До захисту допущено:

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_ Ігор ПАРХОМЕЙ

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2023 р.

**Дипломна робота**

**на здобуття ступеня бакалавра**

**за освітньою програмою «Інженерія програмного забезпечення інформаційних систем»**

**спеціальність 121 «Інженерія програмного забезпечення»**

**на тему: «Система моніторингу аналітичних**

**даних навчального закладу»**

Виконав (-ла):

студент (-ка) IV курсу, групи ЗПІ-зп01

Корнієнко Микита Валерійович

Керівник:

ст. викладач кафедри ІСТ,

Корнага В.І.

Рецензент:

с.н.с., к.т.н., с.н.с. ІК НАНУ

Писаренко Юлія Валеріївна

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Слухач \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра технічної кібернетики**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 121«Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення інформаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_ Ігор ПАРХОМЕЙ

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2023 р.

**ЗАВДАННЯ**

На дипломний проєкт студенту

Корнієнко Микиті Валерійовичу

1. Тема проєкту «Система моніторингу аналітичних даних навчального закладу», керівник проєкту доц., к.т.н., с.н.с. Корнага В.І., затверджені наказом по університету від …
2. Термін подання студентом проєкту …
3. Вихідні дані по проєкту: наукова література з теми, існуючі реалізації системи на різних платформах, технічне завдання на розробку комп’ютерної програми для створення програми моніторингу аналітичних даних навчального закладу.
4. Зміст пояснювальної записки:

Перелік скорочень

Вступ

1. Опис предметної області
2. Методичне та інструментальне забезпечення
3. Реалізація системи

Висновки

1. Графічний матеріал, а саме: Схема архітектури застосунку (А3); Схема бази даних (А3); Діаграма діяльності (А3); Діаграма класів (А3); Процес роботи із застосунком (А3); Схеми графіків статистичних даних застосунку (А3).
2. Дата видачі завдання «» …

Календарний план

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Студент Корнієнко М.В.

Керівник проєкту Корнага В.І.

**АНОТАЦІЯ**

Зростаюча кількість даних, що генерується навчальним закладом, ставить перед його адміністрацією виклики щодо ефективного управління цими даними та отримання висновків, щодо них.

Для полегшення роботи з персональними даними, візуалізації багатьох статистичних показників, що їх генерує будь який навчальний заклад я і розробив свій проєкт – електронну систему моніторингу.

Із статистичних показників, що поступатимуть кожен навчальний період, таких як оцінки, персональні дані студентів, фінансова звітність, облік студентів іноземців, тощо – мій проєкт дозволяє побудувати корисну візуалізацію у вигляді зведених таблиць, графіків, pie-чартів, діаграм.

Ці дані будуть корисними абітурієнтам, батькам студентів, професорам, керівництву навчального закладу, ну і звісно ж – самим студентам, які зможуть краще відслідковувати свою успішність, рейтинг та свою фінансову звітність, щодо навчального закладу.

Додаток може бути використаний для аналізу роботи будь-якого навчального закладу, однак нашим тестовим навчальним закладом буде коледж, де вивчаються переважно технічні дисципліни, а також бухгалтерія.

Ключові слова: Аналітичні дані, статистика, візуалізація, агрегація даних, MongoDB, React, Express.js, Node.js, Redux, Compass.

Розмір пояснювальної записки – 64 аркуша, містить 24 ілюстрації, 4 таблиці, 9 додатків.

**ANNOTATION**

The increasing amount of data generated by educational institutions presents challenges for their administration in terms of effectively managing and deriving insights from this data. To facilitate the handling of personal data and visualize various statistical indicators generated by any educational institution, I have developed an electronic monitoring system as my project.

By incorporating statistical indicators received each academic period, such as grades, students' personal data, financial reports, international student records, and more, my project allows for the construction of useful visualizations in the form of consolidated tables, charts, pie charts, and diagrams.

These data will be valuable to prospective students, parents, professors, institution management, and, of course, the students themselves, enabling them to better track their performance, ranking, and financial reports related to the educational institution.

While the application can be used to analyze the operations of any educational institution, our testing institution will be a technical college with a focus on technical disciplines and accounting.

Keywords: Analytical data, statistics, visualization, data aggregation, MongoDB, React, Express.js, Node.js, Redux, Compass.

The size of the explanatory note is 64 pages, including 24 illustrations, 4 tables, and 9 appendices.

