

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра комп'ютерних наук та програмної інженерії

**РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ РЕЙТИНГОВОЇ
СИСТЕМИ PUBLICATION**

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу 441 групи

Спеціальності: 121 Інженерія програмного
забезпечення

Освітньо-професійної програми:

Інженерія програмного забезпечення

Куриленко М.М.

Керівник: кандидат пед.н., доцент

Вінник М.О.

Співкерівник: Співаковський О. В.

Рецензент: Щедролосьєв Д.Є., керівник
центру компанії DataArt

Херсон – 2022

ЗМІСТ

ЗМІСТ	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	3
ВСТУП	4
РОЗДІЛ 1. СИСТЕМИ ПОБУДОВИ НАУКОМЕТРИЧНИХ РЕЙТИНГІВ.	5
1.1 Основні наукометричні показники.	5
1.2 Наукометричні бази даних	6
1.3 Рейтингова система Publication	8
РОЗДІЛ 2. АНАЛІЗ ТЕХНОЛОГІЙ СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ НА REACT NATIVE.	11
2.1 React архітектура	11
2.2 Virtual DOM	11
2.3 Чому React Native?	11
2.4 Клієнт-серверна архітектура	12
2.5 Способи та формати передачі даних	13
РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ РЕЙТИНГОВОЇ СИСТЕМИ PUBLICATION.	15
3.1 Особливості розробки мобільних додатків.	15
3.2 Використані технології.	15
React Native	16
TypeScript	17
Axios	18
React Native Elements	18
Ant Design	19
Nest.js	19
MongoDB	20
React Native Testing Library	20
ВИСНОВОК	26
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	27

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

SCI	Індекс наукового цитування.
JCR	Журнал звітів про цитування.
API	Спеціальний протокол для взаємодії комп'ютерних програм, який дозволяє використовувати функції однієї програми всередині іншої.
ANSI	Американський національний інститут стандартів - об'єднання американських промислових і ділових груп, що розробляє торговельні та комунікаційні стандарти.
HTTP	Протокол прикладного рівня передачі даних.
SOAP	Протокол обміну структурованими повідомленнями у розподіленому обчислювальному середовищі.
XML	Мова розмітки, що розширюється.
ASCII	Назва таблиці, в якій деяким поширеним друкованим та недрукованим символам зіставлені числові коди.
JSON	Текстовий формат обміну даними, що базується на JavaScript.

ВСТУП

Актуальність – даної теми зумовлена тим, що на сьогоднішній день з'явилася потреба в зручному та інтуїтивно зрозумілому мобільному додатку для студентів та викладачів, зацікавлених у розробці рейтингових систем, науко метричних індексів університетів та розробці інформаційної системи показників наукової діяльності.

Сучасний світ часто характеризують як інформаційно-технологічний. Різні автори пишуть про постійний потік інформації, в якому стало важко орієнтуватися, знаходити потрібні видання, наукові роботи окремих вчених. Велика увага приділяється сьогодні вивченню еволюції науки за допомогою численних вимірювань та статистичної обробки наукової інформації (кількість наукових статей, опублікованих в даний період часу, цитування і т. д.). Наукова дисципліна, що займається подібним вивченням, дістала назву «наукометрія» – дослідження науки за допомогою кількісних методів. Наукометричні методи допомагають порахувати наукові результати дослідників та класифікувати ці результати за різними параметрами.

Мета дослідження – розробка мобільного додатку для рейтингової системи publication. В процесі досягнення мети перед нами постали наступні завдання:

- виконати аналіз побудови системи науко метричних рейтингів.
- дослідити особливості розробки мобільних додатків на React Native
- розробити мобільний додаток для рейтингової системи publication.

Об'єкт дослідження – мобільний додаток для рейтингової системи publication.

Предмет дослідження – розробка мобільного додатку для рейтингової системи Publication за допомогою фреймворку React Native.

РОЗДІЛ 1. СИСТЕМИ ПОБУДОВИ НАУКОМЕТРИЧНИХ РЕЙТИНГІВ.

1.1 Основні наукометричні показники.

«Індекс цитування (Science Citation Index — SCI) система Філадельфійського інституту наукової інформації, в основу якої покладені зв'язки між документами по прямим, зворотним і перехресним посиланням.

Система для обчислення SCI (або його інтернет-версія ISI Web of Knowledge) містить бібліографічні описи всіх статей з наукових журналів, що входять до переліку JCR, і висвітлює в основному публікації з фундаментальних галузей науки в провідних міжнародних і національних журналах.»[2]

Індекс Хірша —

це показник авторського рівня, який вимірює як продуктивність, так і вплив цитування публікацій вченого. Індекс h корелює з очевидними показниками успіху, такими як здобуття Нобелівської премії, прийняття на дослідницькі стипендії та зайняття посад у провідних університетах. Показчик базується на наборі найбільш цитованих робіт вченого та кількості посилань, які вони отримали в інших публікаціях. Індекс також можна застосувати до продуктивності та впливу наукового журналу, а також групи вчених, наприклад відділу, університету чи країни. Індекс був запропонований у 2005 році Хорхе Е. Хіршем, фізиком з Каліфорнійського університету в Сан-Дієго, як інструмент для визначення відносної якості фізиків-теоретиків, і його іноді називають індексом Хірша або числом Хірша.

