

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ  
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ  
КАФЕДРА ПРИКЛАДНОЇ ІНФОРМАТИКИ

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_ Гамаюн В.П.  
(підпис) (ПБ)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021р.

**ДИПЛОМНИЙ ПРОЕКТ**  
**(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”**

**Тема:** Веб-агрегатор інформації про Вищі навчальні заклади України на базі платформи Node.js

**Виконавець:** \_\_\_\_\_ Олійник Вікторія  
Ігорівна \_\_\_\_\_ (підпис) (ПБ)

**Керівник:** \_\_\_\_\_ Зудов Олег Миколайович  
(підпис) (ПБ)

**Нормоконтролер:** \_\_\_\_\_ Боровик Володимир  
Миколайович \_\_\_\_\_ (підпис) (ПБ)

**Київ 2021**

## ВСТУП

Сьогодні Інтернет стає все більш розвиненим середовищем спілкування. У зв'язку з активним розвитком Інтернету, дедалі гостріше в програмуванні почала з'являтися окрема галузь – веб-програмування.

World Wide Web (WWW) - це глобальна комп'ютерна мережа, яка наразі містить мільйони сайтів, на яких розміщується будь-яка інформація. Люди отримують доступ до цієї інформації за допомогою Інтернет-технологій. Для пошуку в Інтернеті використовуйте спеціальні програми - веб-браузери, які значно полегшують подорож величезними просторами Інтернету.

Тепер, щоб привернути увагу клієнтів, покупців або партнерів, вам потрібно лише заявити про себе в Інтернеті, створивши веб-сайт. Для цих цілей він і служить. Веб-сайт може містити основну інформацію про організацію, особу, компанію, товари чи послуги, прайс-листи, контактну інформацію тощо. Ці сайти дозволяють зберігати, передавати та продавати різні типи інформації, не виходячи з екрану комп'ютера.

З кожним днем стає все більше сайтів на абсолютно різних темах, інформація на них копіюється, редагується, її стає все більше, в результаті чого виникає проблема, яку також можна вирішити створенням веб-сайту. Проблема полягає у відсутності структурованої інформації за певною темою, що вимагає від користувача витратити багато часу на збір та аналіз інформації. Ця проблема особливо серйозна для тих, хто подає документи до вищих навчальних закладів для вступу, адже спочатку треба дізнатися які університети є, знайти їх сайти та потрібну інформацію на них, структурувати її та проаналізувати. Це займає дуже багато часу, бо кожен університет по різному відображає інформацію на своєму сайті. Для вирішення цієї проблеми можна створити агрегатор інформації у вигляді веб-сайту.

## **РОЗДІЛ 1.**

### **АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ ДЛЯ СТВОРЕННЯ ВЕБ-АГРЕГАТОРУ ІНФОРМАЦІЇ**

#### **1.1. Актуальність дослідження**

Створення веб-сайту це надзвичайно складний процес. Донедавна розробка веб-сайтів була доступна лише для великих організацій та компаній. Однак сьогодні це популярний сервіс, доступний кожному в Україні та за її кордонами. Попит на послуги з розробки веб-сайтів зростає і в освітньому секторі.

Щороку кожен абітурієнт стикається з проблемою вибору університету, де він отримуватиме вищу освіту. Веб-сайти більшості університетів мають різну структуру: основна інформація про університет, його структурні підрозділи, новини, контакти та адреси, вимоги до вступу знаходяться в різних місцях, і часто їх дуже важко знайти через велику інформації на них.

Через відсутність в Інтернеті структурованої інформації про всі університети у формі, придатній для недосвідчених користувачів, більшість людей обирають куди вступати, спираючись на рекомендації інших, які об'єктивно не охоплюють усіх характеристик освіти в цих університетах.

Розробка веб-агрегатора інформації про вищі навчальні заклади України має надати користувачам єдине джерело чітко структурованої інформації про українські університети, щоб вони могли вибрати місце навчання з набагато меншими витратами часу та вищою ефективністю.

#### **1.2. Постановка завдання**

Для розробки веб-агрегатора інформації про Вищі навчальні заклади України на базі платформи Node.js необхідно:

- проаналізувати схожі програмні засоби, дослідити методи отримання інформації із сторонніх веб-сайтів, методи розробки сервісної та клієнтської частини, порівняти їх позитивні і негативні якості;
- спроектувати інтерфейс програми, структуру бази даних, структуру файлів програми та визначити вимоги;
- розробити діючий прототип веб-сервісу агрегації даних із веб-сайтів Вищих навчальних закладів України та видачі цієї інформації у форматі HTML-сторінок.

### **1.3. Відомі підходи до реалізації завдань кваліфікаційної роботи**

Для того, щоб створити веб-ресурс, проводиться детальний аналіз, що визначає критерії, яким він повинен відповідати. Процес створення складається з шести основних фаз:

- а) визначення цілей та постановка завдань;
- б) розробка структури проекту;
- в) розробка інтерфейсу;
- г) HTML-верстка;
- д) програмування та контроль якості;
- е) введення в експлуатацію та оптимізація.

#### **1.3.1. Шаблони структури програми**

Під час розробки програми, як правило, використовується кілька шаблонів, які описують структури програми та порядок виконання її функцій. Найпопулярніші серед них:

- "виконуваний (executable) застосунок" – тип програм, розроблених для конкретного пристрою або системи; такі програми, як правило, характеризуються високою швидкістю і сумісністю;

- "клієнт-сервер" – архітектура клієнт-сервер дає можливість відокремити більшість бізнес-логіки від пристрою користувача, через що зростає швидкість розробки, надійність, але в свою чергу також підвищується вартість підтримки програм; саме цей шаблон буде доцільно використовувати для реалізації веб-агрегатора інформації;

- "моноліт" – архітектурний шаблон, що описує програму як сукупність тісно пов'язаних модулів, які працюють нероздільно (наприклад, класи в Java-програмі);

- "мікросервіси" – архітектурний стиль, в якому одна програма будується як сукупність невеликих сервісів, кожен з яких працює у своєму власному процесі та взаємодіє з іншими за допомогою простих і швидких протоколів даних, зазвичай HTTP.

### 1.3.2. Мови програмування для реалізації програми

Сервісна частина повинна бути написана мовою, яка відповідає наступним вимогам:

- а) забезпечення підтримки обміну повідомленнями за протоколом HTTP / HTTPS;
- б) швидка обробка запитів від великої кількості користувачів;
- в) простота установки;
- г) забезпечення високого рівня відмовостійкості.

На сьогоднішній день мовами, що задовольняють вказаним вимогам є:

- Java;
- PHP;
- C#;
- Python;
- JavaScript (Node.js).

**Java** - це об'єктно-орієнтована мова програмування.

В офіційній реалізації Java-програми компілюються в байт-код, який інтерпретується віртуальною машиною для конкретної платформи. "Oracle" надає програмісту компілятор Java та віртуальну машину Java, які відповідають специфікації Java Community Process за ліценцією GNU General Public License. У більшій мірі дана мова запозичила синтаксис із C та C ++. З основу взято об'єктну модель C ++, але її було дещо змінено. Можливість появи деяких конфліктних ситуацій, які виникали внаслідок помилок програмістів, була усунуто, а також полегшено саме процес розробки об'єктно-орієнтованих програм. Велика кількість дій, які програмісти повинні прописувати на C / C ++ вручну, було доручено віртуальній машині Java. Головна перевага Java заключається в тому, що вона розроблялась як платформно-незалежна мова, тому вона має менше функцій низького рівня для роботи з апаратним забезпеченням, що дещо уповільнює швидкість виконання програми. За такої необхідності Java дозволяє виконання підпрограм, написаних іншими мовами програмування.

**PHP**, спочатку названий *Personal Home Page Tools* - мова програмування сценаріїв, призначена для створення HTML-сторінок на стороні веб-сервера. PHP - одна з найбільш поширених мов у сфері веб-розробки.

Мова PHP дає можливість в її код вбудувати html-код сторінок, що будуть коректно оброблятися PHP-інтерпретатором. Обробник PHP починає виконувати код після відкриваючого (`<? php`) і продовжує виконання доти, доки не зустріне закриваючий тег.

Широкий діапазон функцій PHP дозволяє уникати написання багаторядкових функцій, призначених для користувача, як це є, наприклад, в мовах C та Pascal.

Ефективність є дуже важливим показником для реалізації програм, що розраховуються на велику кількість користувачів, до чого відноситься і веб-розробка. Вагомою перевагою PHP є те, що мова є інтерпретованою, що

дозволяє обробляти сценарії з високою швидкістю. Більшість PHP-скриптів (принаймні, не дуже великих) виконуються швидше за подібні програми, написані на Perl. Але в будь-якому разі виконавчі файли, отримані за допомогою компіляції, працюватимуть в десятки, а іноді і в сотні разів швидше. Однак не зважаючи на це, мови PHP достатньо для створення повноцінних веб-застосунків.

Недоліки PHP:

- а) недоліки дизайну мови:
  - 1) складні назви поширених функцій;
  - 2) змінні із символом \$;
  - 3) Unicode не підтримується у версіях до 6.0;
- б) непередбачуваність нових версій PHP.

**C#** - це об'єктно-орієнтована мова програмування із захищеною системою типізації для платформи .NET.

Синтаксис C # дуже схожий на синтаксис мов C ++ та Java. Мова має строго статичний типізацію, підтримує поліморфізм, перетин операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки та коментарі у форматі XML. C# перейняла багато від своїх попередників — мов програмування Object Pascal, C++ і Smalltalk – та спираючись на практику їх використання, виключила деякі моделі, що викликали велику кількість проблем при розробці програмних систем, наприклад, множинне успадкування класів, що передбачене в мові C++, але не в C#.

**Python** - інтерпретована об'єктно-орієнтована мова програмування зі строгою динамічною типізацією. Головними перевагами цієї мови є структури даних високого рівня, динамічна семантика та динамічне зв'язування, що робить її оптимальною для швидкої розробки програм, а також для поєднування наявних компонентів. Python підтримує модулі та пакети модулів, що забезпечує модульність та повторне використання коду.



Інтерпретатор мови Python та стандартні бібліотеки доступні на всіх основних платформах у скомпільованій формі та у вихідній формі. Мова програмування Python підтримує кілька парадигм програмування, а саме: об'єктно-орієнтована, аспектно-орієнтована, процедурна та функціональна.

**JavaScript** (JS) - це динамічна, об'єктно-орієнтована [4] прототипна мова програмування. Найчастіше вона використовується для створення сценаріїв веб-сторінок, що забезпечує взаємодію з користувачем, керування браузером, асинхронний обмін даними з сервером, змінення структури та зовнішнього вигляду веб-сторінки на боці клієнта.