**Пояснювальна записка**

**до дипломного проєкту**

**на тему: «Система моніторингу аналітичних даних навчального закладу»**

Київ – 2023 року

ЗМІСТ

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

7

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп01.190БАК.005.ПЗ

Т

ПЕРЕЛІК СКОРОЧЕНЬ......................................................................................... 9

ВСТУП .................................................................................................................. 10

РОЗДІЛ 1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ .................................................... 12

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

8

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп01.190БАК.005.ПЗ

Т

ПЕРЕЛІК СКОРОЧЕНЬ

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

9

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

UI - Інтерфейс користувача (User Interface)

UX - Взаємодія користувача (User Experience)

API - Інтерфейс програмного забезпечення (Application Programming Interface)

CRUD - Операції створення, читання, оновлення та видалення (Create, Read, Update, Delete)

JWT - Токен аутентифікації JSON (JSON Web Token)

DB - База даних (Database)

ORM - Об'єктно-реляційне відображення (Object-Relational Mapping)

REST - Представлення стану передачі (Representational State Transfer)

SPA - Односторінкова програма (Single-Page Application)

CLI - Інтерфейс командного рядка (Command Line Interface)

NPM - Менеджер пакетів Node.js (Node Package Manager)

JSX - Розширена синтаксис JavaScript (JavaScript XML)

SEO – Оптимізація у пошуку

ВСТУП

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

10

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

В останні роки у всьому світі сфера освіти змінюється – все більшої ваги набуває онлайн навчання. Причиною цього є розвиток засобів комунікації – все більше поширюється швидкісний інтернет, який стає доступним навіть у віддалених куточках світу. Також онлайн-освіта дозволяє ознайомитись з певним навчальним курсом без відвідування, що робить її набагато більш доступною, ніж класична очна освіта, однак такий вид освіти має і свої неочікувані недоліки, а саме: контакт з викладачем у реальному житті, відвідування лекції чи семінару безпосередньо, а не віддалено, дають можливість студенту краще засвоїти навчальний матеріал, менше відволікатися, більше зосередитись на предметі вивчення.

Крім того, класичні навчальні заклади будь-якого рівня акредитації дозволяють студенту зустріти однодумців, знайти людей зі спільними захопленнями, конкурувати за рейтинг, а також – краще проводити час.

Саме тому, навчальні заклади є ключовими установами, що забезпечують освіту та розвиток суспільства. Ми можемо із впевненістю стверджувати, що вони з нами надовго, і, що навчання у них завжди буде привілеєм.

Якість навчального процесу та ефективність роботи навчальних закладів є питаннями, які стоять перед освітніми системами та органами управління освітою. Оцінка роботи навчальних закладів та аналіз статистичних даних про їхню діяльність є важливими інструментами для забезпечення якості освіти та прийняття обґрунтованих управлінських рішень.

Ця дипломна робота присвячена розробці "Системи статистичної оцінки роботи навчального закладу", та має на меті створити зручний та ефективний інструмент для збору, аналізу та візуалізації статистичних даних, пов'язаних з роботою навчального закладу.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

11

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Основною метою цієї системи є забезпечення доступу до об'єктивних та надійних даних про різні аспекти роботи навчального закладу, таких як академічні показники, фінансові дані, аналіз успішності учнів, відвідуваність, статистика викладачів та інші ключові показники. Ця система дозволить адміністраторам, вчителям, батькам та іншим зацікавленим сторонам отримувати об'єктивну інформацію та здійснювати аналіз даних для покращення якості освіти.

Для реалізації даної системи будуть використані сучасні технології, зокрема фреймворк React для розробки користувацького інтерфейсу, платформа Node.js для створення серверної частини, а також нереляційна база даних Mongo.db для зберігання та управління великим обсягом статистичних даних.

У даній дипломній роботі будуть розглянуті основні принципи проектування та реалізації системи статистичної оцінки роботи навчального закладу. Буде проведено аналіз вимог до системи, розроблено архітектуру, реалізовано необхідні функціональні модулі та проведено тестування системи на практиці.

Остаточним результатом цієї дипломної роботи буде працездатна та ефективна система статистичної оцінки роботи навчального закладу, яка допоможе удосконалити процес прийняття управлінських рішень, забезпечить об'єктивний аналіз та покращення якості освіти.

РОЗДІЛ 1. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

12

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

* 1. Загальний огляд проблеми

У сучасному освітньому середовищі, ефективне управління навчальними закладами та забезпечення якості освіти стають вирішальними завданнями. Адміністрація навчальних закладів, вчителі, батьки та студенти потребують доступу до достовірної та об'єктивної інформації щодо роботи закладу для прийняття обґрунтованих управлінських рішень та покращення якості навчання.

Також студентам могло б бути цікаво відслідковувати свої оцінки то оцінки своїх товаришів онлайн, що, без наявності спеціальної системи, буде неможливо.

Система оцінок має показувати загальний прогрес наших учнів, динаміку їх успішності, різницю в успішності по факультетам, за статтю, за курсом і за країною.

Крім того, батьки студентів та самі студенти часто хочуть висловити свою думку, стосовно того чи іншого питання. Я ставлю собі за мету реалізувати таку можливість у моєму додатку, за допомогою системи опитування, що доступна кожному зареєстрованому користувачу.

Невід’ємною частиною будь-якого навчального закладу є фінансова звітність. Не дивлячись на те, що більша її частина є, авжеж, конфіденційною інформацією, в інтересах відкритості та прозорості, навчальний заклад залишає за собою право ознайомити всіх зацікавлених осіб у деяких аспектах своєї фінансової діяльності, для чого у нашій системі є загальнодоступна фінансова інформація, реалізована у вигляді графіків.

