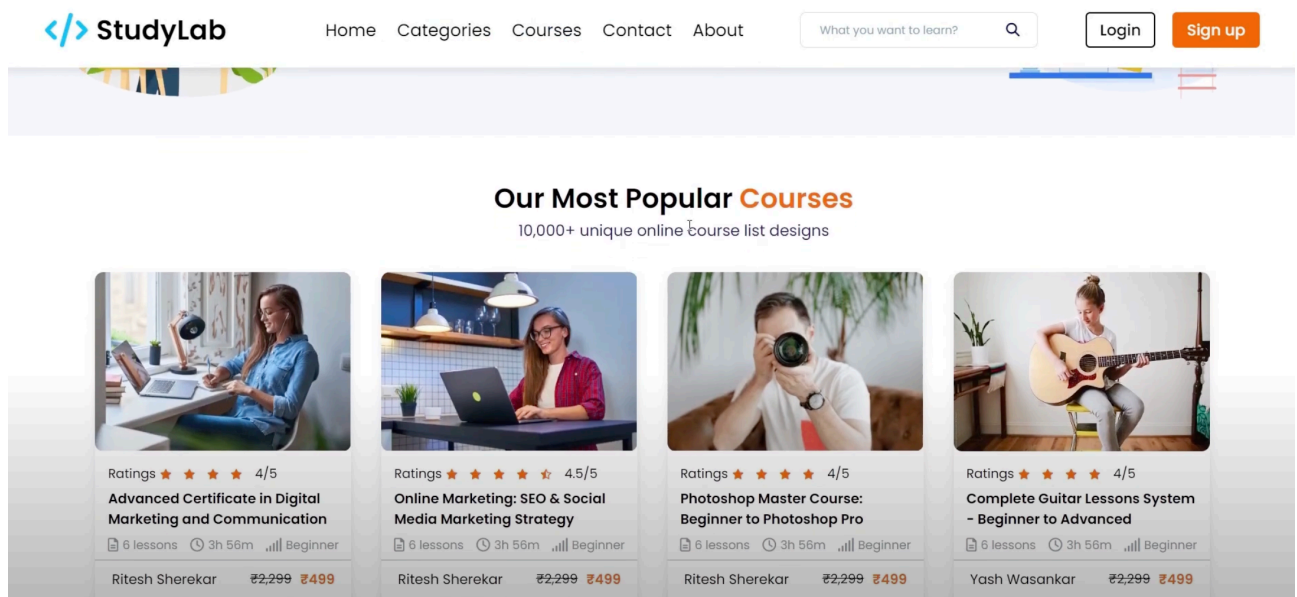


Рисунок 4.8 – Сервіс «Courses»

Джерело: розроблено автором на основі [10]

The screenshot shows the StudyLab website interface. At the top, there is a navigation bar with the StudyLab logo, links for Home, Categories, Courses, Contact, and About, a search bar, and buttons for Login and Sign up. The main content area features a course card for 'Advanced Certificate in Digital Marketing and Communication'. The card includes a 'BEST SELLER' badge, a rating of 4.5 stars, and a price of ₹499 (78% off from ₹2,299). It lists 30 modules, a beginner skill level, a certificate, and full lifetime access. The course is taught by Rishikesh Shah and has 853 enrolled students. A 'Buy Now' button and an 'Add to cart' button are visible at the bottom of the card.

Рисунок 4.9 – Сервіс «Courses»

Джерело: розроблено автором на основі [10]

The screenshot shows the StudyLab website interface for the login page. At the top, there is a navigation bar with the StudyLab logo, links for Home, Categories, Courses, Contact, and About, a search bar, and buttons for Login and Sign up. The main content area features a large illustration of a woman standing next to a large smartphone displaying a user profile, and a man sitting on the floor using a laptop. To the right of the illustration is a login form with the heading 'Welcome'. The form includes input fields for 'Username Or Email' and 'Password', a 'Login' button, and a link for 'Create Account'.

Рисунок 4.10 – Сервіс «Login»

Джерело: розроблено автором на основі [10]

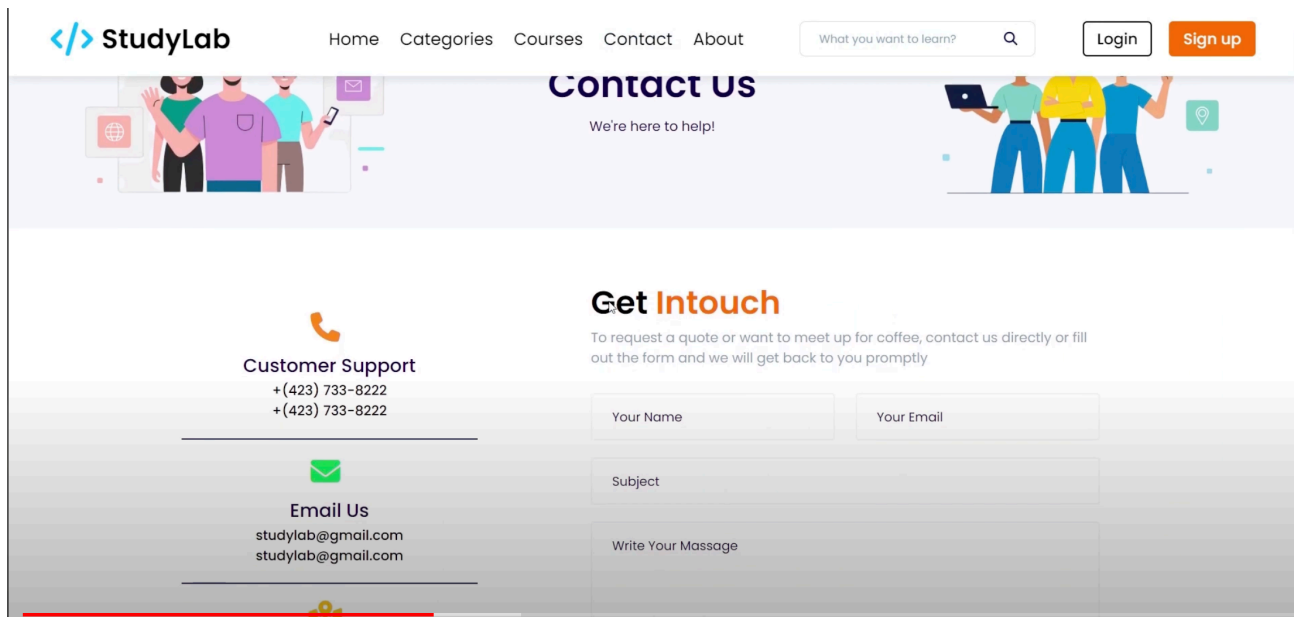


Рисунок 4.11 – Сервіс «Contact Us»