JavaScript використовується для:

- написання сценаріїв веб-сторінок, щоб зробити їх інтерактивними;
- створення односторінкових веб-додатків (AngularJS, React, Vue.js);
- програмування на стороні сервера (Node.js (Express.js));
- написання стаціонарних програм (Electron, NW.js);
- створення мобільних додатків (React Native, Cordova);
- написання скриптів в прикладних програмах (наприклад, у програмах з Adobe Creative Suite або Apache JMeter);
- у документах PDF.

**Node.js** - це платформа з відкритим кодом для обробки високопродуктивних веб-застосунків, написаних на JavaScript. Засновником платформи є Райан Даль. Спочатку Javascript використовувався для обробки даних у браузері користувача, але node.js надав можливість запускати сценарії JavaScript на сервері та надсилати користувачеві результат їх виконання. Платформа Node.js зробила JavaScript спільною мовою для великої спільноти розробників.

Node.js має такі властивості:

- а) однопоточкова асинхронна модель виконання запитів;
- б) неблокуючий ввід / вивід;
- в) система модулів CommonJS;
- г) механізм JavaScript Google V8.

Менеджер пакетів вузлів npm (node package manager – менеджер пакетів вузлів) використовується для управління модулями.

Платформа Node.js призначена для запуску потужних мережеских застосунків, написаних мовою JavaScript. Платформа працює не тільки з серверними скриптами для веб-запитів, але використовується й для створення клієнтських та серверних програм.

### 1.3.3. Парсинг сайтів

Отримання інформації зі сторонніх веб-сайтів здійснюється шляхом парсингу. Це синтаксичний аналіз веб-сайтів, який виконують спеціальні програми чи скрипти. Отримана інформація надається у конкретній формі відповідно до конкретних правил і виконується однією з мов програмування. Об'єктом такого аналізу може бути довідник, каталог, Інтернет-магазин, блог та інші Інтернет-ресурси.

Парсинг веб-сайтів – найкращий спосіб автоматизувати збір та зберігання інформації у потрібному вигляді. Завдяки парсеру (програмі, що виконує синтаксичний аналіз) можна створювати та оновлювати веб-сайти, якщо вони хоч трохи схожі за оформленням, змістом та структурою.

Незалежно від мови програмування, на якій написаний парсер, алгоритм його роботи залишається незмінним:

- а) підключення до мережі Інтернет, отримання доступу та завантаження коду веб-ресурсу;
- б) зчитування, отримання та обробка даних;

в) надання витягнутих даних у зручній формі - .txt, .sql, .xml, .html та інших форматах.

Реалізація парсеру може бути виконана різними способами:

- самостійне написання синтаксичного аналізатора або скрипта, що витягує дані за прописаними вами умовами на обраній вами мові програмування (C, C ++, Python, PHP Java, тощо);
- використання Google Docs та ImportXML (у Google Docs, точніше в Google Spreadsheets, можна витягти багато даних за допомогою різноманітних функцій);
- використання хмарних ресурсів, таких як Import.io, Competera, Ostorparse тощо. Вони пропонують можливість провести аналіз за допомогою хмарних технологій.

Парсинг здійснюється за допомогою емуляції віртуального DOM. Доступ до даних здійснюється через віртуальний DOM. До того ж їх можна читати та змінювати, що корисно при перетворенні дат та видаленні зайвих пробілів.

Потім отримана інформація перетворюється в уніфікований вигляд, в якому вона записується до бази даних.

Схема виконання парсингу веб-сайту зображено на рис. 1.1.

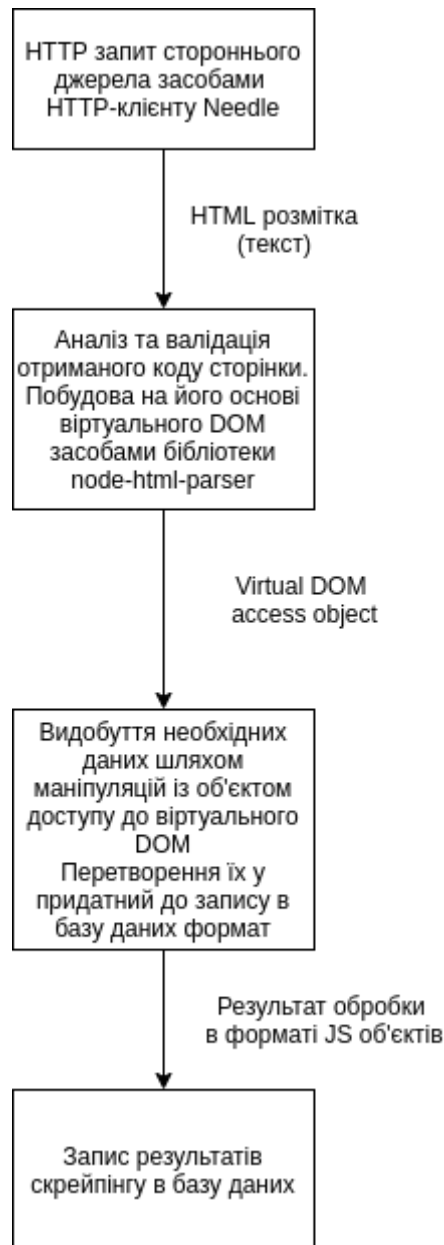


Рис. 1.1. Послідовність виконання парсингу сайту

#### 1.4. Порівняльний аналіз схожих додатків

Osvita.ua це веб-сайт, який дозволяє переглянути набір вузів та списки спеціальностей за вказаним напрямком та регіоном (рис. 1.2.). Має зручний, але морально застарілий інтерфейс, що до того ж погано оптимізований під мобільні пристрої.

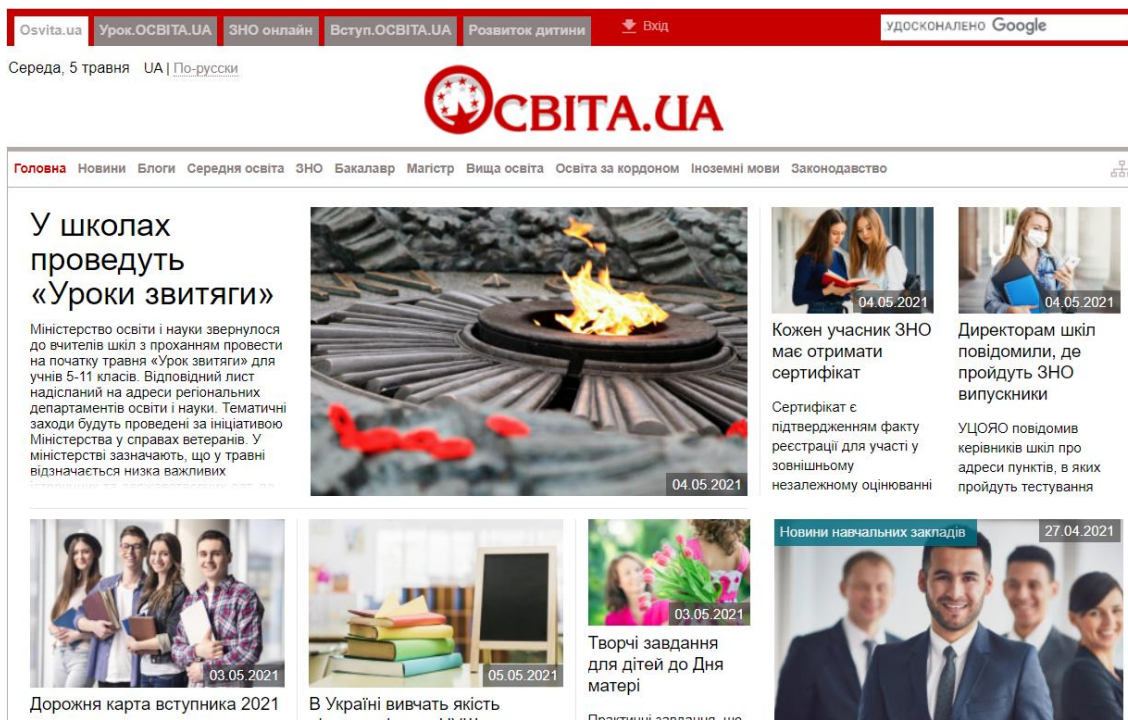


Рис. 1.2. Головна сторінка сайту Osvita.ua

Education.ua веб-сайт, на якому можна здійснити пошук вузів, коледжів, курсів, шкіл та репетиторів. Відрізняється приємним дизайном та зручним інтерфейсом (рис. 1.3) Має зручну навігацію по сторінці і всю необхідну інформацію на ній та можливість залишити відгук чи коментар. Але пошук працює оптимально тільки за заданням міста, при пошуку за спеціальністю користувач отримує список університетів з такими спеціальностями, але далі виконує пошук так само.

Веб-сайт parta.com.ua (рис. 1.4.) надає велику кількість інформації за різними темами: Вузи України, ЗНО, репетитори, навчання за кордоном тощо, але ця інформація часто неактуальна, бо оновлюється людьми, не за допомогою парсингу сайтів.

## Всі навчальні заклади України

ЗВО (вузи) Коледжі Курси Репетитори За кордоном Школи Дитячі садки

Всі міста  Всі спеціальності  [Знайти](#)

Рис. 1.3. Головна сторінка сайту Education.ua

Освіта в Україні: довідник абітурієнта, курсове навчання, каталог репетиторів, реферати і багато іншої інформації для школярів і студентів

[Увійти](#) | [Зареєструватися](#)



[додати у вибране](#) [зробити стартовою](#)

[Вузи України](#) [Курси](#) [Дитячі табори](#) [Онлайн-навчання](#) [Майстер-класи](#) [ЗНО](#) [Робота студентам](#) [Репетитори](#) [Автошколи](#) [Навчання за кордоном](#)

Ласкаво просимо на всеукраїнський освітній портал №11

ТЕХНОЛОГІЇ Gc



**СУЧАСНІ ЗАКЛАДИ ОСВІТИ**  
Дванадцята міжнародна виставка  
**26-28 ТРАВНЯ 2021 року**  
КИЇВСЬКИЙ ПАЛАЦ СПОРТУ  
Вхід вільний

### Освіта в Україні на Parta.ua

#### [Вузи України](#)

Всі вузи в Києві і Україні, університети, інститути і академії країни з розбиттям по регіонах та напрямках навчання.

#### [Курси іноземних мов](#)

Англійські, німецькі та ін. курси в Україні, вивчення більш ніж 30 мов, курси ділової мови, англійська для роботи і бізнесу.

#### [Позашкільна освіта](#)

Секції, гуртки та тематичні клуби для школярів та підлітків, курси для дітей шкільного віку різних напрямків

#### [Освіта за кордоном](#)

Навчання у Великобританії, Франції, США, Канаді, англійська в Англії і на Мальті та інші пропозиції по навчання за кордоном.