У багатьох навчальних закладах є учні з інших країн світу. Так, до нас часто приїздять вчитись студенти з Африки, Близького Сходу, країн колишнього СРСР. Наші студенти, в свою чергу, нерідко вступають до навчальних закладів країн Східної та Центральної Європи, Німеччини, Франції і США.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

13

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Для контролю за кількістю іноземців, їх фінансовими транзакціями та країнами походження я маю намір у даній роботі створити інтерактивну мапу, де буде видно, з якої країни до нашого навчального закладу приїхала певна кількість людей.

Було б також непогано, щоб можна було побачити, які саме країни є, скажімо, найкращими фінансовими партнерами нашого навчального закладу. Для цього у нас в застосунку буде також специфічний графік.

Таким чином, у даному застосунку ми проаналізуємо велику кількість статистичних показників, які є характерними для будь-якого навчального закладу.

* 1. Огляд наявних рішень застосунків для аналізу статистичних даних навчальних закладів

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

14

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Існує досить багато рішень для навчальних закладів, що дозволяють демонструвати та аналізувати показники їх роботи.

Переважно такі рішення є платними та робляться під замовлення навчального закладу.

Демонстрація статистичних даних є у таких системах, зазвичай, лише частиною функціоналу. У них є також, як правило, є функціонал дистанційного навчання, кабінет студента із завданнями, чат із вилкадачем чи адвайзером. У своїй роботі я не ставлю собі таких цілей і фокусуюсь переважно на статистичних даних, їх зборі та демонстрації.

Статистичні дані у інших застосунках показуються, але я ставлю собі за мету, дослідити якомога більше їх різновидів і продемонструвати якнайбільшу кількість різних показників.Але розглянемо і інші застосунки із подібними можливостями.

1.2.1. Застосунок від PowerSchool

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

15

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

PowerSchool **–** це освітня технологічна платформа, власником якої є PowerSchool Group LLC.

Ця платформа є однією із перших подібних платформ в історії, вона написана на мові Java у 1999 році та має останній реліз у 2017 році.

У даному застосунку широко реалізовано отримання та аналіз статистичної інформації, переважно, по успішності учнів. У своїй роботі вони спираються на Microsoft Power BI.

Аналізу у цьому застосунку підлягає широкий спектр даних щодо успішності студентів.

Зображення, що містить текст, програмне забезпечення, Комп’ютерна піктограма, знімок екрана

Автоматично згенерований опис

Рис. 1.2. PowerSchool. Інтерфейс учня

Недоліками даного застосунку є певна архаїчність технологій, відсутність стабільного оновлення, висока ціна, присутність тільки на освітньому ринку США.

* 1. .2 Застосунок Jenzabar

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

16

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Ця система спрямована на навчальні заклади різних рівнів та надає функціонал для збору, аналізу та відображення статистичних даних.

Jenzabar Analytics має в собі три основних компоненти: Program Insights Model, Financial Health Model, і Data Cloud.

The Program Insights Model – дає інформацію про успішність студента, його фінансовий стан, також про його здоров’я.

The Financial Health Model – робить можливим дізнатись все про фінансову складову роботи навчального закладу, зокрема розглядаючи такі показники, як Composite Financial Index, net operating revenue, та інші індикатори.

The Data Cloud – збирає, агрегує та зберігає інші дані навчального закладу.

Зображення, що містить текст, Веб-сторінка, Веб-сайт, програмне забезпечення

Автоматично згенерований опис

Рис. 1.3. Jenzabar. The Financial Health Model

1.2.3 Застосунок Ellucian Banner

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

17

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Ellucian Banner – освітня система, що дозволяє покращити роботу будь-якого навчального закладу.

Вона є комплекною і включає в себе широкий набір різноманітних функцій управління навчальним закладом і є корисною як студентам, так і керівникам.

Студентам вона дозволяє отримати велику кількість даних про свій навчальний прогрес, керівникам – про фінансову частину і роботу персоналу.

Використовується в основному – в США, розповсюджується по моделі підписки для навчальних закладів, є платною.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Веб-сторінка

Автоматично згенерований опис

Рис. 1.3. Ellucian Banner. Особистий кабінет студента

Має веб-інтерфейс, та мобільний застосунок.

Недоліки має ті ж – закритість моделі розповсюдження, висока ціна.

1.2.4 Застосунок HeRo Study Space

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

18

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

HeRo Study розробляє SaaS рішення для цифрової трансформації ВНЗ з фокусом на автоматизацію ключових бізнес-процесів.

Компанія працює у сфері EdTech з 2016 року, коли її засновник Антон Сіленко, разом із невеликою командою розробників створив застосунок Hero Study Space.

Цей застосунок є комплексною освітньою програмою, де представлено багато модулів для комунікації між студентом та навчальним закладом, є блок домашніх завдань, блок комунікації з власним едвайзером та інше.

Статистичні дані доступні як для студентів, так і для керівників навчального закладу. Переважно компанія співпрацює з ВНЗ Казахстану, України та інших країн колишнього СРСР.

Тривалий час я і сам був Front-end розробником у даній компанії.