Джерело: розроблено автором на основі [10]

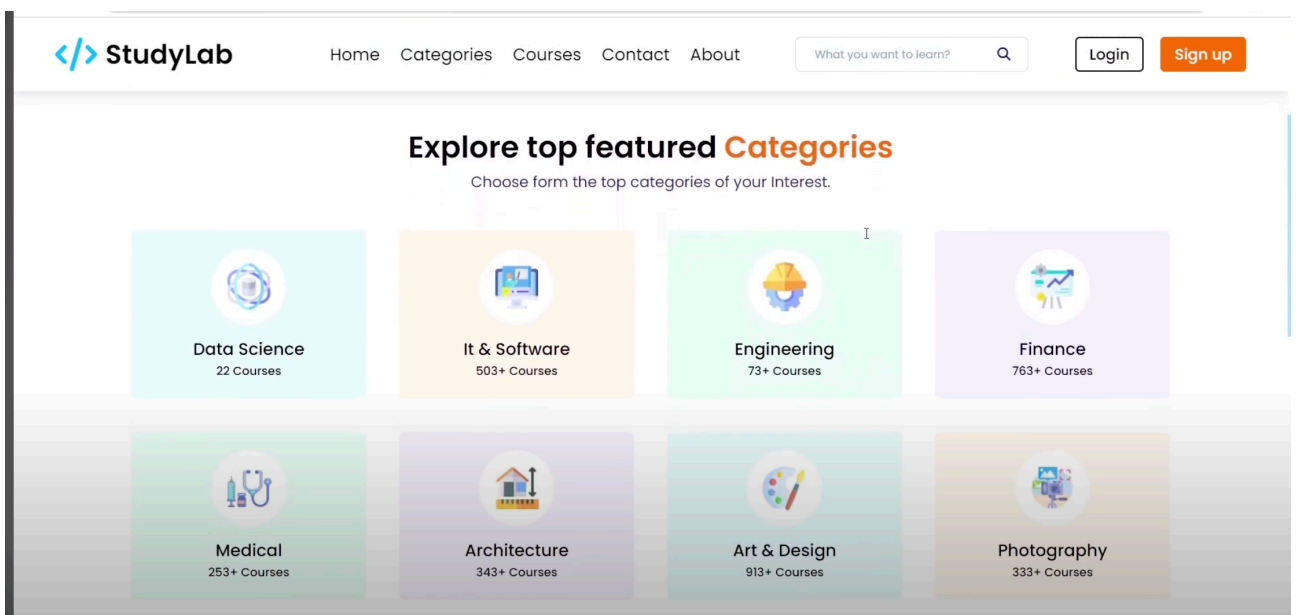


Рисунок 4.12 – Сервіс «Contact Us»

Джерело: розроблено автором на основі [10]

Фрагмент коду:**HTML код сторінки особистого кабінету:**

```

<nav class="navbar navbar-expand-md navbar-light bg-light">
  <div class="container">
    <div class="logo">
      <a href="../index.html">
        
      </a>
    </div>
    <div class="site-menu">
      <div class="menu">
        <div class="collapse navbar-collapse">
          <ul class="nav navbar-nav" itemscope
itemtype="http://www.schema.org/SiteNavigationElement">
            <li class="nav-item">
              <a itemprop="url" href="cabinet-user.html" class="nav-link">
                <span itemprop="name">Cabinet <span class="border border-top"></span>
                <span class="border border-right"></span>
                <span class="border border-bottom"></span>
                <span class="border border-left"></span>
              </span>
            </a>
          </li>
          <li class="nav-item">
            <a itemprop="url" href="deposit.html" class="nav-link" aria-current="page">
              <span itemprop="name">Deposit <span class="border border-top"></span>
              <span class="border border-right"></span>
              <span class="border border-bottom"></span>
              <span class="border border-left"></span>
            </span>
          </a>
        </li>
        <li class="nav-item">

```

```

CREATE TABLE Token (
breakpoints: {
  640: {
    slidesPerView: 1,
    spaceBetween: 15,
  },
  768: {
    slidesPerView: 2,
    spaceBetween: 30,
  },
  1024: {
    slidesPerView: 3,
    spaceBetween: 30,
  },
}
});

```

```
// MAIN SLIDER
```

```
var swiper = new Swiper('.main-slider',
```

Фрагмент JS коду, що відповідає за анімації переходу, задній фон, слайдери та кнопку меню у мобільній версії:

```
// HAMBURGER MENU
```

```
var main = function () {
```

```
  $('hamburger-menu').click(function () {
```

```
    $('.side-widget').animate({
```

```
      left: "0px"
```

```
    }, 200, function () {
```

```
      $(document).on("click.menu", function (event) {
```

```
        var target = $(event.target);
```

```
        console.log(target);
```

```
        if (!target.closest(".side-widget").length || target.closest(".closed").length) {
```

```
          closeMenu(function () {
```

```
    $(document).off("click.menu");  
  });  
}  
})  
});  
});
```

ВИСНОВКИ

Розробка вебсайту для продажу онлайн курсів є складним і багатограним завданням, що вимагає всебічного підходу та детального планування на всіх етапах проекту. У цьому дипломному проекті було розглянуто всі основні аспекти розробки системи, починаючи від аналізу вимог і закінчуючи архітектурою системи та дизайном інтерфейсу користувача.

Аналіз вимог

На початковому етапі проекту був проведений системний аналіз для визначення функціональних і нефункціональних вимог до системи. Було ідентифіковано основні цільові аудиторії, такі як студенти, професіонали, освітні установи та компанії, що шукають можливості для підвищення кваліфікації своїх співробітників. Цей аналіз допоміг зрозуміти потреби користувачів та забезпечити їх задоволення через відповідні функціональні можливості вебсайту.

Системний аналіз

Системний аналіз включав використання UML діаграм для візуалізації та моделювання різних аспектів системи. Було створено діаграми випадків використання, діаграми класів та діаграми послідовності, що дозволило детально описати взаємодію між компонентами системи та користувачами. Це забезпечило чітке розуміння структури та поведінки системи на всіх рівнях.