#### [Курси професій](#)

Інформація про курси професій – курси дизайну, програмування, бухгалтерські курси 1С, курси оператора ПК і багато інших.

#### [Технікуми України](#)

Будівельні, транспортні технікуми в Києві і Україні, умови, ціни навчання та інша інформація для вступників

Рис. 1.4. Головна сторінка сайту parta.com.ua

## **Висновки до розділу**

Дослідження предметної області показало необхідність розробки Веб-агрегатора інформації про Вищі навчальні заклади України для полегшення пошуку інформації абітурієнтами вищих навчальних закладів України.

На основі порівняльного аналізу схожих додатків дозволяє визначити вимоги, яким повинен відповідати сайт, та які функції він має виконувати.

На основі дослідження відомих підходів до реалізації програмних додатків було визначено стек технологій, який використовуватиметься під час вирішення завдання.

## РОЗДІЛ 2.

### ПРОЕКТУВАННЯ ВЕБ-АГРЕГАТОРУ ІНФОРМАЦІЇ ПРО ВИЩІ НАВЧАЛЬНІ ЗАКЛАДИ УКРАЇНИ

#### 2.1. Специфікація вимог до веб-агрегатора інформації

а) *Переглянути інформацію про ВНЗ на карті. Використання карти (зменшення масштабу, збільшення масштабу, перетаскування карти в боки) для пошуку вищих навчальних закладів на певній території. Стан вимоги – відібрано. Пріоритет – важливий. Рівень ризику – звичайний.*

*Вимоги до якості:*

1) Функціональна придатність (функціональна правильність, точність; функціональна повнота, завершеність) – програмне забезпечення повинно мати всі функції для оновлення даних на сайті відповідно до задач.

2) Зручність використання (розпізнавання відповідності потребам; вивчаємість; естетика інтерфейсу користувача) – інтерфейс повинен бути зрозумілим та приємним для користувача.

б) *Переглянути список новин ВНЗ, вказаних на сайті. Отримання списку новин з сайтів вищих навчальних закладів, що додані на даний сайт за допомогою парсингу. Стан вимоги – відібрано. Пріоритет – важливий. Рівень ризику – звичайний.*

*Вимоги до якості:*

1) Ефективність виконання (поведінка у часі, реактивність, швидкодія) – новини повинні завантажуватися швидко

2) Функціональна придатність (функціональна правильність, точність; функціональна повнота, завершеність) – програмне забезпечення повинно мати всі функції для оновлення даних на сайті відповідно до задач.

в) *Обрати ВНЗ для отримання більше інформації. Отримання списку вищих навчальних закладів, що додані на даний сайт, у вигляді*



*посилань на їхні сайти. Стан вимоги – відібрано. Пріоритет – критичний. Рівень ризику – значимий.*

*Вимоги до якості:*

1) Ефективність виконання (поведінка у часі, реактивність, швидкодія) – список повинен завантажуватися швидко.

2) Функціональна придатність (функціональна правильність, точність; функціональна повнота, завершеність) – програмне забезпечення повинно мати всі функції для оновлення даних на сайті відповідно до задач.

г) *Залишити запитання. Ввести в форму свої дані (ім'я, телефон та електронну пошту) і питання та надіслати інформацію.* Стан вимоги – відібрано. Пріоритет – важливий. Рівень ризику – звичайний.

*Вимоги до якості:*

1) Ефективність виконання (поведінка у часі, реактивність, швидкодія) – форма повинна відправлятися швидко.

2) Зручність використання (розпізнавання відповідності потребам; вивчаємість; естетика інтерфейсу користувача) – інтерфейс повинен бути зрозумілим та приємним для користувача.

д) *Оновити дані про ВНЗ. Оновлення інформації про доступні спеціальності вищих начальних закладів, доданих до сайту.* Стан вимоги – відібрано. Пріоритет – важливий. Рівень ризику – звичайний.

*Вимоги до якості:*

1) Функціональна придатність (функціональна правильність, точність; функціональна повнота, завершеність) – програмне забезпечення повинно мати всі функції для оновлення даних на сайті відповідно до задач.

2) Ефективність виконання (поведінка у часі, реактивність, швидкодія) – оновлення повинно виконуватися швидко.

є) *Оновити дані про новини. Додати у список новини, що були опубліковані пізніше останнього оновлення.* Стан вимоги – відібрано. Пріоритет – важливий. Рівень ризику – звичайний.

*Вимоги до якості:*

1) Функціональна придатність (функціональна правильність, точність; функціональна повнота, завершеність) – програмне забезпечення повинно мати всі функції для оновлення даних на сайті відповідно до задач.

2) Ефективність виконання (поведінка у часі, реактивність, швидкодія) – оновлення повинно виконуватися швидко.

ж) *Переглянути запитання. Вивести на екран список запитань, що залишилися користувачами.* Стан вимоги – відібрано. Пріоритет – важливий. Рівень ризику – звичайний.

*Вимоги до якості:*

1) Зручність використання (розпізнавання відповідності потребам; вивчаємість; естетика інтерфейсу користувача) – інтерфейс повинен бути зрозумілим товарознавцю, щоб швидко оперувати ним

2) Ефективність виконання (поведінка у часі, реактивність, швидкодія) – клієнт повинен отримувати результат швидко.

При виконанні даних вимог, веб-агрегатор інформації матиме необхідний функціонал. Зазначимо, що це основні вимоги до сайту для забезпечення його мінімально достатніх функцій.

Вище описані вимоги також зображені на діаграмі, що описує послуги, які система надає користувачу (рис. 2.1.).

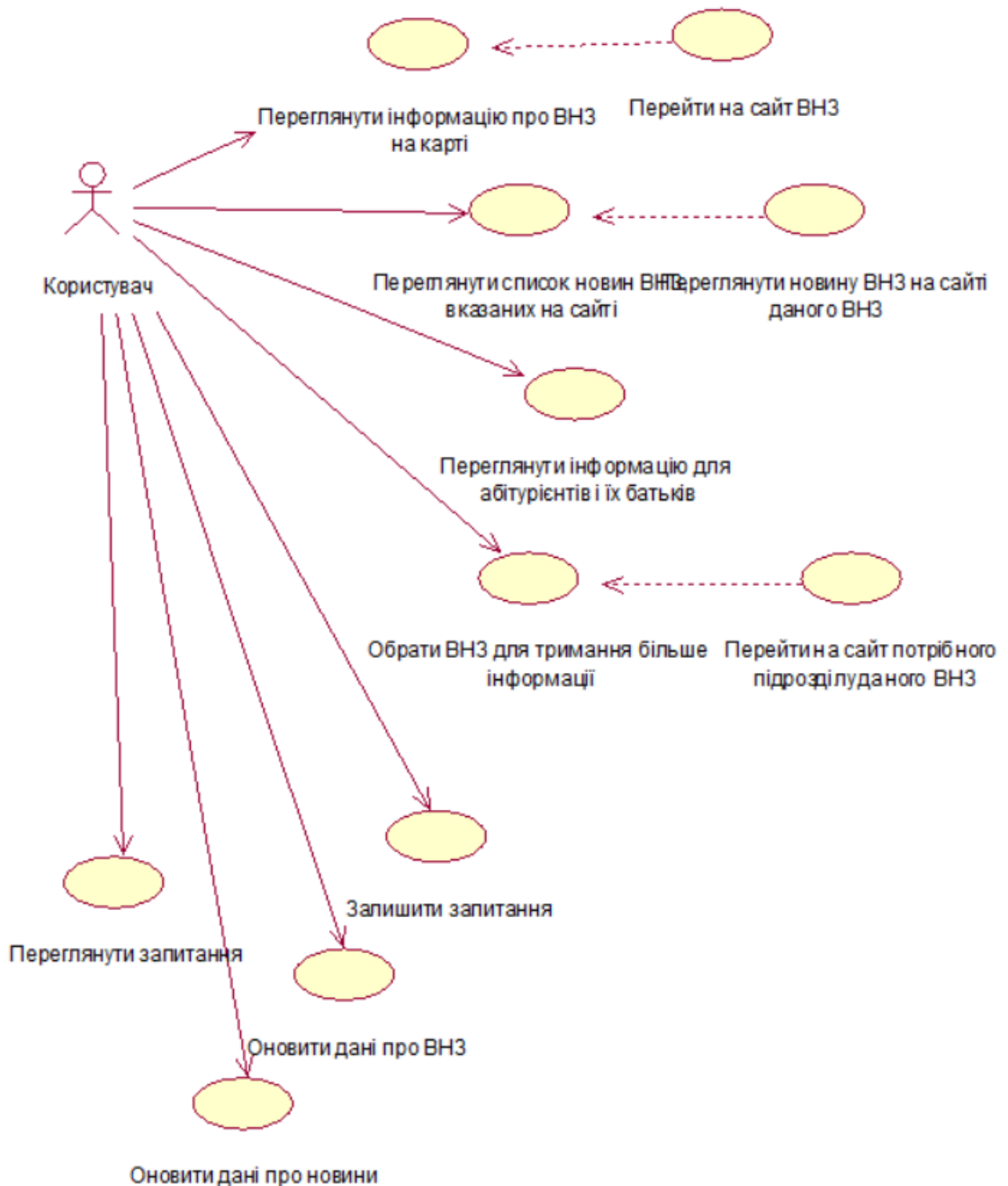


Рис. 2.1. Діаграма варіантів використання

## 2.2. Розробка інтерфейсу

Кожна сторінка сайту повинна містити незмінну плажку меню сайту, яка забезпечить йому просту навігацію.

Титульна сторінка («Головна») повинна відображати інформацію про призначення даного сайту, а також карту з відміченими на ній Вищими навчальними закладами.

Сторінка «Новини» відображатиме список новин, розміщений на сайтах ВНЗ, у порядку їх публікування за часом. Натиснувши на новину, повинна відкритися сторінка з публікацією цієї новини на сайті відповідного Вищого навчального закладу.

Сторінка «Абітурієнтам та батькам» повинна містити інформацію про умови вступу до Вищих навчальних закладів України.

Сторінка «ВНЗ» повинна відображати список ВНЗ, інформація про які, присутня на даному сайті, у вигляді посилань на інші сторінки. При натисненні на одне з посилань, повинна відкритися сторінка з загальною інформацією про відповідний Вищий навчальний заклад, його підрозділами, новинами та картою з розташуванням на ній цього ВНЗ.

Сторінка «Задати запитання» повинна містити форму з полями «Ім'я», «Контактний номер», «Електронна скринька», «Запитання» та кнопкою відправлення інформації «Ок».