Зображення, що містить текст, знімок екрана

Автоматично згенерований опис

Рис. 1.4. Hero Study Space. Особистий кабінет абітурієнта

* 1. Постановка задачі

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

19

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Після того, як я детально проаналізував зарубіжні та вітчизняні рішення, я вирішив розробити своє.

Мій застосунок я назвав «EduStatsTracker». Він також збиратиме достатньо великий обсяг даних, які генерує навчальний заклад та демонструватиме аналітику у вигляді простих, доступних графіків та чартів.

Вимоги до застосунку:

* Має надавати доступ до великого набору статистичних даних навчального закладу;
* Повинен містити свій кабінет, з авторизацією та різними рівнями доступу;
* Повинен бути mobile-responsive;
* Повинен бути універсальним;
* Мови інтерфейсу – Українська і Англійська;

Висновки до розділу

Я розглянув деякі наявні на ринку рішення застосунків, подібних до того, що я створив. Розглянув їх переваги та недоліки та визначив той напрям, у якому я буду рухатись для створення свого унікального рішення, що буде враховувати функціонал схожих продуктів та намагатиметься повторити та покращити його.

РОЗДІЛ 2. МЕТОДИЧНЕ ТА ІНСТРУМЕНТАЛЬНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Опис задач застосунку

Після огляду всіх подібних застосунків, було проаналізовано завдання, що стоятимуть при розробці застосунку «EduStatsTracker» і було визначено набір дій, що їх він буде виконувати:

* Авторизація і аутентикація; Власний кабінет;
* Перегляд різноманітної статистики всього університету;
* Перегляд власних успіхів та досягнень;
* Перегляд досягнень інших студентів;
* Перегляд фінансової звітності;
* Перегляд інформації по іноземним студентам та країнам, з яких вони прибули;

Оскільки такі завдання ставлять нас перед вибором певних технологій, які буде висвітлено у наступному розділі даної записки.

2.2. Сучасні технології веб-додатків

Оскільки я хочу зробити програму якомога доступнішою для широкого кола користувачів (абітурієнтів, студентів, персоналу, керівництва, батьків), я вирішив розробляти її у вигляді веб-застосунку, який завжди можна розширити на інші платформи (десктоп, мобільний додаток).

Так як ми не маємо у наших планах великої кількості взаємодій у реальному часі, що потребували б швидких, динамічних відповідей, а всі наші дії у

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

14

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

застосунку будуть досить стандартизованими, ми не потребуємо нічого більше, крім стандартної REST-архітектури, яку я і вважаю оптимальною для побудови даного додатку.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Зображення, що містить текст, знімок екрана, Шрифт, схема

Автоматично згенерований опис

Рис. 2.1. Схема стандартної REST API

Наша система також матиме front-end, back-end і базу даних.

Front-end створений на фреймворку React. Back-end – монолітний і виконаний на Node.js із фреймворком Express. База даних – нереляційна Mongo.db. Причини вибору цих технологій – далі.

2.2. Сучасні підходи до рендерінгу веб-додатків

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

**Рендерінг** – процес перетворення даних та програмного коду у кінцевий html, що його буде бачити користувач.

Цей процес може відбуватись як на сервері, так і на стороні клієнта, або поєднувати певним чином ці варіанти.

Розглянемо спочатку класичну стратегію – Статичний вебсайт. Такі вебсайти були популярними з часів Web 1.0, але й зараз такі сайти мають місце у Web.

При роботі зі статичним веб-сайтом, кожна сторінка генерується на сервері заздалегідь і повертається користувачу для подальшої взаємодії.

На даний час існують фреймворки, що допомагають у створенні таких вебсайтів – наприклад Jekyll, Hugo та 11ty.

Такий тип рендерінгу не дуже придатний для веб-сайтів, де дані часто змінюються. Такі сайти не є дуже динамічними, кожен клік означає нове завантаження сторінки, що погіршує UX.

Multi-page rendering – стратегія рендерінгу, при якій html та дані об’єднуються на сервері в момент надходження нового запиту від користувача.

Такий спосіб використовується сьогодні у багатьох e-commerce проєктах, наприклад – на сайті компанії Amazon, де ми можемо побачити, як на кожен наш клік виводиться нова, динамічно-згенерована сторінка.

На сьогодні існує велика кількість фреймворків, що дають можливість створювати додатки – це і Ruby on Rails, i Django, і PHP-фреймворк Laravel.

Такий підхід був розповсюджений до появи Iphone, коли користувачі збагнули, що завантаження після кожного кліку не є оптимальним з точки зору UX.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Однак, слід зазначити, що перевагою даного підходу є досить високий рівень SEO.

Зображення, що містить текст, знімок екрана, Шрифт, логотип

Автоматично згенерований опис

Рис. 2.2. Схема рендерінгу багатосторінкового застосунку

Односторінковий застосунок, або SPA(Single page application) – Роком народження даної парадигми рендерінгу вважається 2010, коли вийшла перша версія Angular.js, і, згодом, - у 2011 – React.js.

Сайти, чий рендерінг відбувається у цей спосіб, починають роботу з лише однієї html-сторінки – так званої «обгортки» і, далі, використовують мову Javascript для рендерінгу UI-елементів, а також – для HTTP-запитів.