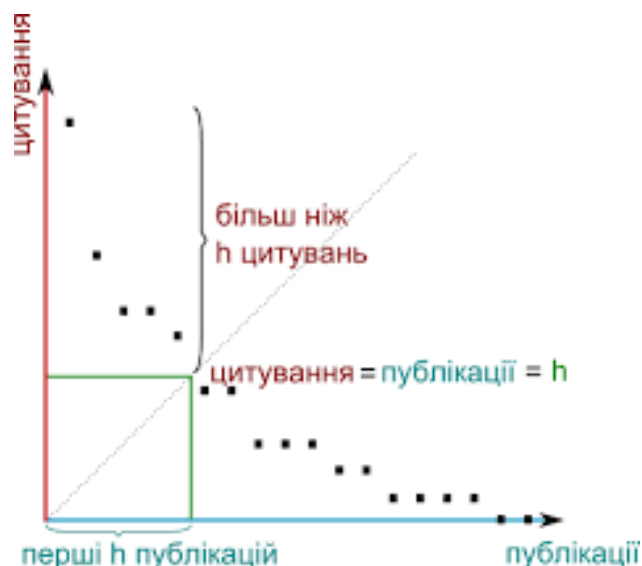


Рис. 1. Індекс Хірша

«Коефіцієнт впливовості, імпаکت-фактор» — коефіцієнт співвідношення цитування наукових журналів. Часто використовується як оцінка важливості журналу в певній галузі. Розроблений у 1975 році для бази даних «Journal Citation Reports», є зареєстрованою торговою маркою компанії «Інститут наукової інформації», яка є власністю компанії «Clarivate Analytics».

Обчислення імпакт-фактору відбувається лише за базою даних «Journal Citation Reports», інші наукометричні коефіцієнти, які розраховуються за різними базами, наприклад, Google Scholar чи Scopus, будуть мати інші значення.»[3].

1.2 Наукометричні бази даних

Наукометрія –

це підхід до вимірювання та аналізу різних аспектів науки. Хоча термін «наукометрія» часто використовується як синонім «аналіз цитування», були розроблені інші методи оцінки науки та її впливу. Нижче ви знайдете деякі з найбільш часто використовуваних традиційних наукометрій:

- ISI Web of service

- Google Scholar
- Scopus
- ResearchGate

Scopus – це єдина найбільша у світі абстрактна база даних, яка індексує понад 17 000 одиниць науково-технічних і медичних журналів близько 4 000 міжнародних видавництв. Система Scopus покликана підтримувати ефективний робочий процес дослідників, допомагаючи їм: знаходити нові статті зі сфери своєї спеціалізації; знайти інформацію про Автор; аналізувати видавничу діяльність у предметній галузі; відстежувати цитування; переглядати h-індекс; визначити найбільш цитовані статті та авторів; оцінити актуальність дослідження. Scopus дає змогу дослідникам об'єднувати свої статті під одним профілем.

Google Scholar є вільно доступною пошуковою системою, яка індексує повні тексти наукових публікацій усіх форматів і дисциплін. Google Scholar виконує не лише інформаційну, а й наукометричну функцію. Від списків результатів ми можемо отримати інформацію, як багато та які документи пов'язані з публікацією в базі даних Google Scholar. Число відображає ступінь авторитетності та публічності видання.

Web of Science – Міжнародна створена база даних цитування, представлена компанією Thomson Reuters. Web of Science дає можливість пошуку серед 12 000 журналів і 148 000 матеріалів конференцій у галузі природничих, соціальних, гуманітарних і мистецтвознавчих наук, що дозволяє отримати найбільш актуальну інформацію. Крім пошуку, Web of Science встановлює еталонний зв'язок між конкретним дослідженням з використанням цитованих матеріалів і тематичними зв'язками між статтями встановлений авторитетними дослідниками, що працюють у цій галузі. Це найширша база рефератів.

ResearchGate – вважається одним із найпривабливіших академічних соціальних мереж для наукової спільноти. Останнім часом він значно

покращив інтерфейси, орієнтовані на користувача, щоб зробити його більш привабливим для вчених усього світу. Перегляд показників, пов'язаних зі статтею (наприклад, імпаکت -фактор, цитати тощо) є важливою функцією в цій мережі. Аналогічно, публікація у відкритому доступі додала більшу видимість роботи та легкий доступ до інформації, пов'язаної з дослідженнями. Крім того, наукова спільнота була дуже зацікавлена у просуванні своєї роботи та демонстрації її впливу на інших за допомогою надійних наукометричних заходів. Однак із зростанням ринку публікацій та вдосконалень у сфері досліджень ця спільнота стала жертвою кіберзлочинів у вигляді журналів-привидів, підроблених видавців та фальшивих заходів впливу.