### **2.3. Розробка структури проекту**

Проект буде складатися з папок, що вказані на рис. 2.2.

Загалом проект складається з таких пакетів:

- а) `node_modules` — залежності програми;
- б) `client` — клієнтська частина програми;
- в) `server` — серверна частина програми;
- г) `src` — код програми.

У пакеті `client` знаходяться наступні папки:

- `resources` — папка з шрифтами, зображеннями та іншими файлами, що не відносяться до коду;
- `api` — папка з файлами в яких описано методи доступу до серверу;

- app — папка в якій знаходяться головні файли програми;
- components — файл із компонентами інтерфейсу (React);
- const — папка із файлами в яких описані константи (шляхи віртуального роутингу, шляхи доступу до арі);
- utils — файли із допоміжними функціями та методами;
- stores — файли, в яких описано логіку обробки даних засобами бібліотеки MobX.

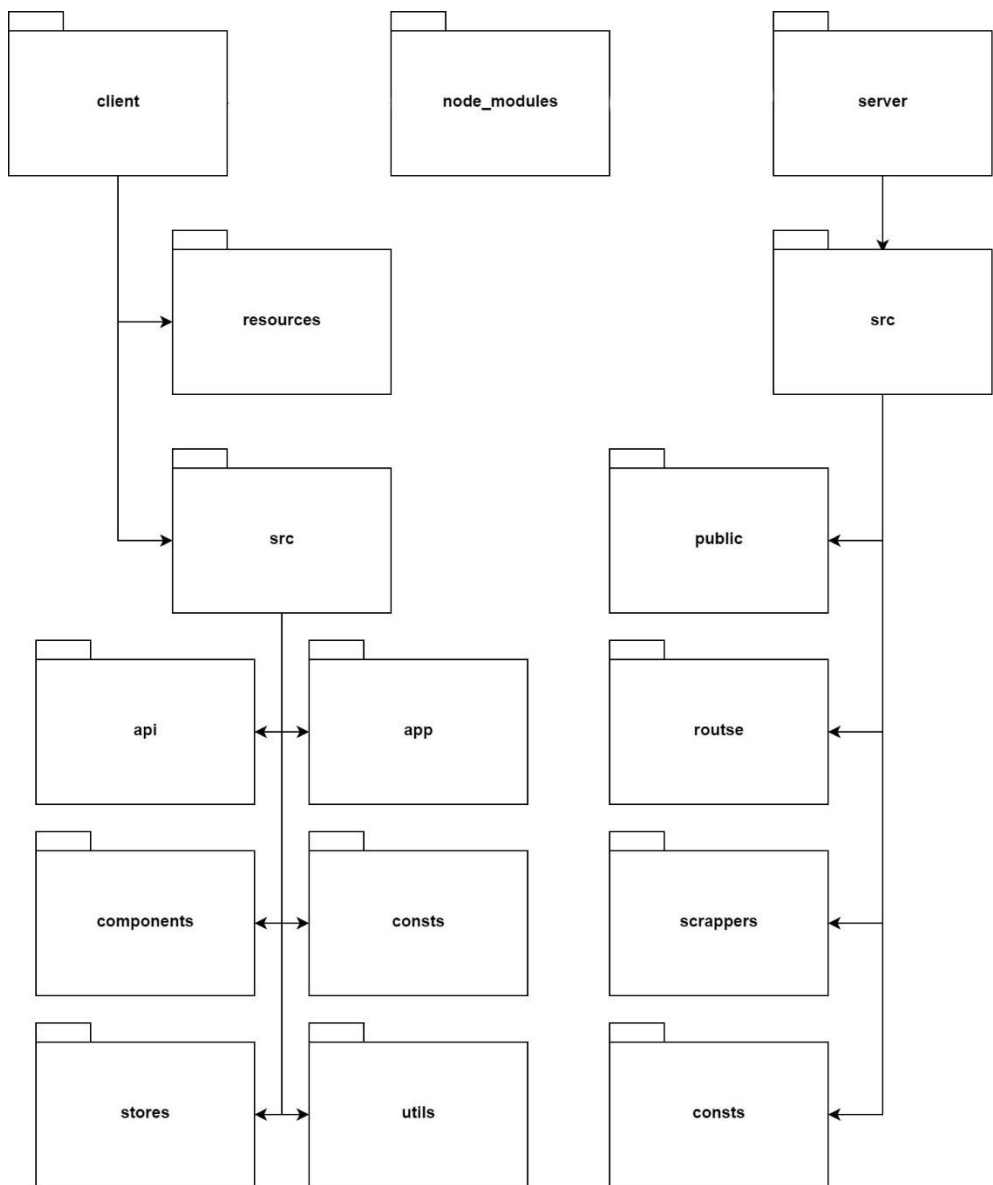


Рис. 2.2. Структура проекту

У пакеті `server` знаходяться наступні папки:

- `public` — компільовані файли клієнської частини (Bundle);
- `routes` — файли в яких описано методи обробки http запитів засобами фреймворку `Express.js`;
- `scrapers` — файли, в яких описано методи отримання даних із сторонніх сайтів методом веб-скрейпінгу.

## 2.4. Структура бази даних

На рис. 2.3 зображено структуру бази даних, яка використовується сервісом для довгострокового зберігання новин, запитань та інформації про вузи.

В програмі використовується нереляційна документоорієнтована база даних `MongoDb`. Дані зберігаються в популярному на сьогодні форматі `JSON` (JavaScript Object Notation).

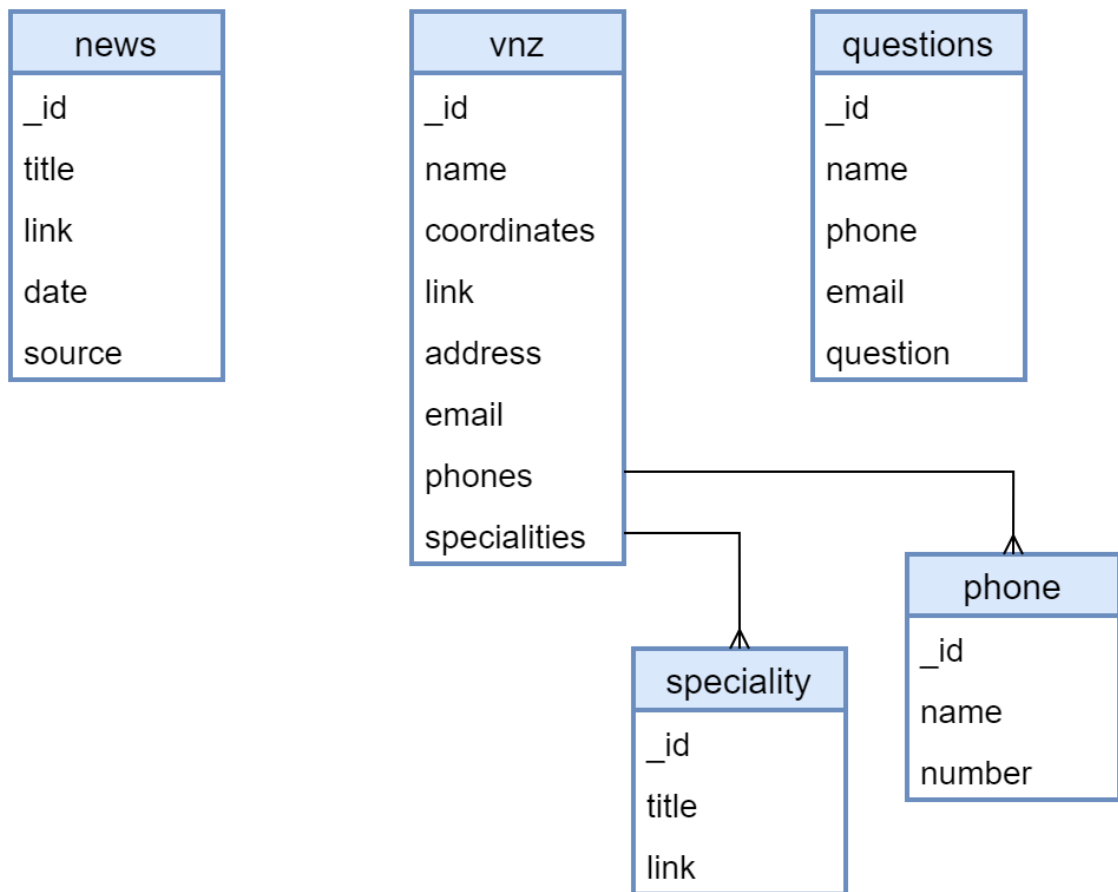


Рис. 2.3. Структура бази даних

## 2.5. Розгортання сервісу

Веб-агрегатор інформації про Вищі навчальні заклади України на базі платформи Node.js потребує таких мінімальних вимог системи:

- POSIX-сумісна операційна система (зазвичай це Unix-подібні системи та їх клони);
- встановлена платформа Node.js та пакетний менеджер npm;
- процесор від 1ГГц;
- оперативна пам'ять від 1 Гб;
- пам'ять жорсткого диску від 6,5 Гб;
- мережевий адаптер, що підтримує швидкість від 100 Мбіт.

Розгортання сервісу заключається у виконанні наступних дій:

- a) встановлення СКБД MongoDB;

- б) встановлення платформи Node.js та пакетного менеджера npm;
- в) завантаження на сервер вихідного коду програми;
- г) встановлення залежностей;
- д) бандлінг клієнту;
- е) запуск серверу.

Після встановлення сервіс готовий до використання. Організацію розміщення модулів сервісу зображено на схемі (рис. 2.4).