Не дивлячись на те, що це лише один документ, у ньому можуть бути «маршрути», але вони не вказують на шлях до серверу – їх обробляє Javascript.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

У такий спосіб ми робимо перехід між «сторінками» нашого застосунку непомітним для користувача.

Недоліками даного підходу є досить великий розмір сторінки «обгортки», що знижує швидкість першого завантаження. Також, пошукові двигуни поки що не навчились адекватному пошуку по таким «віртуальним» сторінкам, тому SEO є ускладненим і ці два недоліки знижують привабливість таких застосунків для комерційних проєктів (хоча, все ж, інтернет-магазинів, зроблених на базі SPA існує достатньо багато).

Зображення, що містить текст, знімок екрана, схема, число

Автоматично згенерований опис

Рис. 2.2. Схема рендерінгу односторінкового застосунку

Рендерінг на стороні сервера із «зволожуванням» - «зволожування», або hydration – процес наповнення об'єкту або екземпляру класа – даними.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

У зв’язку з веб-розробкою та рендерінгом, термін використовується для описання заповнення «обгортки» - функціоналом SPA.

У цій парадигмі – перший запуск застосунку, на відміну від SPA, не створює «обгортку» на рівні клієнта, а повертає її із сервера, а вже Javascript, що у ній наявний, заповнює її контентом на функціоналом.

Цей підхід є «кращим із двох світів», бо він запобігає повільному першому завантаженню, що є характерним для SPA, а також при такому підході, пошукові двигуни набагато краще знаходять подібні сторінки.

Цей підхід зараз є найбільш популярним і використовується сучасними «мета-фреймворками», такими як Nuxt.js, Next.js, SvelteKit та Angular Analog.

Варіацією минулого підходу є Статична генерація сайтів зі «зволоженням».

Така парадигма дозволяє створити всі необхідні сторінки сайту заздалегідь, помістити їх на сервер і віддавати користувачу за необхідністю, всі ці сторінки будуть перетворені на Javascript і будуть змінюватись відповідно до дій користувача.

Такі вебсайти називаються JamStack. Їх створюють на тих самих метафреймворках, що і попередній тип, і, як і для минолого виду рендерінгу, недоліком є необхідність мати потужний сервер, тільки на цей раз – сервер має бути із великим запасом пам’яті, щоб вмістити сторінки для усіх юзерів.

Поступова статична регенерація – як і в попередніх видах рендерінгу сервер повертає первинну сторінку «обгортку», але на цей раз Javascript слідкує за завантаженнями на сторінці, їх швидкістю та часом, коли кеш стає не валідним. Якщо сайт завантажується занадто довго, або інвалідизується кеш – повертається нова сторінка-«обгортка».

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Часткове «зволоження» - на відміну від усіх попередніх видів рендерінгу, включно з SPA, при цьому варіанті, ми проводимо наповнення сторінки даними не одразу, а частково – коли користувач робить якісь дії, наприклад – рухається по сторінці.

«Острови» - при цьому варіанті ми проводимо «зволоження» лише тих ділянок веб-сайту, з якими користувач активно взаємодіє, створючи таким чином «острови» гідратації. Це один із найефективніших видів рендерінгу, оскільки заповнення даними відбувається динамічно тільки тих компонентів застосунку, з якими взаємодіють тут і зараз. Таким чином, якщо ви створили статичну сторінку, в якій немає взаємодій, не буде відбуватись зайвої роботи Javascript, на відміну від SPA.

Фреймворк Astro працює таким чином і демонструє чудові показники швидкості роботи застосунків.

Потокова SSR – новий вид рендерінгу, що доступний у Next.js 13-ї версії.

«Зволоження» відбувається у вигляді інтерактивних потоків, що надходять постійно від серверу.

А чи можна не заповнювати сторінку при первинному завантаженні взагалі, тобто – обійтись без «зволоження»?

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Так. Остання парадигма рендерінгу – продовжуваність (Resumability) – піонером у даному виді рендерінгу є фреймворк qwik. За його розробку відповідають розробники Angular.

При роботі цього фреймворку, все, що відбувається у веб-застосунку, серіалізується у вигляді HTML, а Javascript розбивається на велику кількість фрагментів. Тому перше завантаження – це завантаження порожньої статичної сторінки, а увесь javascript завантажується поступово у фоновому режимі.

Ми ознайомились з усіма наявними на даний час парадигмами рендерінгу контенту у веб. Я впевнений, що таких варіантів згодом буде ще більше, однак, обираючи зараз, я зупинюсь на SPA. Такий вибір я роблю тому, що у даному застосунку час першого завантаження (так званий FCP – перше малювання контенту) не є дуже важливим і при наявних розмірах застосунку навряд чи змінитися в «червоний» бік.

Також, ми не ставимо собі на меті піднятись у перші щаблі пошуку Google. Якщо наш застосунок колись прийме вигляд промислового продукту він все одно буде лише для підписників (як і розглянуті раніше аналоги), тому ефективного SEO він не потребує.