1.3 Рейтингова система Publication

У поточній версії системи реалізована можливість обробки індикаторів вченого на основі даних Scopus і Google Scholar. Алгоритм автоматичного пошуку для посилань на профілі українських вчених розроблено алгоритм автоматичного розподілу профілів науковців за назвою організації, в якій вони працюють, реалізовано автоматичне формування рейтингів кафедр, факультетів та наукових досліджень команди, також створено можливість надсилати повідомлення на електронну пошту вченим про зміни академічних індексів. Наукометричними індексами, на основі яких базуються рейтинги в системі, є:

- h-індекс (Scopus&Goggle Scholar). Індекс h що базується на найбільшій кількості робіт, які мали принаймні однакову кількість цитувань;
- Цитування (Scopus& Goggle Scholar). Кількість загальних посилань документів, які індексуються системою;
- i10-індекс (Goggle Scholar). Кількість посилань за документами, які мають десять або більше цитат;

Наразі в Scopus оброблено близько 3000 профілів вчених системи, з яких 680 визначено як укр. профілі. Автоматична обробка знайдених профілів дозволила побудувати рейтинг українських науковців за їхніми індексами в Scopus. Відсортувавши результати за належністю вчених до університету, на прикладі ХДУ було реалізовано можливість автоматичного створення рейтинги кафедр, факультетів та наукових досліджень колективів університету (рис.2).

Name	Belonging	h-index	citation	docum
Shapoval, Galina S.	Institute of Bioorganic Chemistry and Petrochemistry ...	9	citations by 520 documents	83
Shapoval, Anatoly N.	Institute of Plasma Physics, Kharkov, Kharkov, Ukrai...	9	citations by 162 documents	58
Shy	Name Department	h-index	citation	document
Use	Кафедра ботаніки	5	citations by 8 documents	15
Shz	Кафедра інформатики, програмної інженерії та еко...	2	citations by 9 documents	18
Shz	Кафедра фізики та методики її навчання	2	citations by 16 documents	9
Pol	Кафедра екології та географії	2	citations by 13 documents	6
	Кафедра алгебри, геометрії та математичного анал...		citations by 0 document	5
	Кафедра органічної та біологічної хімії		citations by 0 document	2

Рис. 2. Приклад роботи системи

Аналіз наукової діяльності вчених ХДУ свідчить про те, що найбільшу кількість публікацій мають викладачі кафедри інформатики, програмної інженерії та Економічної кібернетики. А найбільший h-індекс мають викладачі кафедри Ботаніки. Побудова подібних рейтингів за Goggle Scholar наразі можлива лише за наявності посилань на нього, на відміну від Scopus, самому автору необхідно зареєструватися в системі. Таким чином, було оброблено записи, надані посилання науковцями університету. Зараз для перегляду та аналізу доступні показники вчених факультетів фізики, математики та інформатики ХДУ, факультету дошкільної та початкової освіти ХДУ, кафедри філософії та соціології, гуманітарних наук. Наступним етапом розвитку та вдосконалення системи буде:

- автоматична інтеграція та аналіз інформації про наукометричні показники вченого у разі дублювання його профілю в цих наукометричних базах даних;

- удосконалення алгоритму обробки інформації за наукометричними показниками організації, наукові колективи у разі неправильного написання або зміни їх назви;
- удосконалення алгоритму пошуку посилань на профілі науковців за їх належність до країни;
- можливість автоматичного порівняння показників наукової діяльності вчених, науково-дослідні групи, організації та структурні підрозділи.

Реалізація алгоритму автоматичного пошуку посилань до наукового профілю членства певної країни та організації та підвищення ефективності алгоритму дозволяє говорити про можливість вибірки та обробки великих обсягів інформації. Таким чином, у наступній версії в система передбачає побудову сховища даних на принципах Big Data і MapReduce. Це, у свою чергу, сформує рейтинги наукової діяльності вчених, наукові групи та організації з мінімальними ресурсами та витратами часу.

РОЗДІЛ 2. АНАЛІЗ ТЕХНОЛОГІЙ СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ НА REACT NATIVE.

2.1 React архітектура

React і React Native використовують архітектуру Flux для обробки потоку даних у програмі. Ця технологія була створена компанією Facebook як альтернатива традиційній моделі MVC. Flux відрізняється від MVC тим, що дотримується концепції односпрямованого потоку даних, це полегшує знаходження будь-яких помилок.