Архітектура системи

Архітектура системи була розроблена з урахуванням сучасних принципів проектування, таких як розподілена архітектура, масштабованість та безпека. Було виділено основні компоненти архітектури, включаючи клієнтську частину, серверну частину, базу даних та мережеву інфраструктуру. Використання балансувальників навантаження та проксі-серверів забезпечило ефективний розподіл ресурсів та надійність системи.

Дизайн інтерфейсу користувача

Особлива увага була приділена дизайну інтерфейсу користувача, який має бути інтуїтивно зрозумілим, зручним та доступним для всіх категорій користувачів. Принципи простоти, послідовності, доступності та реагуючого дизайну були

застосовані для створення привабливого та функціонального інтерфейсу, що сприяє задоволенню користувачів та підвищенню їх лояльності.

Технічна реалізація

Для технічної реалізації проекту було обрано сучасні технології та фреймворки, що забезпечують високу продуктивність, безпеку та гнучкість системи. Використання фронтенд та бекенд фреймворків, таких як React, Node.js та Django, дозволило створити модульну та легко підтримувану систему.

Цей проект надає всебічне розуміння процесу розробки вебсайту для продажу онлайн курсів, забезпечуючи теоретичні та практичні знання, необхідні для успішної реалізації подібних систем. Результати проекту можуть бути використані як основа для подальшого розвитку та вдосконалення системи, а також як приклад для інших проектів у цій галузі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Векторний онлайн-сервіс розробки інтерфейсів [Електронний ресурс] // Figma – Режим доступу до ресурсу: <https://www.figma.com/>. (дата звернення 02.03.2024)
- 2) Створення та редагування діаграм [Електронний ресурс] // LucidChart – Режим доступу до ресурсу: <https://www.lucidchart.com/pages>. (дата звернення 02.03.2024)
- 3) Framework spring boot [Електронний ресурс] // Spring Boot – Режим доступу до ресурсу: <https://spring.io/projects/spring-boot>. (дата звернення 02.03.2024)
- 4) Онлайн-програмне забезпечення для створення блок-схем [Електронний ресурс] // Draw.IO – Режим доступу до ресурсу: <https://app.diagrams.net/>. (дата звернення 06.03.2024)
- 5) UML modeling tools for Business, Software, Systems [Електронний ресурс] – Режим доступу до ресурсу: <https://sparxsystems.com/>. (дата звернення 02.03.2024)
- 6) IntelliJ IDEA [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/IntelliJ_IDEA. (дата звернення 02.03.2024)
- 7) База даних MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.photo-555.com/2686691-what-is-mysql-database>. (дата звернення 10.03.2024)
- 8) IBM Knowledge Center. What is System Analysis? [Електронний ресурс] / IBM. – URL: <https://www.ibm.com/docs/en/what-is-system-analysis>. (дата звернення 02.03.2024)
- 9) TechTarget. Definition of System Analysis [Електронний ресурс] / TechTarget. – URL: <https://www.techtarget.com/definition/system-analysis>. (дата звернення 02.03.2024)
- 10) Coursera. Online Learning Platforms [Електронний ресурс] / Coursera. – URL: <https://www.coursera.org/about>. (дата звернення 02.03.2024)

- 11) Microsoft Docs. Requirements Gathering [Электронный ресурс] / Microsoft. – URL: <https://docs.microsoft.com/en-us/azure/devops/requirements-gathering>. (дата звернення 02.04.2024)
- 12) Agile Alliance. Functional vs Non-functional Requirements [Электронный ресурс] / Agile Alliance. – URL: <https://www.agilealliance.org/glossary/functional-requirements>. (дата звернення 02.03.2024)
- 13) ISO/IEC 25010:2011. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models [Электронный ресурс] / ISO. – URL: <https://www.iso.org/standard/35733.html>. (дата звернення 02.03.2024)
- 14) Harvard Business Review. Business Requirements [Электронный ресурс] / HBR. – URL: <https://hbr.org/topic/business-requirements>. (дата звернення 12.04.2024)
- 15) W3Schools. UML Class Diagrams [Электронный ресурс] / W3Schools. – URL: https://www.w3schools.com/uml/uml_class.asp. (дата звернення 02.03.2024)
- 16) Lucidchart. Sequence Diagrams [Электронный ресурс] / Lucidchart. – URL: <https://www.lucidchart.com/pages/sequence-diagram>. (дата звернення 02.03.2024)
- 17) Mozilla Developer Network. Web Application Architecture [Электронный ресурс] / MDN. – URL: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Architecture>. (дата звернення 02.03.2024)
- 18) Google Design. Principles of UI Design [Электронный ресурс] / Google. – URL: <https://design.google/library/ui-design-principles>. (дата звернення 02.03.2024)
- 19) AWS. Testing and Deployment Strategies [Электронный ресурс] / AWS. – URL: <https://aws.amazon.com/test-deployment-strategies>. (дата звернення 02.03.2024)
- 20) W3Schools. UML Class Diagrams [Электронный ресурс] / W3Schools. – URL: https://www.w3schools.com/uml/uml_class.asp. (дата звернення 02.05.2024)
- 21) Developer Network. Web Application Architecture [Электронный ресурс] / MDN. – URL: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Architecture>. (дата звернення 02.05.2024)

- 22) WS. Testing and Deployment Strategies [Электронный ресурс] / AWS. – URL: <https://aws.amazon.com/test-deployment-strategies>. (дата звернення 02.05.2024)
- 23) MySQL Documentation. What is MySQL Database? [Электронный ресурс] – URL: <https://uk.photo-555.com/2686691-what-is-mysql-database>. (дата звернення 02.05.2024)
- 24) Coursera. Online Learning Platforms [Электронный ресурс] / Coursera. – URL: <https://www.coursera.org/about>. (дата звернення 02.05.2024)
- 25) Techopedia. What is a Website? [Электронный ресурс] / Techopedia. – URL: <https://www.techopedia.com/definition/5392/website>. (дата звернення 02.05.2024)
- 26) Atlassian. Requirements Documentation [Электронный ресурс] / Atlassian. – URL: <https://www.atlassian.com/software/confluence/requirements-documentation>. (дата звернення 02.05.2024)
- 27) EdTech Magazine. Understanding the Needs of Online Learners [Электронный ресурс] / EdTech Magazine. – URL: <https://edtechmagazine.com/higher/article/2020/03/understanding-needs-online-learners>. (дата звернення 02.05.2024)
- 28) Visual Paradigm. What is UML? [Электронный ресурс] / Visual Paradigm. – URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml>. (дата звернення 02.05.2024)
- 29) GeeksforGeeks. UML Sequence Diagrams [Электронный ресурс] / GeeksforGeeks. – URL: <https://www.geeksforgeeks.org/uml-sequence-diagrams>. (дата звернення 02.05.2024)
- 30) Adobe XD Ideas. Prototyping for Web Design [Электронный ресурс] / Adobe XD Ideas. – URL: <https://xd.adobe.com/ideas/guides/prototyping-web-design>.