Розроблятимемо Front-end на фреймворку React. Зберемо за домомогою Vite.

React вибраний з-поміж інших фреймворків через швидкість розгортання, простий синтаксис, та особисті преференції.

2.3. Огляд бібліотеки React

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Бібліотека, а потім і повноцінний фреймворк, розроблений компанією Facebook у 2013 році.

Його можна назвати найвпливовішою бібліотекою сучасності.

Основним принципом його роботи є поділ UI на обмежені логічно та функціонально компоненти. Кожен такий компонент являє собою ніщо інше, аніж Javascript-функцію.

Така функція має специфічне значення, що обов’язково повертається із return – а саме – JSX. Це об’єкт із специфічним синтаксисом, що візуально є схожим на HTML, а також може містити певні функції та методи, що будуть обробляти події, що відбуватимуться з цим компонентом після рендерінгу.

Також, ключовими для даного фреймворку є поняття State та Hooks.

State – стан компоненту, який відслідковується екземпляром React, а зміни у ньому відбуваються виключно через спеціальні функції – хуки. При кожній зміні стану відбувається повторний рендерінг компоненту.

На відміну від інших фреймворків, React містить в собі лише базову систему рендерінгу jsx компонентів, та декілька стандартних функцій-хуків, що змінюють стан компонента. Все інше, а саме – маршрутизацію, анімації, менеджмент глобального стану застосунку віддається стороннім бібліотекам, яких є велика кількість.

Зображення, що містить текст, знімок екрана, програмне забезпечення, Шрифт

Автоматично згенерований опис

Рис. 2.3. Схема типового класового JSX-компоненту

На даній схемі зображений класовий React-компонент. Вони є більш застарілими, хоча і використовуються інколи у різних проектах, переважно для написання бібліотек.

Функціональні реакт-компоненти більш розповсюджені та легші для розуміння. Вони не використовують метод render, а просто повертають JSX.

Їх hooks, або хукі життєвого циклу – функції, що змінюють стан компоненту. Можуть бути, вбудованими в бібліотеці React, запозиченими із бібліотек, та кастомними – написаними користувачем.

Найпоширенішими є UseState та UseEffect. UseState – дозволяє змінювати стан компоненту, UseEffect – робити сторонні виклики.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

2.3. Підходи до менеджмента стану застосунку. Redux.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Стан застосунку – набір значень змінних у різних компонентах застосунка, у одного користувача у конкретний момент часу.

При користуванні програмою користувач може виконувати багато різних дій, результати яких мають бути візуально відображені, або збережені у застосунку. Це і переключення різноманітних режимів, колорових тем, мови, відкриття різноманітних сайдбарів, модальних вікон, повідомлень, також це можуть бути дії на кшталт додавання в корзину товару або збереження їх у список базових покупок.

Всі ці дії як правило відбуваються у різних частинах застосунку і в різних же частинах може відображатись результат. Таким чином, слідкування за усіма змінними є досить складним та потребує передачі даних між різними гілками дерева компонентів.

Систему менеджменту стану, такі як Redux, Mobx, Zustand полегшують дану задачу.

Redux – бібліотека, яку я використав у своєму застосунку, слугує «single source of truth» – єдиним джерелом істини. Вона зберігає дані про те, чи вікритий у нас боковий блок – сайдбар, чи має користувач дані про логін, яку роль має користувач, та багато іншого.

Я використовую Redux разом із бібліотекою Redux-toolkit, яка зменшує кількість додаткового коду для ініціалізації стану і роботи з ним.

Як же працює Redux?

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

З точки зору Redux, у застосунку є один незмінний State. Будь яка дія, яку виконує користувач, створює новий state, а не змінює існуючий стан.

Стан поділється на «шари», або slices, відповідно до функції у застосунку.

Щоб дістатися до значення у слайсі використоввуються селектори – на які можна підписатись та відстежувати кожен новий стейт у різних частинах застосунку, що ніяк не пов'язані між собою, окрім як через Redux state.

Кожен раз, для отримання нового state, користувач створює дію – action, який redux реєструє, оцінює, що в собі action несе (чи є об’єкт payload) і, відповідно до типу дії – повертає новий state (рис. 2.4.).

Зображення, що містить текст, знімок екрана, пляшка

Автоматично згенерований опис

Рис. 2.4. – Схема роботи бібліотеки Redux

2.4. Архітектура баз даних. Вибір бази даних для застосунку

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Щоб зрозуміти, яку, із великої кількості баз даних, ми будемо виористовувати у нашому проекті, нам необхідно знати їх різновиди.

Баз даних є велика кількість, та й навіть парадигм, за якими вони влаштовані, є близько десятку.

Найбільш розповсюдженими ж є SQL та NoSql бази даних.

Про них та про причини вибору саме тієї бази даних, що використана у засосунку – далі.

2.3.1 SQL, або Structured Query language

SQL – мова, розроблена для роботи з базами даних, що виникла на основі алгебри відношень, за участі Дональда Камбертіна та Реймонда Бойса у 1974 році.

У цій мові є досить великий набір команд для роботи з таблицями.