Flux запобігає каскадним оновленням, які часто трапляються в моделі MVC. Єдиний диспетчер обробляє всі зміни в різних сховищах на рівні даних додатку. Додаткові дії не запускаються, поки сховище не завершить оновлення за допомогою значень даних. Перегляд програми повторно відтворюється з відповідними змінами.

2.2 Virtual DOM

В React і React Native, UI виражений як функція даних у програмі. Тому розробнику не потрібно дбати про те, як вміст інтерфейсу користувача буде відображатися в DOM. Це досягається завдяки наявності проміжної DOM у рамках React, відома як Virtual DOM. Пакетуючи всі необхідні зміни, віртуальний DOM може порівняти стани до і після цих змін та визначити мінімальну кількість маніпуляцій з DOM, які необхідно виконати.

2.3 Чому React Native?

React Native дозволяє розробнику налаштувати простий і робочий додаток. Це ефективний спосіб створення мобільних додатків. Завдяки тому, що розробник може бачити оновлення внесених змін в режимі реального часу на мобільному пристрої, UI та UX можна тестувати ефективніше та швидше. Розробник також може використовувати React Native як розширення існуючого нативного проекту, що полегшує підтримку кількох

платформ за допомогою однієї кодової бази. Інструменти всередині React Native дають змогу використовувати специфічні для платформи функції, ізольовані від іншої платформи, вони не заважають один одному.

2.4 Клієнт-серверна архітектура

Поява архітектури клієнт-сервера, як і багато інших нових технологій, супроводжувалося народженням нової термінології:

- Прикладний програмний інтерфейс (Application Programming Interface, API) – це набір функцій і підпрограм, забезпечуючих взаємодію клієнтів і серверів;
- Клієнт - об'єкт, що робить запити інформації по мережі. Як правило, це персональний комп'ютер або робоча станція, що просить інформацію з серверу;
- Проміжне програмне забезпечення - набір драйверів, прикладних програмних інтерфейсів і програмного забезпечення, що дозволяє усунути взаємодію між клієнтськими додатками та сервером на пряму;
- Реляційна база даних - база даних, в якій доступна інформація обмежена вибором рядків, задовольняючих визначеним критерієм пошуку;
- Сервер - комп'ютер (як правило, високопродуктивна робоча станція, міні-комп'ютер або мейнфрейм), зберігаючий інформацію, з якою працюють мережеві клієнти;
- Мова структурованих запитів (SQL) – мова, розроблена корпорацією IBM та стандартизована інститутом ANSI мова для створення, управління та зміни баз даних;
- Архітектура «клієнт сервер» - мережеве оточення, у якому керування даними здійснюється на серверному вузлі, а іншим вузлам надається доступ до даних.

- Обчислювальна модель клієнт-сервер - спільна з клієнтом обробка запитів клієнта. сервером та повернення результатів клієнту. У цій моделі обробка даних програмою розподілена (не обов'язково порівну) між клієнтом та сервером. Обробка даних ініціюється і частково керується клієнтом, але не в режимі «ведучий-ведений», а швидше в режимі співпраці.

Архітектура "клієнт - сервер" використовується для організації мережових додатків. Далі буде розглянуто структуру мережного проходження.

У рамках багаторівневого представлення обчислювальних систем можна виділити три групи функцій, орієнтованих на вирішення різних під задач:

- 1) функції введення і відображення даних (забезпечують взаємодію з користувачем).
- 2) прикладні функції, притаманні даній предметній області;
- 3) функції управління ресурсами (файловою системою, базою даних)

2.5 Способи та формати передачі даних

Для передачі даних між клієнтом та веб-сервером використовується HTTP-протокол – це протокол передачі. HTTP використовується також як «транспорт» для інших протоколів прикладного рівня, таких як SOAP.

У ході розробки було виділено два підходи до організації клієнт-серверної взаємодії - це SOAP та REST-підходи.

SOAP активно використовує XML для кодування клієнт-серверних взаємодій, а також строгу типізацію даних, що гарантує їх цілісність, використовуючи для цього формат XML документа. REST надає більшу свободу дій, оскільки запити та відповіді можуть передаватися до ASCII, XML, JSON або будь-яких інших форматів, що розпізнаються одночасно і клієнтом, і сервером. В результаті пакети запитів і відповідей у REST мають набагато менші розміри, ніж відповідні пакети SOAP.

У моделі SOAP рівень передачі даних протоколу HTTP є «пасивним спостерігачем», та її роль обмежується передачею запитів SOAP від клієнта серверу з допомогою методу POST. Деталі сервісного запиту, такі як ім'я дистанційної процедури та вхідні аргументи, кодуються у тілі запиту. Архітектура REST, навпаки, розглядає рівень передачі даних HTTP як одну із сторін взаємодії, використовуючи існуючі методи HTTP, такі як GET, POST, PUT і DELETE. Тому запити REST у загальному випадку більш прості для формулювання та розуміння, оскільки вони використовують існуючі та добре зрозумілі інтерфейси HTTP.