ДОДАТКИ

Тестова документація

Сценарій: Купівля курсу

Учасники:

User

System

Course

Order

Payment Gateway

Послідовність дій:

User вибирає курс та додає його до кошика.

System відображає кошик із вибраними курсами.

User переходить до оформлення замовлення.

System створює замовлення та ініціює платіж.

Payment Gateway обробляє платіж та підтверджує його.

System оновлює статус замовлення та надає доступ до купленого курсу.

User отримує підтвердження про успішну покупку та доступ до матеріалів курсу.

Класи:**User**

Атрибути: ID, Name, Email, Password

Методи: register(), login(), viewCourses(), purchaseCourse()

Course

Атрибути: ID, Title, Description, Price, Instructor

Методи: addCourse(), editCourse(), deleteCourse(), viewDetails()

Instructor

Атрибути: ID, Name, Bio, Rating

Методи: createCourse(), updateCourse(), viewReviews()

Order

Атрибути: ID, User, Course, Date, Amount

Методи: createOrder(), viewOrderHistory()

Review

Атрибути: ID, User, Course, Rating, Comment

Методи: addReview(), editReview(), deleteReview()

Payment

Атрибути: ID, Order, Amount, Status

Методи: initiatePayment(), confirmPayment(), cancelPayment()

Зв'язки між класами:

Асоціації - відображають взаємодію між двома класами.

Агрегації - відображають відношення "частина-ціле", де частини можуть існувати незалежно від цілого.

Композиції - відображають відношення "частина-ціле", де частини не можуть існувати без цілого.

Наслідування - відображають ієрархію між класами, де один клас успадковує властивості та методи іншого.

Фрагменти коду (контролери)**InvestmentDeposit:**

```

package reflectiveInv.entity;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import reflectiveInv.repository.InvestmentDepositRepository;
import javax.persistence.*;
import java.util.List;

```

```
@Getter
```

```
@Setter
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
@Entity
```

```
@Table(name = "investmentDeposits")
```

```
public abstract class InvestmentDeposit implements InvestmentDepositRepository {
```

```
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```
    private long id;
```

```
    private String token;
```

```
    private String investmentPlan;
```

```
    private double tokenAmount;
```

```
    @ManyToOne
```

```
    @JoinColumn(name = "user_id")
```

```
    private Users users;
```

```
    @OneToMany(mappedBy = "investmentDeposits")
```

```
    private List<TokenHasInvest> tokenHasInvests;
```

```
}
```

Token:

```

package reflectiveInv.entity;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import javax.persistence.*;
import java.util.List;

```

```

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor

@Entity
@Table(name = "token")
public class Token {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    public void setToken_name(String token_name) {
        this.token_name = token_name;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    @Getter
    private String token_name;
    @Getter
    private double price;

    @OneToMany(mappedBy = "token")
    private List<Withdrawal> withdrawal;

    @OneToMany(mappedBy = "token")
    private List<TokenHasInvest> tokenHasInvestList;
}

```

TokenHasInvest:

```

package reflectiveInv.entity;
import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;
import reflectiveInv.repository.TokenHasInvestRepository;
import javax.persistence.*;

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor

@Entity
@Table(name = "token_has_invest")
public abstract class TokenHasInvest implements TokenHasInvestRepository {

```

```

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private long id;

private int quantity;

@ManyToOne
@JoinColumn(name = "token_id")
private Token token;

@ManyToOne
@JoinColumn(name = "investmentDeposits_id")
private InvestmentDeposit investmentDeposit;

public TokenHasInvest(int quantity, Withdrawal withdrawal, InvestmentDeposit investmentDeposit)
{
    this.quantity = quantity;
    this.token = token;
    this.investmentDeposit = investmentDeposit;
}
}

```

Users:

```

package reflectiveInv.entity;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

import javax.persistence.*;
import javax.validation.constraints.Size;
import java.util.List;

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor

@Entity
@Table(name = "users")
public class Users {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
    private String name;
    private String surname;
    private int phone;
    private int balance;
}

```

```

private boolean wallet;

@Size(min=4, message = "Введіть 4 символа")
private String login;

@Size(min=4, message = "Введіть 4 символа")
private String pass;

@OneToMany(mappedBy = "users")
private List<InvestmentDeposit> listInvestmentDeposits;
}

```

Withdrawal:

```

package reflectiveInv.entity;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

import javax.persistence.*;

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor

@Entity
@Table(name = "withdrawal")
public class Withdrawal {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    public void setWithdrawalAmount(double withdrawalAmount) {
        this.withdrawalAmount = withdrawalAmount;
    }

    @Getter
    private double withdrawalAmount;
    private int userId;
    private double availableBalance;

    @ManyToOne
    @JoinColumn(name = "token_id")
    private Token token;
}