Додавання у таблицю, видалення, об’єднання таблиць, запит певної інформації із окремої таблиці чи із багатьох – це все можна робити за допомогою SQL-синтаксису.

Сама база даних підтримує відношення. Базуються вони на чітких інструкціях, які є частиною мови і для реалізації цих інструкцій нам потрібні схеми бази даних, або Schema.

У Schema є чітко детерміновані поля – fields.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Кожен новий запис у таблицю буде являти собою рядок, що чітко вписується у дану схему – має всі поля і значення, або містить значення null.

Неможливо, щоб запис мав додаткові поля. Якщо один запис має поле – всі інші також матимуть його.

Наступним «будівельним каменем» бази даних такого типу є те, що ми зазвичай працюємо не з одною, а з багатьма таблицями, які є пов'язаними. Тобто – мають відношення між собою.

Відношення бувають різними. Якщо у нас одна таблиця зв'язана з іншою таблицею, то такий зв'язок називають зв'язком «один до одного» (Рис. 2.5)

Зображення, що містить текст, знімок екрана, ряд, Шрифт

Автоматично згенерований опис

Рис. 2.5 – Відношення виду «один до одного»

Прикладом такого відношення може бути таблиця користувача, та таблиця його контактів. Одному користувачу належить лише одна таблиця контактів – і навпаки. Якщо нам треба зробити запит на контакти користувача – ми не будемо отримувати їх із таблиці самого користувача, ми їх об’єднаємо через відношення.

Наступним видом зв’язків є зв’язок типу «один до батьох».

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Такий зв’язок є характерним, коли ми маємо справу із однією таблицею, яка об’єднується із багатьма – наприклад, коли ми маємо одного клієнта, що робить багато замовлень (Рис. 2.6).

Зображення, що містить текст, знімок екрана, Шрифт, ряд

Автоматично згенерований опис

Рис. 2.6 – Відношення виду «один до багатьох»

Id у даному випадку є первинним ключом (primary key), сreator\_id у таблиці Product – foreign key.

Через таку зв’язку ключів і відбувається взаємодія.

Останнім видом зв’язків є «many-to-many» відношення.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Такий вид зв’язку використовується, коли обидві таблиці можуть належати багатьом іншим таблицям.

Наприклад, автор може написати багато книг, а одна книга, в свою чергу, може бути написаною декількома авторами.

Виникає необхідність в створенні третьої таблиці author\_book, яка буде поєднувати обидві попередні таблиці через ключі (рис. 2.7).

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Рис. 2.7 – Відношення виду «багато до багатьох»

Таким чином, через відношення між таблицями, що називаються joins, можна об’єднувати декілька таблиць, при наявності чіткої схеми та ключів, які повторюються у декількох таблицях (primary – foreign key),

Таким чином, у SQL-світі, ми матимемо нормалізовані, систематизовані таблиці, які завжди будуть мати поля, по яким ми зможемо зробити запит. І саме така модель взаємодії робить SQL ідеальним для багатьох проектів.

2.3.2 NoSql бази даних. MongoDB

Існує декілька баз даних цього типу, одним із прикладів була б DynamoDb, але найпопулярнішою NoSql базою даних зараз можна із впевненістю назвати – MongoDb.

Назва цієї бази походить від англійського слова «humongous», тобто – величезний. Її так назвали тому, що вона мала вміщювати багато даних і ефективно обробляти їх.

Вона була створена у 2009 році на мові C++.

Як же вона працює?

У такій базі даних немає схем і таблиць – лише так звані «колекції».

В кожній колекції є документи. Кожен документ, що лежить у колекції має подібний до JSON формат, але, при цьому – немає жодної схеми, що її необхідно дотримуватись!

Зображення, що містить текст, знімок екрана, Прямокутник, Шрифт

Автоматично згенерований опис

Рис. 2.8 – Документи у базі даних без схеми

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

При роботі із NoSql базою даних, немає жодного контракту, якого ми маємо дотримуватись. Ми можемо покласти у одну колекцію абсолютно різні документи.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Такий підхід може бути не ідеальним, якщо ми працюємо з існуючими даними, але якщо у нас є більш розширений набір даних, то ми можемо і не хотіти врешті решт притримуватись схеми, якщо ми точно знаємо, що шукана інформація у колекції є.

Також у такій базі даних немає відношень. Ми, звісно, можемо їх встановити вручну, через такі самі пари ключів у різних таблицях, проте NoSql бази даних набагато менше покладаються на них у своїй роботі.

Ідея роботи таких баз даних в іншому – ми згуртовуємо всі дані разом, у великі колекції і робимо декілька рівнів вкладеності, отримуючи таким чином у кожній колекції – великі за розміром документи, де є вся нам потрібна інформація.

Недоліком є те, що якщо у нас відбувається оновлення інформації у одній колекції, у нас також буде можливим оновлення у багатьох інших.

Такий підхід як раз підходить для застосунків у яких є багато операцій читання із бази даних, і не так багато операцій запису, оновлення та оновлення зв'язаних документів або таблиць.

Переглялянемо ще раз переваги та недоліки двох найрозповсюдженіших підходів до побудови баз даних.