РОЗДІЛ 3. РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ РЕЙТИНГОВОЇ СИСТЕМИ PUBLICATION.

3.1 Особливості розробки мобільних додатків.

Мобільний додаток - є програмним забезпеченням, яке спеціально розроблено для конкретної мобільної платформи (iOS, Android, Windows Phone та ін). Призначений для застосування на смартфоні, планшеті, розумних годинниках та інших мобільних пристроях. Процес створення мобільних додатків протікає так:

1. Написання мовою програмування високого рівня;
2. Компіляція в машинний код операційної системи для максимальної продуктивності;

При розробці додатків необхідно враховувати деякі особливості: робота мобільних пристроїв здійснюється на батарейках та осначені не такими потужними продуктивними процесорами, як у персональних комп'ютерів.

3.2 Використані технології.

Front-end розробка:

- React Native
- Redux
- TypeScript
- Axios
- React Native Elements
- Ant Design

Back-end розробка:

- Nest.js
- MongoDB

Тестування:

- React Native Testing Library

React Native



Рис. 3. Логотип React Native

Створення мобільного додатка часто вимагає від розробників створення одного додатку для Android і iOS, двох провідних операційних систем для мобільних пристроїв. Ці дві програми можуть мати однакову компоновку та логіку, але деякі компоненти інтерфейсу користувача (UI) будуть відрізнятися, і самі програми потрібно розробляти двома різними мовами. Цей процес жахливий, оскільки створення двох програм займає багато часу і вимагає двох різних наборів знань. Були спроби створити методики, сервіси чи фреймворки, щоб вирішити цю проблему, але ці гібриди не змогли забезпечити ефективне створення мобільних додатків.

React Native може створювати програми як для iOS, так і для Android шляхом компіляції коду, написаного на React. Отримані програми можуть спільно використовувати код і складаються з компонентів інтерфейсу користувача, які є унікальними для кожної платформи.

Redux



Рис. 4. Логотип Redux

Redux — це передбачуваний контейнер стану. Це означає, що причину зміни стану можна легко визначити. Redux використовує одне сховище даних замість багатьох різних, як інші flux бібліотеки. Це часто називають «одним джерелом правди», оскільки сховище є єдиним місцем, де можна отримати доступ до state. У міру того, як програми стають складнішими, вони починають використовувати масивніший state.

Redux робить зміни state передбачуваними. Це досягається шляхом встановлення кількох правил. По перше, state повинен бути імутабельний. По-друге, інформація про зміни state повинні містити дії. І по-третє використання reducer, які є чистими функціями, вони вказують як дії перетворюють state. Посилання на кілька state-ів можна зберігати, оскільки state у програмі Redux незмінний. Це полегшує впровадження історії.

TypeScript



Рис. 5. Логотип TypeScript

TypeScript — це розширення JavaScript, призначене для полегшення розробки великомасштабних програм JS. Хоча кожна програма JS є

програмою TS, TS пропонує систему модулів, класи, інтерфейси та багату систему поступового введення. Мета полягає в тому, щоб TS забезпечив плавний перехід для програмістів JS добре встановлені ідіоми програмування JS підтримуються без будь-яких серйозних переписів або анотацій. Одним цікавим наслідком є те, що система типів TS не є статично надійною за проектом.

Axios



Рис. 6. Логотип Axios

Axios — це HTTP-клієнт на основі промісів для `node.js` і браузера. Він ізоморфний, тобто, він може запускатися в браузері та в `nodejs` з однаковою кодовою базою. На стороні сервера він використовує власний `http`-модуль `node.js`, тоді як на клієнті (браузері) він використовує `XMLHttpRequests`.

React Native Elements

React Native Elements — це кросплатформний інструментарій React Native UI, який об'єднує в одному місці ряд чудових компонентів інтерфейсу користувача з відкритим кодом, створених розробниками. Пакет включає в себе цілу низку компонентів, таких як ціни, значоки, накладення, роздільник і панелі пошуку для платформи. Вони прості у використанні і гнучко налаштовуються. Реквізити для всіх компонентів визначаються в одному центральному місці, що дозволяє легко оновлювати або змінювати компоненти. Крім того, він може служити платформою, що об'єднує невеликі команди, які розробляють комерційні програми React Native з відкритим кодом.

Ant Design



Рис. 6. Логотип Ant Design

Ant Design — це повноцінна дизайн-система, візуальна мова. Зі своїми принципами, стайлгайдами і бібліотекою компонентів. Проект підтримується розробниками з Alibaba Group. Ant Design, написаний на TypeScript, стилізований за допомогою Less і портований на Angular і Vue.