```

HTML код сторінки особистого кабінету:

```
<nav class="navbar navbar-expand-md navbar-light bg-light">
  <div class="container">
    <div class="logo">
      <a href="./index.html">
        
      </a>
    </div>
    <div class="site-menu">
      <div class="menu">
        <div class="collapse navbar-collapse">
          <ul class="nav navbar-nav" itemscope
            itemtype="http://www.schema.org/SiteNavigationElement">
            <li class="nav-item">
              <a itemprop="url" href="cabinet-user.html" class="nav-link">
                <span itemprop="name">Cabinet <span class="border border-top"></span>
                <span class="border border-right"></span>
                <span class="border border-bottom"></span>
                <span class="border border-left"></span>
              </span>
            </a>
            </li>
            <li class="nav-item">
              <a itemprop="url" href="deposit.html" class="nav-link" aria-current="page">
                <span itemprop="name">Deposit <span class="border border-top"></span>
                <span class="border border-right"></span>
                <span class="border border-bottom"></span>
                <span class="border border-left"></span>
              </span>
            </a>
            </li>
            <li class="nav-item">
```

```
<a href="operation-cach.html" data-toggle="dropdown" aria-expanded="false"
    class="dropdown-toggle nav-link">
    <span itemprop="name">Withdrawal<span class="border border-top"></span>
      <span class="border border-right"></span>
      <span class="border border-bottom"></span>
      <span class="border border-left"></span>
    </span>
  </a>
</li>
<li class="nav-item">
  <a itemprop="url" href="operations.html" class="nav-link">
    <span itemprop="name">Operations <span class="border border-top"></span>
      <span class="border border-right"></span>
      <span class="border border-bottom"></span>
      <span class="border border-left"></span>
    </span>
  </a>
</li>
<li class="nav-item">
  <a itemprop="url" href="partners.html" class="nav-link">
    <span itemprop="name">Partners <span class="border border-top"></span>
      <span class="border border-right"></span>
      <span class="border border-bottom"></span>
      <span class="border border-left"></span>
    </span>
  </a>
</li>
<li class="nav-item">
  <a itemprop="url" href="settings.html" class="nav-link">
    <span itemprop="name">Settings <span class="border border-top"></span>
      <span class="border border-right"></span>
```

```
        <span class="border border-bottom"></span>
        <span class="border border-left"></span>
    </span>
</a>
</li>
<li class=" nav-item">
    <a itemprop="url" href="tickets.html" class="nav-link">
        <span itemprop="name">Tickets <span class="border border-top"></span>
        <span class="border border-right"></span>
        <span class="border border-bottom"></span>
        <span class="border border-left"></span>
    </span>
    </a>
</li>
</ul>
</div>
</div>
<div class="hamburger-menu">
    <span></span>
    <span></span>
    <span></span>
</div>
<div class="navbar-button">
    <div class="telh">
        &nbsp;&nbsp;&nbsp;timwer
    </div>
</div>
</div>
</nav>
```

```

<header class="page-header" data-background="">
  <div class="container">
    <h2>PERSONAL STATISTICS
    </h2>
  </div>
</header>
<main>
  <div class="tablo">
    <div class="tablo--1-ve-3 wow fadeInLeft cabinet" data-wow-delay="0.5s"
      style="visibility: visible; animation-delay: 0.5s; animation-name: fadeInLeft;">
      <div class="ozellik-kutu-iletisim" onclick="location.href='#';"
        style="cursor: pointer; will-change: transform; transform: perspective(300px)
rotateX(0deg) rotateY(0deg);"
        data-tilt="">
        
        <h3 class="baslik-4 h-yazi-margin-kucuk-2">0 USDT</h3>
        <a href="#" class="custom-button wow fadeInUp cabinet" data-wow-delay="0.6s"
          style="visibility: visible; animation-delay: 0.6s; animation-name: fadeInUp;">Deposit
        </a>
        <p class="ozellik-kutu-iletisim--yazi">Referral: 0 USDT</p>
        <p class="ozellik-kutu-iletisim--yazi">Withdrawal: 0 USDT</p>
        <p class="ozellik-kutu-iletisim--yazi">Profit: 0 USDT</p>
      </div>
    </div>
    <div class="tablo--1-ve-3 wow fadeInLeft cabinet" data-wow-delay="0.5s"
      style="visibility: visible; animation-delay: 0.5s; animation-name: fadeInLeft;">
      <div class="ozellik-kutu-iletisim" onclick="location.href='#';"
        style="cursor: pointer; will-change: transform; transform: perspective(300px)
rotateX(0deg) rotateY(0deg);"
        data-tilt="">
        
        <h3 class="baslik-4 h-yazi-margin-kucuk-2">0 ETH</h3>
        <a href="#" class="custom-button wow fadeInUp cabinet" data-wow-delay="0.6s"

```

```
→</a>
        style="visibility: visible; animation-delay: 0.6s; animation-name: fadeInUp;">Deposit
</div>
<p class="ozellik-kutu-iletisim--yazi">Referral: 0 ETH</p>
<p class="ozellik-kutu-iletisim--yazi">Withdrawal: 0 ETH</p>
<p class="ozellik-kutu-iletisim--yazi">Profit: 0 ETH</p>
</div>
</div>
<div class="tablo--1-ve-3 wow fadeInLeft cabinet" data-wow-delay="0.5s"
    style="visibility: visible; animation-delay: 0.5s; animation-name: fadeInLeft;">
    <div class="ozellik-kutu-iletisim" onclick="location.href='#';"
        style="cursor: pointer; will-change: transform; transform: perspective(300px)
rotateX(0deg) rotateY(0deg);"
        data-tilt="">
        
        <h3 class="baslik-4 h-yazi-margin-kucuk-2">0 BTC</h3>
        <a href="#" class="custom-button wow fadeInUp cabinet" data-wow-delay="0.6s"
            style="visibility: visible; animation-delay: 0.6s; animation-name: fadeInUp;">Deposit
→</a>
        <p class="ozellik-kutu-iletisim--yazi">Referral: 0 BTC</p>
        <p class="ozellik-kutu-iletisim--yazi">Withdrawal: 0 BTC</p>
        <p class="ozellik-kutu-iletisim--yazi">Profit: 0 BTC</p>
    </div>
</div>
<div class="tablo--1-ve-3 wow fadeInLeft cabinet" data-wow-delay="0.5s"
    style="visibility: visible; animation-delay: 0.5s; animation-name: fadeInLeft;">
    <div class="ozellik-kutu-iletisim" onclick="location.href='#';"
        style="cursor: pointer; will-change: transform; transform: perspective(300px)
rotateX(0deg) rotateY(0deg);"
        data-tilt="">
        
        <h3 class="baslik-4 h-yazi-margin-kucuk-2">0 XRP</h3>
        <a href="#" class="custom-button wow fadeInUp cabinet" data-wow-delay="0.6s"
            style="visibility: visible; animation-delay: 0.6s; animation-name: fadeInUp;">Deposit
→</a>
```

```
<p class="ozellik-kutu-iletisim--yazi">Referral: 0 XRP</p>
  <p class="ozellik-kutu-iletisim--yazi">Withdrawal: 0 XRP</p>
  <p class="ozellik-kutu-iletisim--yazi">Profit: 0 XRP</p>
</div>
</div>
<div class="tablo--1-ve-3 wow fadeInLeft cabinet" data-wow-delay="0.5s"
  style="visibility: visible; animation-delay: 0.5s; animation-name: fadeInLeft;">
  <div class="ozellik-kutu-iletisim" onclick="location.href='#';"
    style="cursor: pointer; will-change: transform; transform: perspective(300px)
rotateX(0deg) rotateY(0deg);"
    data-tilt="">
    
    <h3 class="baslik-4 h-yazi-margin-kucuk-2">0 LTC</h3>
    <a href="#" class="custom-button wow fadeInUp cabinet" data-wow-delay="0.6s"
      style="visibility: visible; animation-delay: 0.6s; animation-name: fadeInUp;">Deposit
→</a>
    <p class="ozellik-kutu-iletisim--yazi">Referral: 0 LTC</p>
    <p class="ozellik-kutu-iletisim--yazi">Withdrawal: 0 LTC</p>
    <p class="ozellik-kutu-iletisim--yazi">Profit: 0 LTC</p>
  </div>
</div>
</div>
<section class="lastOperations">
  <h2 class="h2-baslik-hizmetler-2 wow fadeInUp">LAST TRANSACTIONS</h2>
  <div class="col-lg usttablo">
    <div class="box wow flipInY" data-wow-delay="0.7s"
      style="visibility: visible; animation-delay: 0.7s; animation-name: flipInY;">
      <span></span>
      <span></span>
      <span></span>
      <span></span>
      <div class="content1">
        <div class="paketler__pr-kutu">
```

```
<div class="iconw">
  <i class="flaticon-badge"></i>
</div>
<h3 class="baslik-3 h-yazi-margin-kucuk">You have not made any
transactions</h3>
</div>
</div>
</div>
</div>
</section>
```