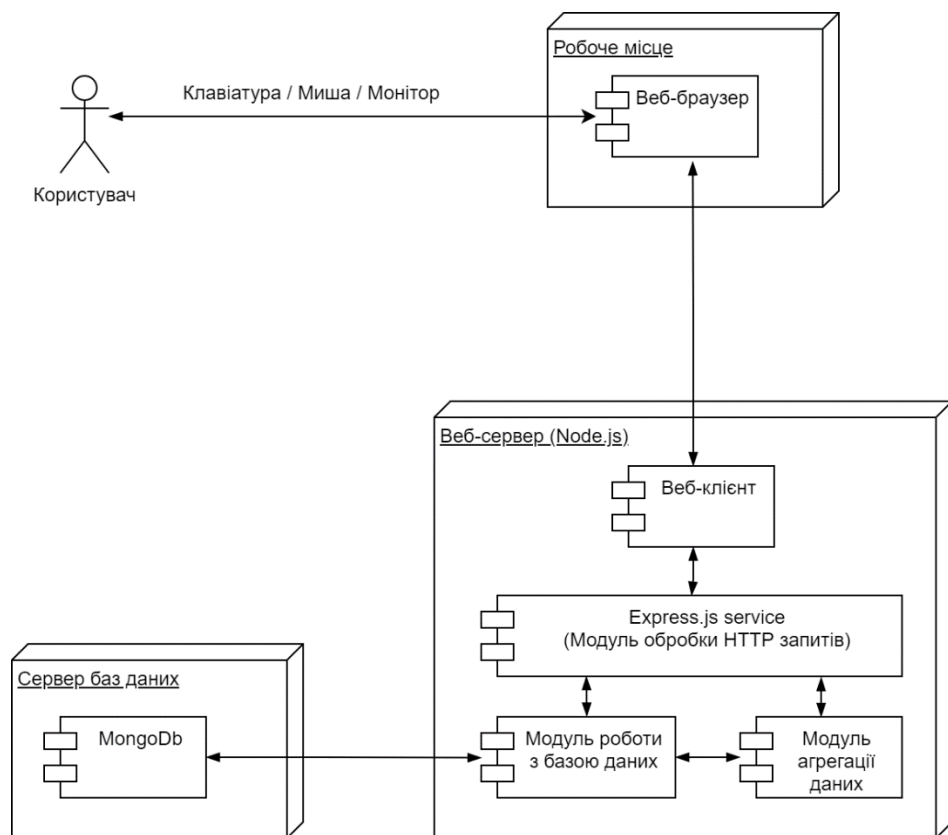


Рис. 2.4. Діаграма розгортання



## **Висновки до розділу**

На основі досліджень приведених в попередньому розділі було визначено вимоги до веб-агрегатора інформації про вищі навчальні заклади України та його функції.

На основі визначених вимог було описано структуру рішення, а саме: структура сайту, структура бази даних, діаграма пакетів та діаграма розгортання сервісу.

## РОЗДІЛ 3.

### РОЗРОБКА ВЕБ-АГРЕГАТОРУ ІНФОРМАЦІЇ ПРО ВИЩІ НАВЧАЛЬНІ ЗАКЛАДИ УКРАЇНИ

#### 3.1. Застосування архітектури клієнт-сервер для побудови веб-агрегатору інформації

Клієнт-серверна архітектура - це загальна структура додатків, яка розподіляє завдання або робочі навантаження між постачальниками ресурсів або послуг, які називаються серверами, та ініціаторами гостьових служб, які називаються клієнтами. [1] Клієнти та сервери часто спілкуються через комп'ютерну мережу на окремих пристроях, але клієнти та сервер можуть жити і в одній системі. Одна або декілька серверних програм запускаються на хостовому сервері та розподіляють його ресурси між клієнтами. Клієнт зазвичай не ділиться своїми ресурсами, він може виконувати програму за допомогою ресурсів сервера або своїх. Таким чином, клієнти ініціюють сеанси спілкування із серверами та чекають запрошень від них. Приклади комп'ютерних програм клієнт-сервер включають електронну пошту, мережевий друк та Всесвітню павутину.

Ранньою формою архітектури клієнт-сервер є введення віддаленого завдання в OS / 360 (1964), де вимогою було запустити завдання та отримати результат.

Під час формулювання моделей клієнт-сервер в 1960-70-х роках комп'ютерні мережі, створені ARPANET (в Стенфордському науково-дослідному інституті), використовували терміни сервер-хост (або підтримуючий хост) і хост-член.

Сучасна клієнт-серверна взаємодія в основному визначається розподілом відповідальності між клієнтом та сервером. Логічно виділити три рівні роботи:

- а) рівень подання даних;
- б) прикладний рівень (рівень програми);
- в) рівень управління даними.

Презентація даних, по суті, є користувацьким інтерфейсом і відповідає за представлення даних користувачеві та введення його команд для управління.

Рівень програми реалізує основну логіку програми та виконує потрібну обробку даних.

Рівень управління даними забезпечує зберігання даних та доступ до них.

Двоступенева клієнт-серверна архітектура забезпечує взаємодію двох програмних модулів - клієнта та сервера. В залежності від розподілу між ними вищезазначених функцій, існують є такі їх моделі:

- модель тонкого клієнта (у ній уся логіка управління програмами та даними зосереджена на сервері; клієнтська програма надає функції лише на рівні презентації);
- товста клієнтська модель (у ній сервер керує тільки даними, а їх обробка та користувацький інтерфейс орієнтовані на сторону клієнта. Товстих клієнтів також часто називають пристроями з обмеженою потужністю: КПК, стільникові телефони тощо).

Трирівнева архітектура клієнт-сервер, що почала свій розвиток в середині 1990-х років, передбачає відокремлення рівня додатків від управління даними. Виділяється окремий програмний рівень, який базується на логіці програми. Програми середнього рівня можуть працювати на виділених серверах застосунків, але такі програми можуть працювати і на звичайному веб-сервері. Нарешті, даними керує сервер даних.

Для роботи з даною системою користувач використовує стандартне програмне забезпечення - звичайний браузер. Це позбавляє від необхідності

завантажувати та встановлювати спеціальні додаткові програми (хоча іноді все одно це потрібно робити). Однак користувач повинен бути забезпечений інтерфейсом, за допомогою якого він може взаємодіяти з системою та генерувати запити на неї. Форми, що створюють цей інтерфейс, публікуються на веб-сайтах і завантажуються разом з ними.

Веб-браузер генерує запит і надсилає його на сервер для обробки. При необхідності сервер викликає серверні програмні модулі, які виконують обробку запитів і, при необхідності, надають доступ до сервера даних. Сервер даних виконує операції з даними, що зберігаються в системі, і створює свою інформаційну базу. Він може сканувати відповідно до запиту з бази даних і передавати його в модуль проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, зазвичай організовані як реляційна база даних.

У багатьох випадках веб-сервер і серверні модулі середнього рівня розміщуються на одному комп'ютері, хоча це окремі та логічно незалежні програмні модулі (рис. 3.1).

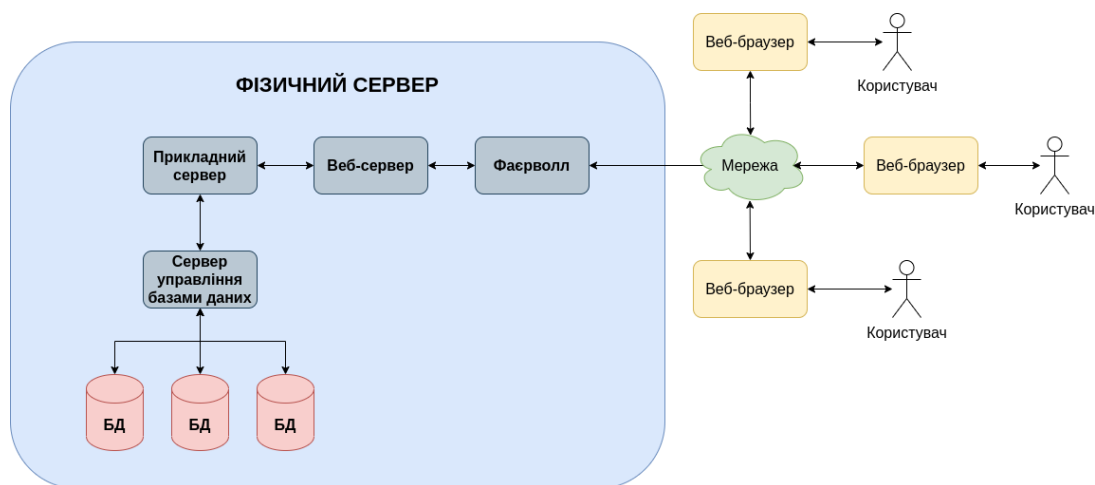


Рис 3.1 Застосування моделі клієнт-сервер на прикладі сучасних веб-застосунків

### 3.2. Практична реалізація програмного додатку з використанням архітектурного патерну клієнт-сервер.

На основі аналізу інструментів збору даних з веб-сайтів, та найбільш популярних мов програмування в попередніх розділах було визначено пакет

технологій для розробки веб-агрегатора інформації про Вищі навчальні заклади України. Цей пакет складається з:

- а) Node.js (як основа серверного компоненту) [2];
- б) Express.js (фреймворк для обробки http-запитів на боці сервера);
- в) MongoDB (система керування базами даних);
- г) React (бібліотека для створення клієнтських веб-застосунків)[3];
- д) MobX (бібліотека для реалізації реактивної обробки даних)[4]
- е) React-router (бібліотека для розробки веб застосунків з лише одною сторінкою із віртуальною навігацією);
- ж) Styled components (бібліотека побудови стилів для клієнтських веб-застосунків).[5]

Результат розробки програмного модуля показано на рисунку (рис 3.2).

Вихідний код програмного модуля являє собою набір файлів з програмним кодом мовою JavaScript і умовно розділений на дві директорії: server та client, як було описано в попередньому розділі.