Nest.js



Рис. 7. Логотип Nest.js

Nest (NestJS) — це платформа для створення ефективних, масштабованих додатків на Node.js. Він використовує прогресивний JavaScript, побудований на основі TypeScript і повністю підтримує його (проте все ще дозволяє розробникам кодувати на чистому JavaScript) і поєднує елементи ООР (об'єктно-орієнтоване програмування), FP (функціональне програмування) і FRP (функціональне реактивне програмування).

Nest використовує надійні фреймворки HTTP-сервера, такі як Express (за замовчуванням).

Nest забезпечує рівень абстракції вище цих поширених фреймворків Node.js (Express/Fastify), але також надає їхні API безпосередньо розробнику. Це дає розробникам свободу використовувати безліч модулів сторонніх розробників, які доступні для базової платформи.

MongoDB



Рис. 8. Логотип MongoDB

MongoDB — це документно-орієнтована база даних NoSQL, яка використовується для зберігання великих обсягів даних. Замість використання таблиць і рядків, як у традиційних реляційних базах даних, MongoDB використовує колекції та документи. Документи складаються з пар ключ-значення, які є основною одиницею даних у MongoDB. Колекції містять набори документів і функцій, що є еквівалентом таблиць реляційної бази даних.

React Native Testing Library



Рис. 9. Логотип React Native Testing Library

React Native Testing Library — це бібліотека для тестування React Native, натхненна бібліотекою тестування React. Оскільки React Native не запускається в середовищі браузера, основні запити реалізуються незалежно,

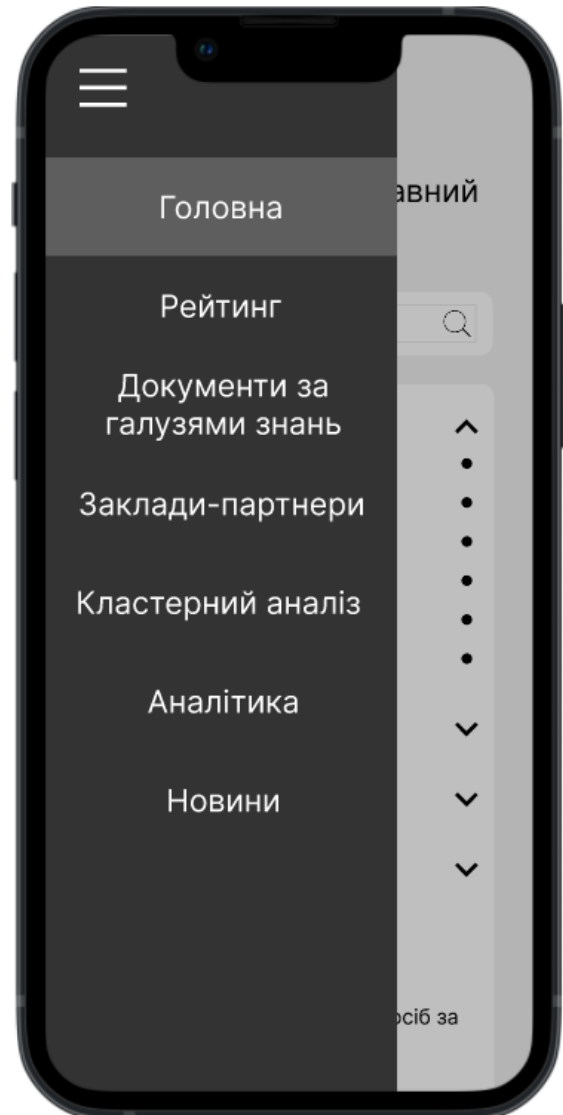
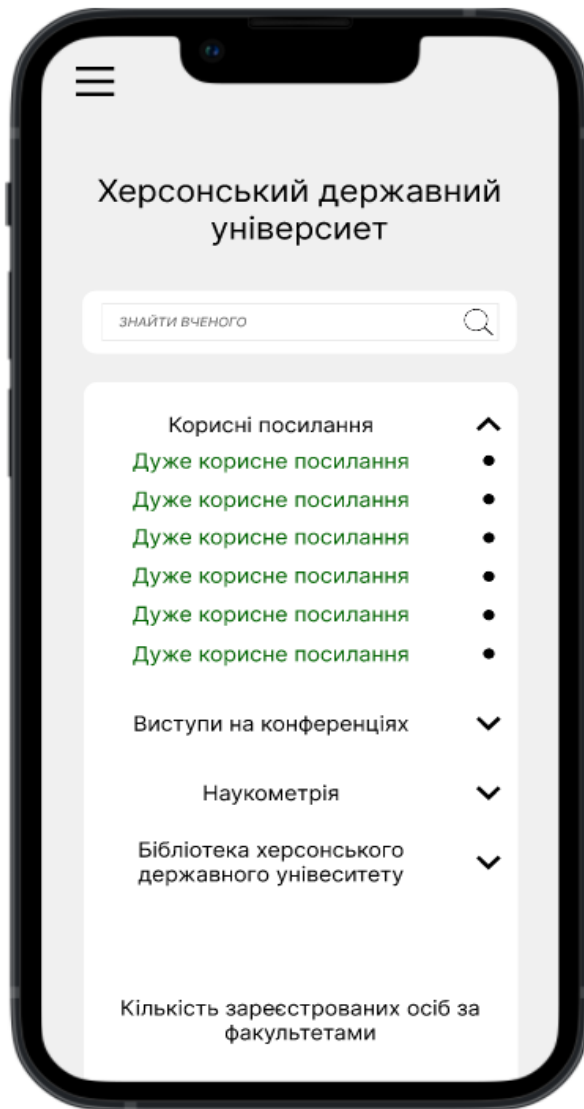
на відміну від інших обгорток, які використовують бібліотеку тестування DOM як основу.