Фрагмент CSS коду:

```
.inputLabel {
  color: #D9B08C;
  margin: 0;
  font-weight: 600;
  font-size: 24px;
}

.radioLabel {
  position: relative;
  font-size: 18px;
  line-height: 24px;
  color: #D9B08C;
  font-weight: 400;
  letter-spacing: .05em;
  cursor: pointer;
}

.cabMoneyMoveResult,
.cabMoveDataItem {
  padding-bottom: 10px;
```

```
}
```

```
#add_Sum {  
    margin-bottom: 10px;  
}
```

```
.iconn.icon-profile {  
    font-size: 24px;  
}
```

```
.tablo-1-ve-3.wow.cabinet {  
    padding-bottom: 20px !important;  
}
```

```
.custom-button.wow.fadeInUp.cabinet {  
    display: inline !important;  
    padding: 10px 40px;  
}
```

```
.crypto {  
    padding-bottom: 10px;  
    width: 48px;  
}
```

```
.lastOperations {  
    margin-bottom: 40px;  
    margin-top: 40px;  
}
```

```
.withd {  
    font-size: 24px;  
    color: #D9B08C;
```

```
}
```

```
.payItem {  
  display: flex;  
}
```

```
.filters {  
  display: flex;  
}
```

```
.operationItems {  
  display: flex;  
  flex-wrap: wrap;  
  margin: 0;  
  margin-bottom: 100px !important;  
}
```

```
.partnersHeader {  
  color: #D1E8E2;  
  font-weight: 500;  
}
```

```
.invited {  
  font-size: 20px;  
  color: #D1E8E2;  
  margin-top: 20px;  
}
```

```
.iletisim-form-alani.settings {  
  /* padding-top: 10rem; */  
}
```

```
.settingsHeader {  
  font-size: 24px;  
  color: #D9B08C;  
  text-align: center;  
  font-weight: 600;  
}
```

```
.settingsHeader.sec {  
  padding-top: 60px;  
}
```

```
.payIcon.settings {  
  margin-bottom: 10px;  
}
```

```
.walletName {  
  color: #2c3531;  
  margin-bottom: 10px;  
}
```

```
.logItems {  
  text-align: center;  
  display: flex;  
  gap: 30px;  
  justify-content: center;  
}
```

```
.logItem {  
  text-align: center;  
  color: #D9B08C;  
  font-size: 18px;  
  margin: 0;
```

```

}

.logItem:hover {
  color: #2c3531;
}

@media (max-width: 901px) {
  .iletisim-form-alani.register {
    margin-top: -19vh !important;
  }
}

```

Фрагмент JS коду, що відповідає за анімації переходу, задній фон, слайдери та кнопку меню у мобільній версії:

```

// HAMBURGER MENU
var main = function () {
  $('hamburger-menu').click(function () {
    $('.side-widget').animate({
      left: "0px"
    }, 200, function () {

      $(document).on("click.menu", function (event) {
        var target = $(event.target);
        console.log(target);
        if (!target.closest(".side-widget").length || target.closest(".closed").length) {
          closeMenu(function () {
            $(document).off("click.menu");
          });
        }
      });
    });
  });
});

```

```
// DATA BACKGROUND IMAGE
var pageSection = $("*");
pageSection.each(function (indx) {
  if ($(this).attr("data-background")) {
    $(this).css("background", "url(" + $(this).data("background") + ")");
  }
});
```

```
// DATA BACKGROUND COLOR
var pageSection = $("*");
pageSection.each(function (indx) {
  if ($(this).attr("data-background")) {
    $(this).css("background", $(this).data("background"));
  }
});
```

```
// WOW ANIMATION
wow = new WOW({
  animateClass: 'animated',
  offset: 0
});
wow.init();
```

```
// CAROUSEL CLASSES SLIDER
var swiper = new Swiper('.carousel-classes', {
  slidesPerView: '3',
  spaceBetween: 10,
  loop: 'true',
  pagination: {
    el: '.swiper-pagination',
    clickable: true,
  },
});
```

```
breakpoints: {
  640: {
    slidesPerView: 1,
    spaceBetween: 15,
  },
  768: {
    slidesPerView: 2,
    spaceBetween: 30,
  },
  1024: {
    slidesPerView: 3,
    spaceBetween: 30,
  },
}
});

// MAIN SLIDER
var swiper = new Swiper('.main-slider', {
  slidesPerView: '1',
  speed: 900,
  loop: true,
  effect: 'fade',
  opacity: 1,
  autoplay: {
    delay: 15000,
    disableOnInteraction: false,
  },
  navigation: {
    prevEl: '.button-prev',
    nextEl: '.button-next',
  },
  pagination: {
```

```
el: '.swiper-pagination',  
clickable: true,  
},  
});
```

```
// PRELOADER
```

```
$(window).load(function () {  
  $("body").addClass("page-loaded");  
});
```

Сторінки прототипу системи

Розглянемо прототипи сторінок інформаційної системи «Криптоінвестиції» що представлені на Рисунок В.1 – В.29.