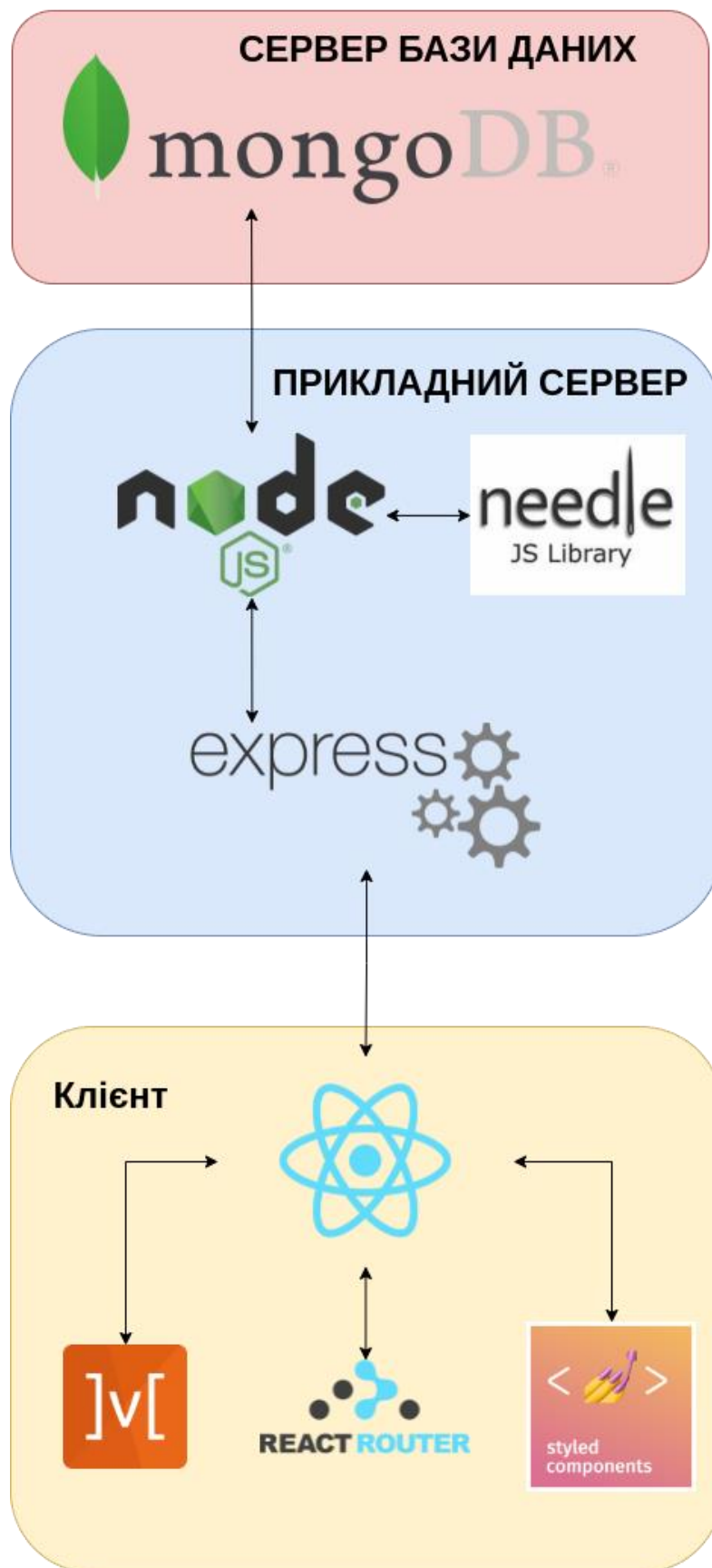


Рис 3.2 Архітектура програми

### 3.3 Інтерфейс програмного додатку

Інтерфейс програмного додатку являє собою набір веб сторінок і складається з таких вікон:

- головна сторінка, що містить слайдер з зображеннями (рис. 3.3) та карту з позначеними на ній ВНЗ (рис. 3.4.);
- сторінка новин (рис. 3.5);
- сторінка корисної інформації (рис. 3.6);
- сторінка із списком ВНЗ (рис. 3.7);
- сторінка ВНЗ (рис. 3.8);
- форма запитань (рис. 3.9.);
- сторінка керування (рис. 3.10).