3.3 Демонстрація розробленого додатку

На скріншотах показана робота додатку на прикладі телефону iPhone 11 Pro.

Рис. 10. Головна сторінка

Рис. 11. Меню на головній сторінці



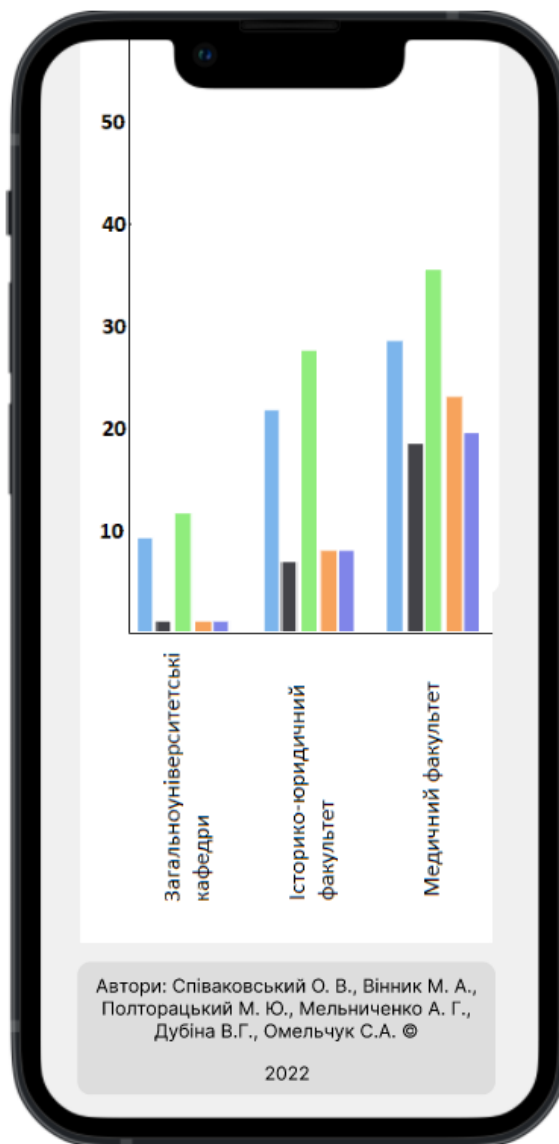
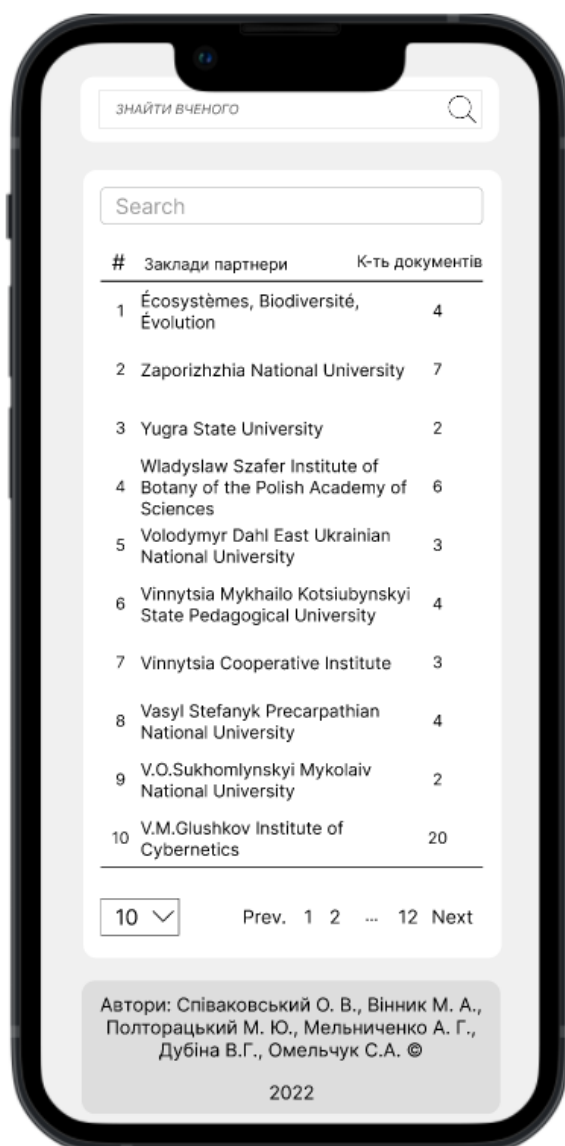