Рисунок В.1 – Головна сторінка

Trusted by the world's best

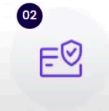
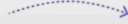


How it works?

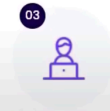
10,000+ unique online course list designs



01
Browse courses from our expert contributors.



02
Purchase quickly and securely.



03
That's it! Start learning right away.

Рисунок В.2 – Сторінка про нас



Welcome to **StudyLab** Enhance your skills with best Online courses

You can start and finish one of these popular courses in under a day – for free! Check out the list below.. Take the course for free

Neque convallis a cras semper auctor. Libero id faucibus nisl tincidunt egetnvalis a cras semper auctonvalis a cras semper aucto. Neque convallis a cras semper auctor. Liberoe convallis a cras semper atincidunt egetnval

Start Learning For Free

Рисунок В.3 – Сторінка «About Us»

Our Most Popular **Courses**

10,000+ unique online course list designs



Ratings ★★★★★ 4/5

Advanced Certificate in Digital Marketing and Communication

6 lessons 3h 56m Beginner

Ritesh Sherekar ₹2,299 ₹499

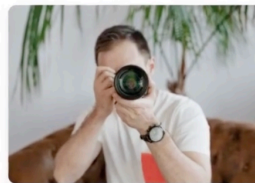


Ratings ★★★★★ 4.5/5

Online Marketing: SEO & Social Media Marketing Strategy

6 lessons 3h 56m Beginner

Ritesh Sherekar ₹2,299 ₹499



Ratings ★★★★★ 4/5

Photoshop Master Course: Beginner to Photoshop Pro

6 lessons 3h 56m Beginner

Ritesh Sherekar ₹2,299 ₹499



Ratings ★★★★★ 4/5

Complete Guitar Lessons System - Beginner to Advanced

6 lessons 3h 56m Beginner

Yash Wasankar ₹2,299 ₹499

Рисунок В.4 – Випадаючий список «Courses»

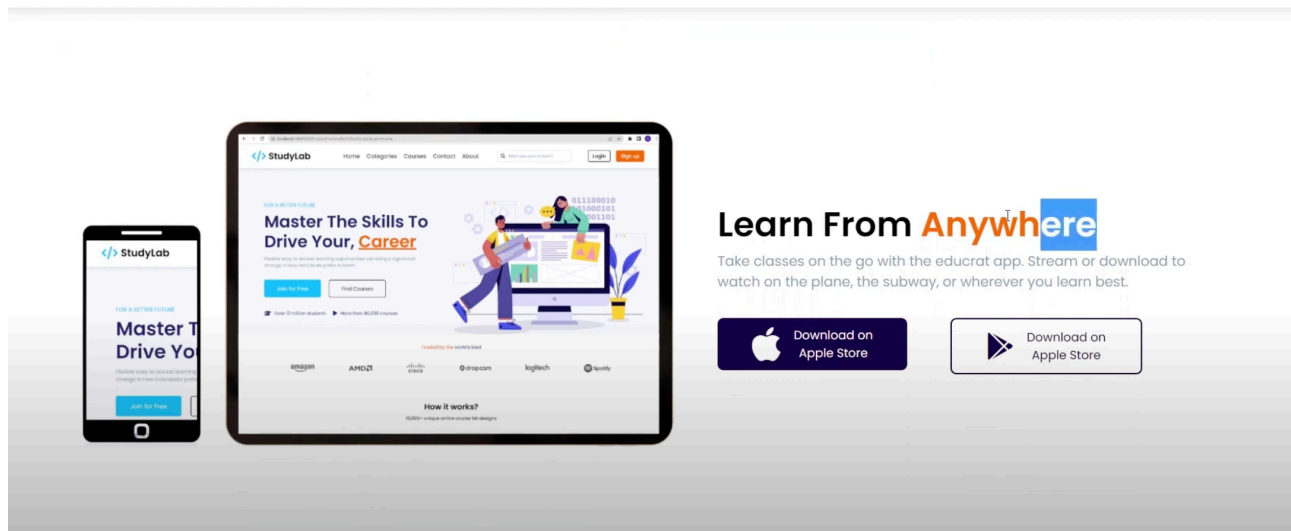


Рисунок В.5 – Завантажити додаток на телефон

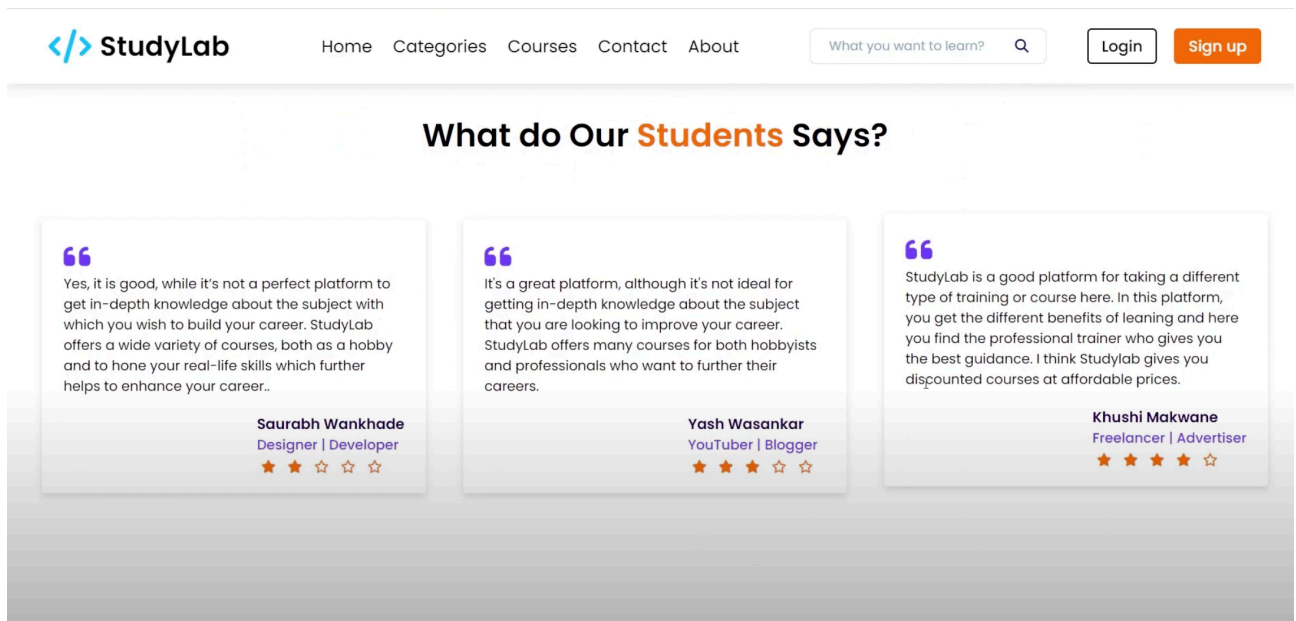


Рисунок В.6 – Сервіс «Відгуки»