2.5. Переваги різних типів баз даних

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Зразу можна сказати, що абсолютного переможця тут немає. Все залежить від тих завдань, які ставить перед собою розробник, та того проекту, який з базою даних пов'язаний. Часто також можуть використовуватись різні бази даних для різних потреб, якщо застосунок має мікросервісну архітектуру, та все ж, для чого краще підійде кожен тип баз даних?

Спочатку поговоримо про SQL.

Наявність у таких базах даних схеми може бути перевагою, а може і недоліком – в залежності від того, наскільки суворо типізованим є застосунок.

Якщо застосунок містить багато пов’язаних між собою даних, які часто змінюються, варто задуматись про використання SQL, так як зміни даних відбуваються в різних таблицях, що спрощує до них доступ та робить доступ більш швидким.

Також треба згадати про розширення (Scaling). Що це таке?

Розширення – це процес у якому змінюються потужності накопичуючих систем у бік збільшення.

Горизонтальне розширення – додавання нових серверів.

Вертикальне розширення – збільшення потужності існуючого серверу (накопичувача).

Якщо вертикальне розширення можливе для SQL, горизонтальне є нереальним, через неможливість розділити базу даних. Це може бути проблемою, якщо нам потрібно додати потужності.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Для SQL також ми маємо деякі обмеження в секунду на кількість запитів читання/запису.

У NoSql все навпаки – там немає схеми, немає (або дуже мало) відношень, можна з легкістью розділити базу даних на різні сервери, колекцій як правило менше, ніж таблиць і швидкодія набагато краща, ніж в SQL, але тільки коли йдеться про запити типу Read.

Зображення, що містить текст, знімок екрана, Шрифт, дизайн

Автоматично згенерований опис

Рис. 2.9 – SQL vs NoSql

Я вибрав для свого додатку саме NoSql базу даних через те, що у мене немає багато операцій запису, а є багато операцій читання та об’єднання даних, та і масив да

Висновки до розділу

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

Я дослідив методи розробки веб-додатків, стратегії рендерінгу, наявні фреймворки, та архітектуру баз даних і визначився з тими технологіями, що їх я буду використовувати у своєму застосунку.

Я обрав React з-поміж інших фреймворків через легкість розгортування, простий синтаксис, компонентну архітектуру, гарну підтримку та велику кількість бібліотек, що їх можна використати для демонстрації матеріалу у застосунку «EduStatsTracker».

Також я вирішив використати Node.js з фреймворком Express, через простий та зрозумілий підхід до створення бекенду, з використанням MVC паттерну розробки.

Також я взяв за основу базу даних Mongo.db, через те, що вона більше підходить для потреб застосунку «EduStatsTracker», а саме – зберігання великої кількості документів, що не завжди пов’язані між собою, та більшу потребу у читанні, ніж у записі нових даних.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ СИСТЕМИ

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

3.1. Архітектура програми

Наш застосунок «EduStatsTracker» був реалізований на фреймворку React із застосуванням React-router-dom для маршрутизації, Redux – для менеджменту state та Redux-toolkit для зменшення кількості boilerplate-коду.

Наш застосунок можна запустити, виконавши збірку за допомогою утіліти Vite, яка є заміною стандартного create-react-app.

На бекенді у нас використовується Node.js та Express.js, як і було вже зазначено. Ми під’єднуємося до бази даних за допомогою ORM Mongoose.

Бекенд у нас має монолітну архітектуру, і ми виконуємо вимоги шаблону MVC, тобто у нас є контроллер – виконавець запитів, модель – містить бізнес-логіку та View - яким можуть слугувати маршрутизатори, запит до яких повертає дані (Додаток Б).

Пакет concurrency збирає нам увесь додаток – і фронтенд і бекенд, для комфортної розробки.

Ми також використали npm пакет faker для створення mock-даних.

За показ графіків відповідає бібліотека nivo, що надає доступ до великої кількості чартів та графіків. Вона є «обгорткою» над існуючою і перевіреною бібліотекою D3.js.

Хешування паролей відбувається у bcrypt.js.

Наведемо приклад точки входу у застосунок та конфігурації middleware (рис 3.1)

Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис

Рис 3.1 – Конфігурація точки входу застосунок

3.2. Структура бази даних

Оскільки була вибрана нереляційна база даних, у нас немає чіткої схеми, та, все ж, ми маємо задати певну модель кожного із документів, що їх ми матимемо у даному застосунку.

Для цього у нас є Mongoose, який дозволяє нам не тільки визначити всі необхідні поля у базі даних, а й задати валідацію за типом для кожного з полів.

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

У нас будуть наявні такі п’ять моделей (далі можна додати і інші):

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

* Студент
* Професор
* Користувач
* Предмет
* Оцінка

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т

##### КПІ ім. Ігоря Сікорського

##### Група ЗПІ-зп42

36

1

Аркушів

Арк.

Літ.

Затв.

Перевірив

Н. контр.

Розробив

Підпис

№ докум.

### Лист

### Зм.

###### ЗПІ-зп42.190БАК.005.ПЗ

Т