Рис. 12. Нижня частина головної сторінки

Рис. 13. Сторінка закладів-партнерів



3.4 Тестування додатку

React Testing Library - це набір допоміжних функцій, що дозволяє тестувати React-компоненти, не покладаючись на їхню внутрішню реалізацію. Такий підхід спрощує рефакторинг, а також підштовхує застосовувати найкращі практики щодо підвищення доступності. Незважаючи на те, що бібліотека не дозволяє робити «поверхневий» рендер компонента без дочірніх компонентів.


```
PASS src/components/__tests__/Container.test.js
PASS src/components/__tests__/List.test.js
PASS src/components/__tests__/Details.test.js
PASS src/components/__tests__/Header.test.js

Test Suites: 4 passed, 4 total
Tests:       10 passed, 10 total
Snapshots:  0 total
Time:        1.954s, estimated 2s
Ran all test suites related to changed files.

Watch Usage
> Press a to run all tests.
> Press p to filter by a filename regex pattern.
> Press t to filter by a test name regex pattern.
> Press q to quit watch mode.
> Press Enter to trigger a test run.
█
```

Тести по рендерингу компонентів, зворотнім викликам та станам компонентів були виконані успішно.

ВИСНОВОК

Аналіз React Native показав ефективність інструменту в розробці крос-платформових мобільних додатків за короткий проміжок часу. Інтерфейси користувача, які може створити React Native, ідентичні до тих, які доступні з нативною розробкою.

Створений додаток зміг надати кросплатформове рішення, яке буде використовуватися як клієнтський інструмент для надання інформації студентам та викладачам, зацікавлених у розробці рейтингових систем, наукометричних індексів університетів та розробці інформаційної системи показників наукової діяльності.

У ході виконання цієї роботи було отримано такі основні результати:

- Виконано аналіз побудови системи наукометричних рейтингів.
- Було досліджено особливості розробки мобільних додатків на React Native
- Розроблено мобільний додаток для рейтингової системи publication.

Основним інструментом для тестування мобільного додатку було використано бібліотеку React Native Testing Library.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The largest database of peer-reviewed literature.: [онлайн ресурс]. URL: <https://www.elsevier.com/solutions/scopus> (дата останнього звернення: 07.02.2022).
2. Національна бібліотека України: [он-лайн ресурс]. URL: <http://www.nbuv.gov.ua/node/1367> (дата останнього звернення: 15.02.2022).
3. Design and Development of Information System of Scientific Activity Indicators: [онлайн ресурс]. URL: <https://t.ly/i8Tp> (дата останнього звернення: 15.02.2022).
4. Design and Development of Information System of Scientific Activity Indicators: [онлайн ресурс]. URL: http://ekhsuir.kspu.edu/xmlui/bitstream/handle/123456789/4079/paper_2.pdf?sequence=1 (дата останнього звернення: 08.03.2022).
5. React Native application development: [онлайн ресурс]. URL: <https://www.diva-portal.org/smash/get/diva2:998793/FULLTEXT02.pdf> (дата останнього звернення: 08.03.2022).
6. React Native application development: A comparison between native Android and React Native: [онлайн ресурс]. URL: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A998793&dswid=-7910> (дата останнього звернення: 18.03.2022).
7. Introduction React Native: [онлайн ресурс]. URL: <https://reactnative.dev/docs/getting-started> (дата останнього звернення: 28.10.2022).
8. Getting Started with Redux [онлайн ресурс]. URL: <https://redux.js.org/introduction/getting-started> (дата останнього звернення: 03.04.2022).
9. Introduction NestJS: [онлайн ресурс]. URL: <https://docs.nestjs.com/> (дата останнього звернення: 05.04.2022).

10. Getting Started with Axios: [онлайн ресурс]. URL: <https://axios-http.com/docs/intro> (дата останнього звернення: 16.04.2022).
11. Introduction React Native Testing Library: [онлайн ресурс]. URL: <https://testing-library.com/docs/react-native-testing-library/intro/> (дата останнього звернення: 20.04.2022).
12. React Native Testing Library documentation: [онлайн ресурс]. URL: <https://callstack.github.io/react-native-testing-library/docs/getting-started/> (дата останнього звернення: 24.04.2022).