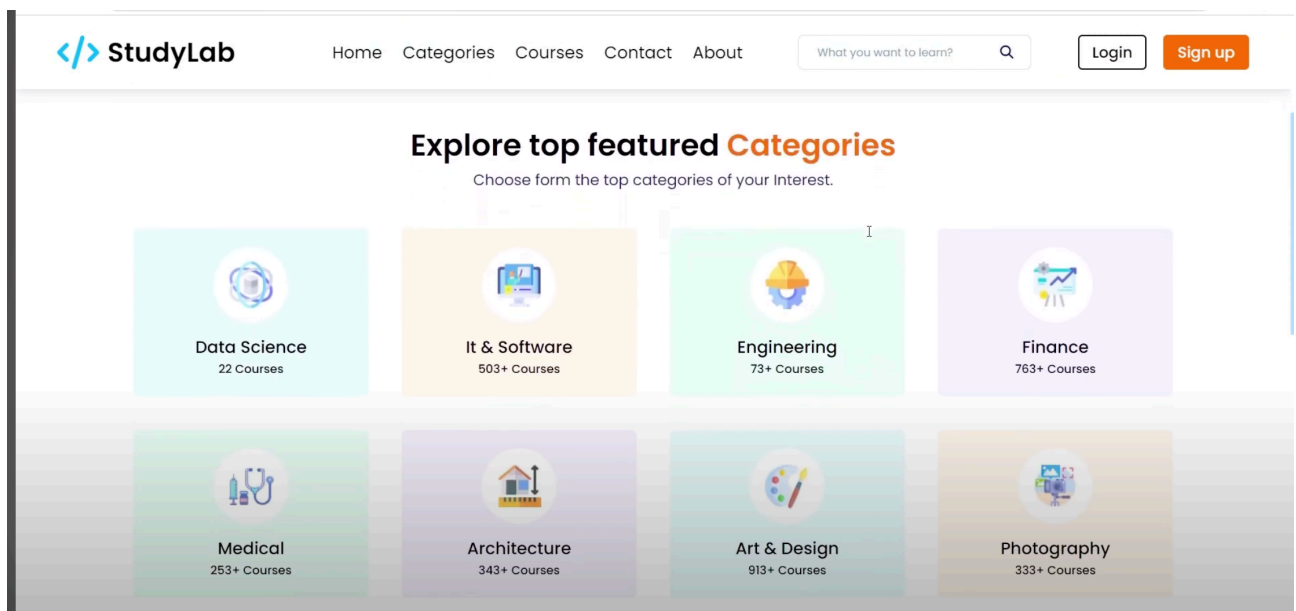


Рисунок В.7 – Сервіс «Categories»

Our Most Popular Courses

10,000+ unique online course list designs






| | | | |
|---|---|---|---|
|  <p>Ratings ★★★★★ 4/5</p> <p>Advanced Certificate in Digital Marketing and Communication</p> <p>6 lessons 3h 56m Beginner</p> <p>Ritesh Sherekar ₹2,299 ₹499</p> |  <p>Ratings ★★★★★ 4.5/5</p> <p>Online Marketing: SEO & Social Media Marketing Strategy</p> <p>6 lessons 3h 56m Beginner</p> <p>Ritesh Sherekar ₹2,299 ₹499</p> |  <p>Ratings ★★★★★ 4/5</p> <p>Photoshop Master Course: Beginner to Photoshop Pro</p> <p>6 lessons 3h 56m Beginner</p> <p>Ritesh Sherekar ₹2,299 ₹499</p> |  <p>Ratings ★★★★★ 4/5</p> <p>Complete Guitar Lessons System - Beginner to Advanced</p> <p>6 lessons 3h 56m Beginner</p> <p>Yash Wasankar ₹2,299 ₹499</p> |
|---|---|---|---|

Рисунок В.9 – Сервіс «Courses»



[Home](#)
[Categories](#)
[Courses](#)
[Contact](#)
[About](#)

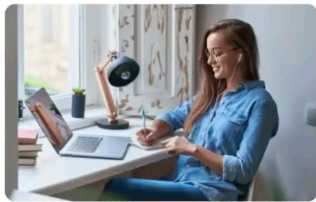
BEST SELLER
NEW
POPULAR

Advanced Certificate in Digital Marketing and Communication

Top Digital Marketing Course | Facebook Ads, Google/YouTube, Display/Video & more|Grow your career (German/Portuguese)

4.5 ★★★★★ (1991) 853 enrolled on this course Last updated 11/2021

Teached By :  Rishikesh Shah



₹499 ~~₹2,299~~ 78% off

| | |
|------------------------|----------|
| 📁 Modules | 30 |
| 📊 Skill level | Beginner |
| 📄 Certificate | Yes |
| ∞ Full lifetime access | Yes |

Рисунок В.10 – Сервіс «Courses»

Короткий звіт подібності



Ім'я користувача: **Комп'ютерної математики та інформаційної безпеки...** ID перевірки: **1016335027**
Дата перевірки: **08.06.2024 13:31:14 EEST** Тип перевірки: **Doc vs Internet + Library**
Дата звіту: **08.06.2024 14:02:52 EEST** ID користувача: **100005746**

Назва документа: **Диплом_ГерасименкоОО**
Кількість сторінок: **51** Кількість слів: **7077** Кількість символів: **57403** Розмір файлу: **10.15 MB** ID файлу: **1016135652**

8.61% Схожість

Найбільша схожість: 2.74% з джерелом з Бібліотеки (ID файлу: 1016114611)

| | | |
|----------------------------|-----|-------------|
| 2.8% Джерела з Інтернету | 214 | Сторінка 53 |
| 7.01% Джерела з Бібліотеки | 295 | Сторінка 54 |

0% Цитат

- Вилучення цитат вимкнене
- Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

| | |
|------------------|---|
| Замінені символи | 1 |
|------------------|---|