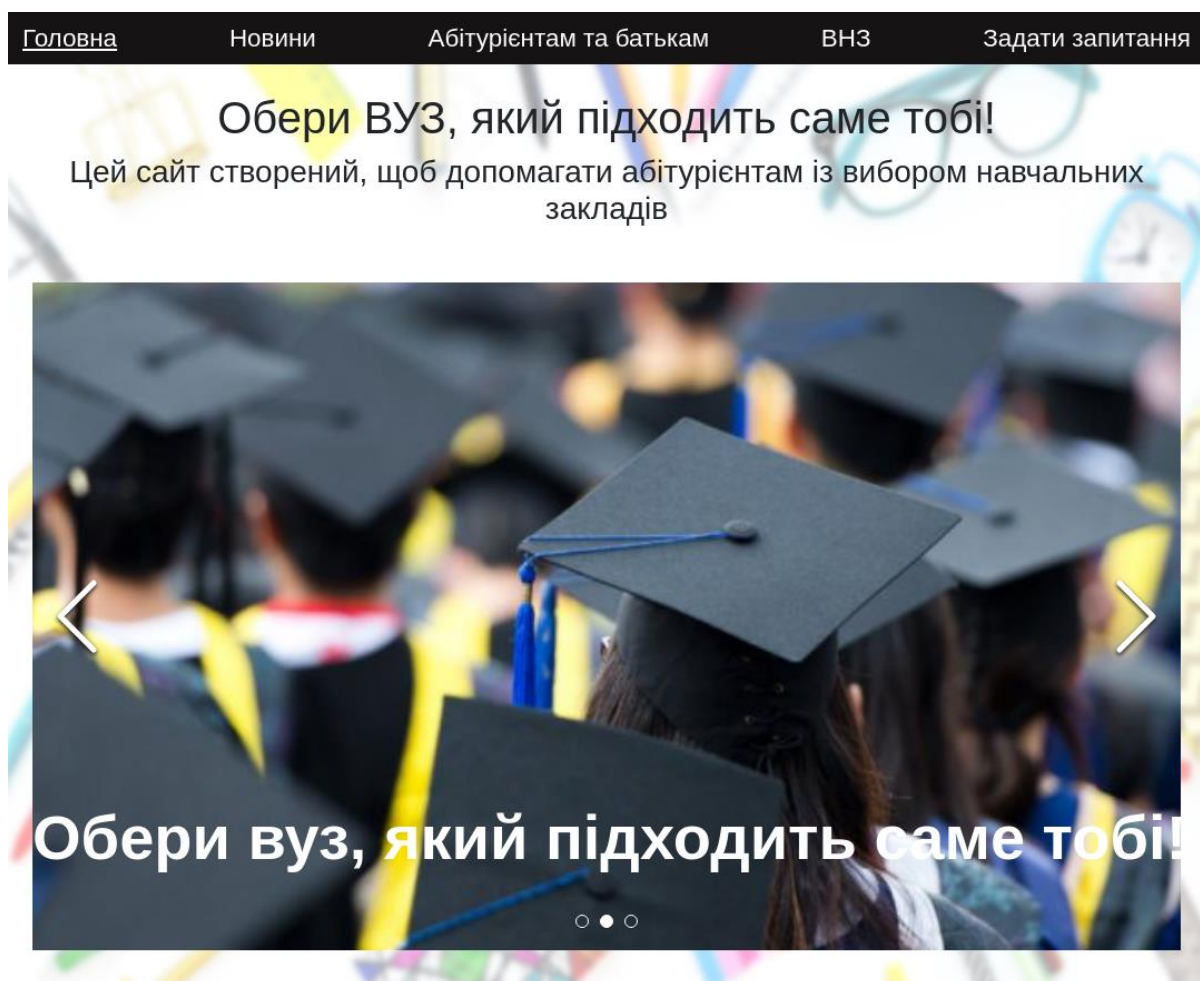


Рис 3.3 Головна сторінка

## Обирайте серед сотень начальних закладів України!

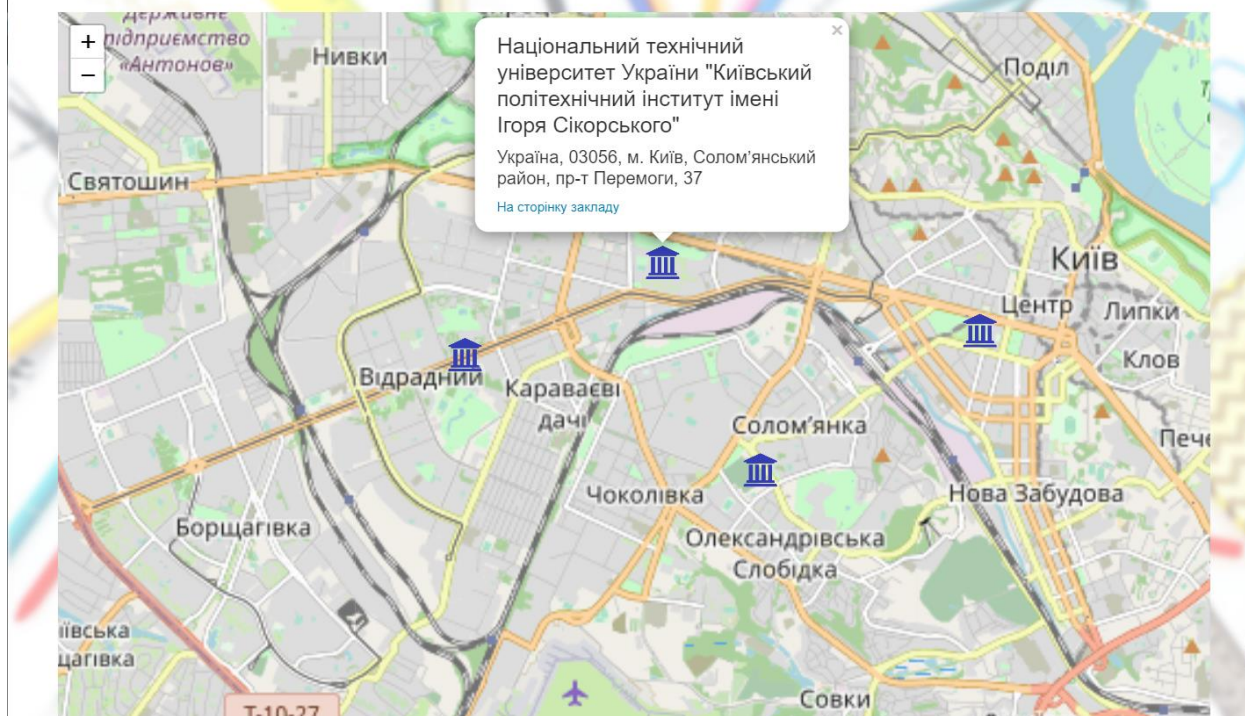


Рис. 3.4. Карта на головній сторінці

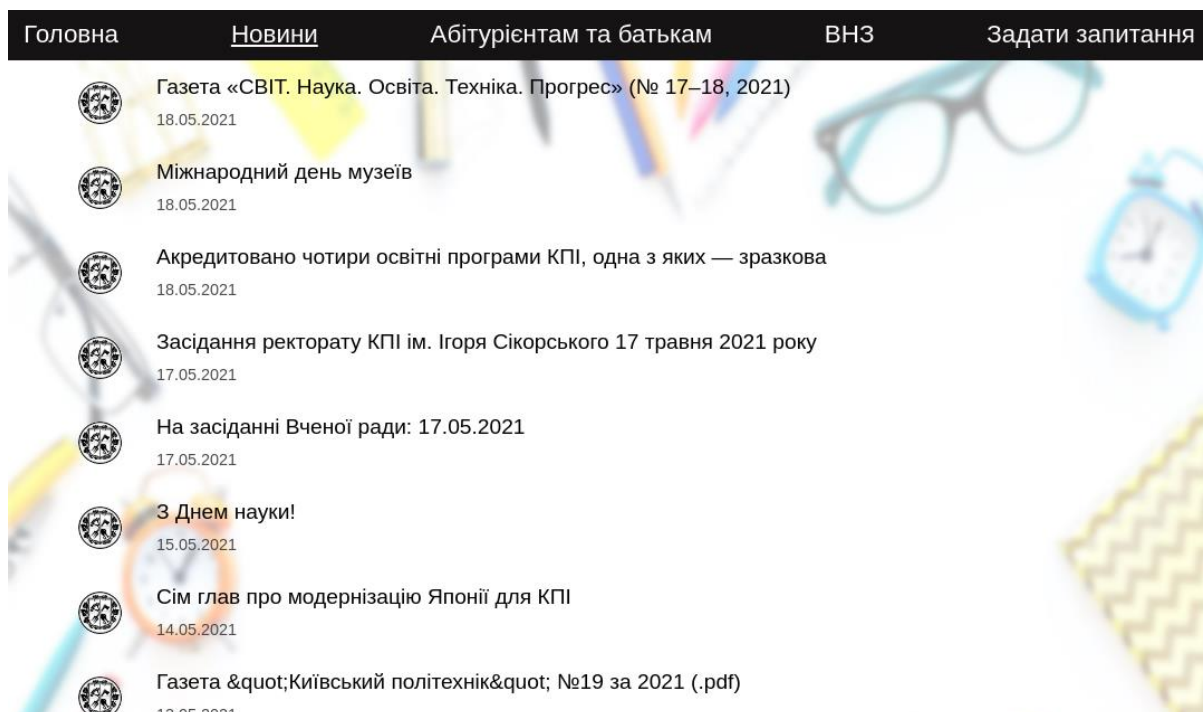


Рис 3.5 Сторінка новин



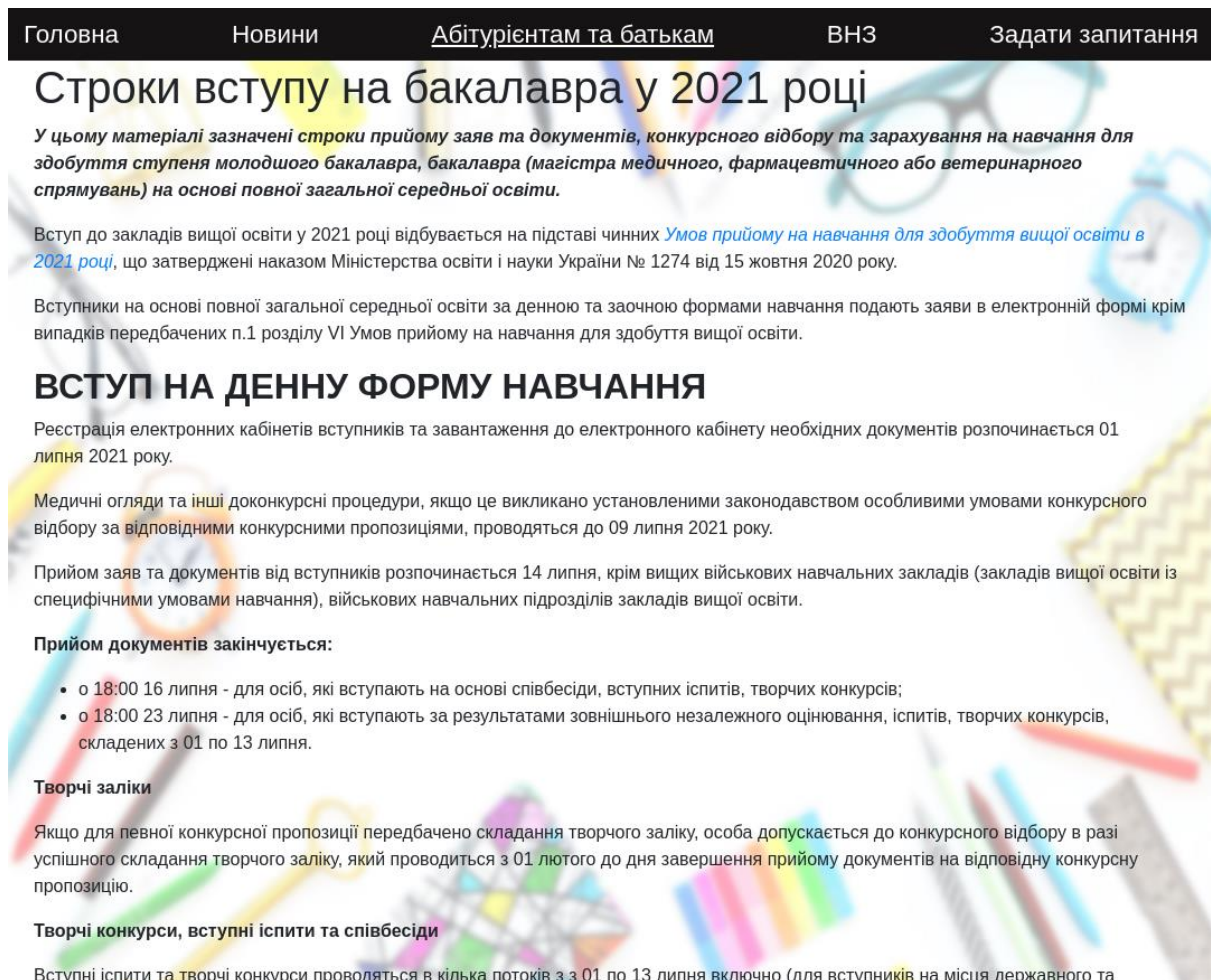


Рис 3.6 Сторінка корисної інформації

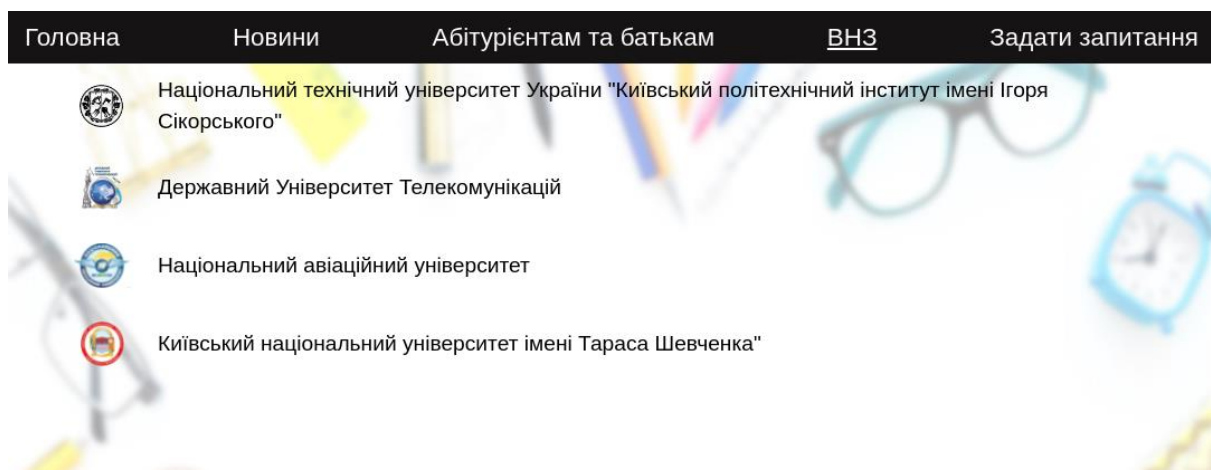
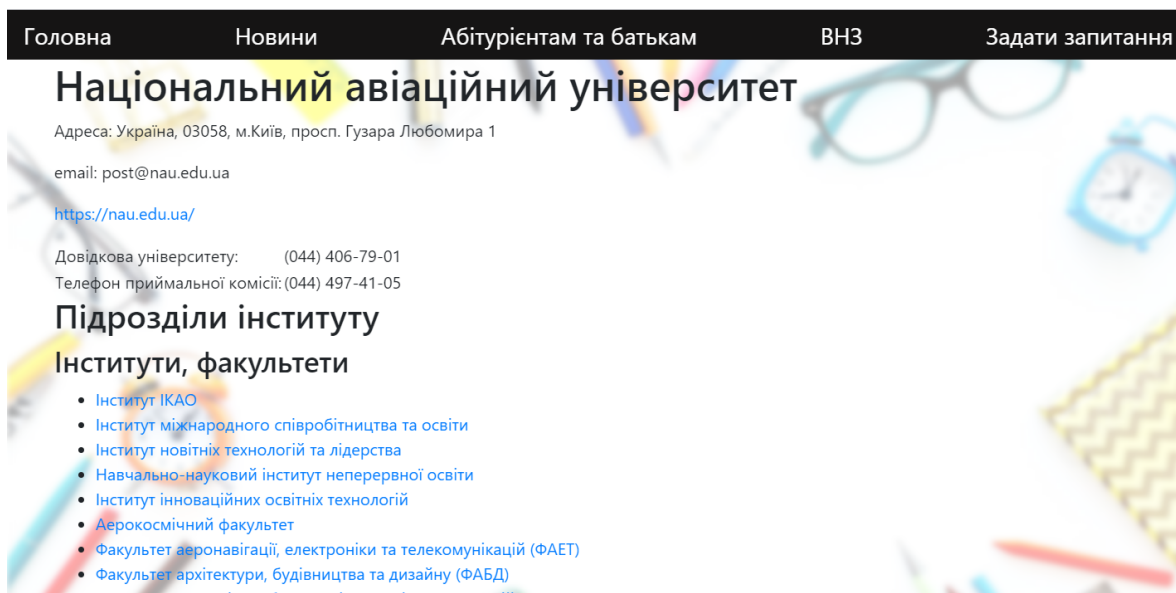


Рис 3.7 Сторінка вузів

Сторінка "ВНЗ" має список з посилань на сторінки з структурованою інформацією про обраний вищий навчальний заклад, що складаються з

загальної інформації, списків посилань на структурні підрозділи ВНЗ, списку новин та картою з відміткою даної установи на ній.



Головна      Новини      Абітурієнтам та батькам      ВНЗ      Задати запитання

## Національний авіаційний університет

Адреса: Україна, 03058, м.Київ, просп. Гузара Любомира 1

email: [post@nau.edu.ua](mailto:post@nau.edu.ua)

<https://nau.edu.ua/>

Довідкова університету: (044) 406-79-01  
Телефон приймальної комісії: (044) 497-41-05

### Підрозділи інституту

#### Інститути, факультети

- [Інститут ІКАО](#)
- [Інститут міжнародного співробітництва та освіти](#)
- [Інститут новітніх технологій та лідерства](#)
- [Навчально-науковий інститут неперервної освіти](#)
- [Інститут інноваційних освітніх технологій](#)
- [Аерокосмічний факультет](#)
- [Факультет авіонавігації, електроніки та телекомунікацій \(ФАЕТ\)](#)
- [Факультет архітектури, будівництва та дизайну \(ФАБД\)](#)
- [Факультет екологічної безпеки, інженерії та техніки](#)

Рис. 3.8. Сторінка інформації про певний ВНЗ

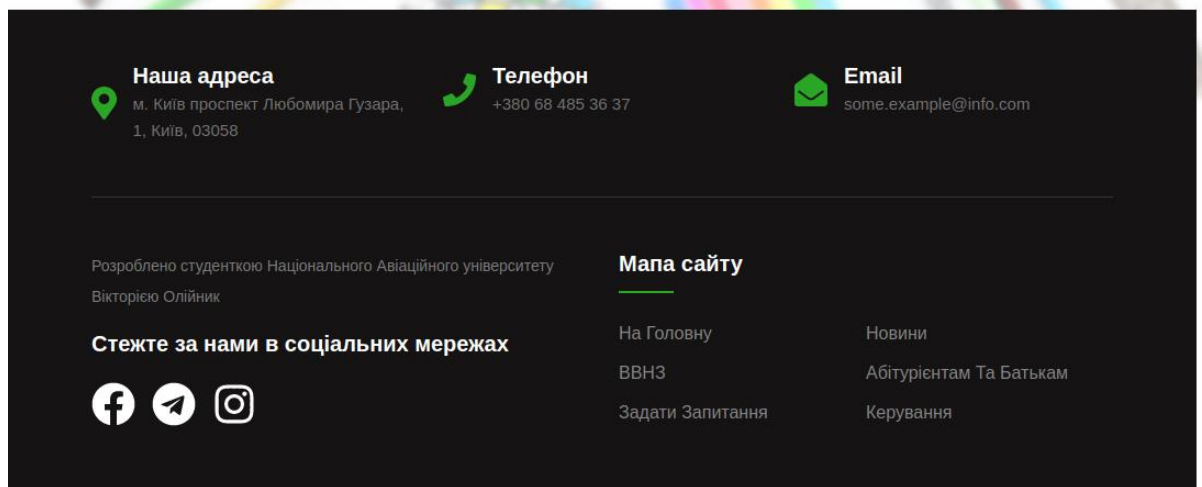
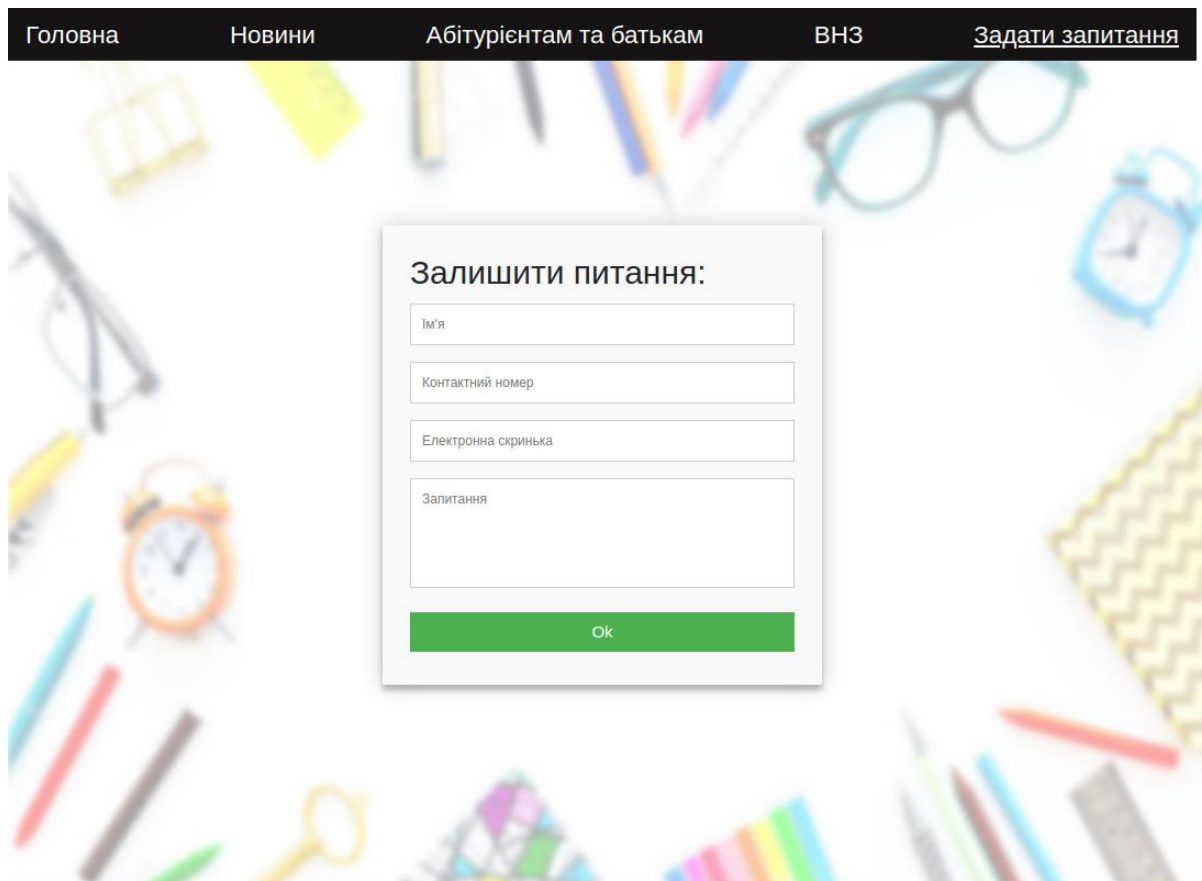


Рис 3.9 Сторінка подачі запитань

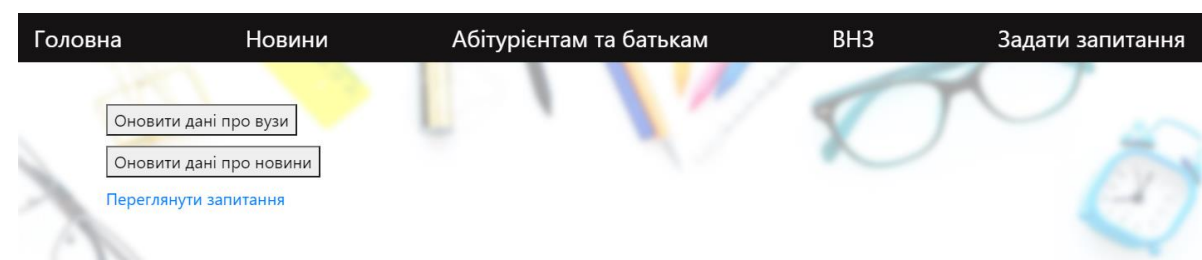


Рис. 3.10. Сторінка керування

### **3.4 Оцінка ефективності розробленого програмного додатку**

Методика проведення оцінки ергономічності інтерфейсу системи.

Оцінювання ергономічності інтерфейсу проводиться методом анкетного опитування групи експертів.

Експертам пояснюється порядок роботи системи та принцип виконання тесту. Експерти візуально та шляхом виконання тестових завдань оцінюють ергономічність інтерфейсу системи. Результати тестування заносяться в таблицю.

Оцінці підлягають наступні питання:

- розміщення елементів на сторінках;
- інформативність цих елементів;
- надлишковість інтерфейсу;
- зручність виконання операцій.

Компактне розміщення елементів характеризує можливість перегляду найрелевантніших даних у робочій області без виконання додаткових дій: прокрутки, натискання на кнопки, перемикання між вікнами.

Розташування елементів, які виконують однакову функцію в різних робочих областях, їх координати, назви, дизайн та розташування повинні бути однаковими у всіх робочих областях.

Інформативність елементів визначається наявністю необхідних даних у робочих областях.

Надмірність інтерфейсу характеризує наявність у робочій області непотрібних кнопок, написів та вікон.

Зручність виконання операцій можна досягти шляхом практичного відпрацювання тестових завдань. Оцінка проводиться за такими питаннями:

– кількість виконання кліків, рухів миші й натискань клавіш (якщо порівнювати з паперовим бланком): менша – +20, дорівнює – +10, більша – мінус 2, значно більша – мінус 20;

– максимальна відстань між елементами при послідовному переході між ними: менша чверті ширини робочої області – +20, – +10, більша – мінус 2, значно більша – мінус 20;

– сумарне перенесення курсора миші для однієї операції: менше ширини робочої області – +20, дорівнює – +10, більший – мінус 2, значно більший – мінус 20;

– загальне враження: подобається – +20, задовольняє – +10, не подобається – мінус 5.

На основі отриманих даних складається таблиця результатів, критерій висновку якої залежить від кількості балів в сумі і визначається таким чином:

- вище 4 балів – приймається;
- від мінус 40 до +4 балів – приймається;
- нижче мінус 40 балів – не приймається.

В результаті проведення тестування були отримані результати, приведені в таблиці 3.1.

Таблиця 3.1.

## Результати проведення тестування

Найменування операції	Оцінка в балах				Сума балів	Висновок
	Компактність розміщення	Стабільність розміщення	Порядок розміщення	Загальне враження		
Розташування елементів	20	10	-2	10	38	Приймається
Інформативність	5	10	5	10	30	Приймається
	Зайві кнопки	Зайві написи	Зайві вікна	Загальне враження		
Надлишковість	20	10	20	10	60	Приймається
	Кількість рухів	Відстань між елементами	Сумарний пробіг	Загальне враження		
Зручність	10	10	10	-2	28	Приймається

## **Висновки до розділу**

У сфері розробки веб-сервісів представлено програмне рішення реалізується застосуванням патерну проектування *Model-View-ViewModel*. У цьому розділі описано архітектуру програмного застосунку та принципи роботи ключових технологій. Застосування цього дозволить ефективно обробляти велику кількість http запитів та видавати на них інформативну інформацію..

Побудова архітектури на основі веб-сервісу має гнучку будову для забезпечення можливості швидкої розробки, тестування, розгортання і подальшого супроводження. Відповідно до архітектурного стилю і патернів проектування було спроектовано Веб-агрегатор інформації про Вищі навчальні заклади України на базі платформи Node.js. Результати процесу проектування представлені діаграмою класів та послідовностей. Розробка програмного модулю здійснювалася із застосуванням технологій динамічного обміну даними і базувалася на парадигмі реактивного програмування.

## ВИСНОВКИ

Під час розробки дипломного проекту на тему "Веб-агрегатор інформації про Вищі навчальні заклади України на базі платформи Node.js" було вирішено такі завдання:

- проаналізовано предметну область та відомі підходи до реалізації кваліфікаційної роботи;
- проаналізовано популярні мови програмування і підходи до розробки програмного забезпечення;
- здійснено порівняльний аналіз схожих рішень;
- визначено вимоги до програмного продукту;
- спроектовано структуру проекту, бази даних і схему розгортання готового програмного рішення;
- розроблено діючий прототип веб-агрегатора інформації про Вищі навчальні заклади України;
- проведено оцінку ефективності розробленого програмного продукту.

Оцінка готового програмного продукту показала його готовність до практичного застосування