

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

РОЗРОБКА ЗАСТОСУНКУ ЗА ДОПОМОГОЮ БІБЛІОТЕКИ REACT

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки»

Керівник курсової роботи
старший викладач Борозенний С.О.
(прізвище та ініціали)

_____ (підпис)

“ ___ ” _____ 2022 р.

Виконав студент Винник А.І.

“ ___ ” _____ 2022 р.

Київ 2022

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ
Старший викладач
Борозенний С.О.

(підпис)

“ ___ ” _____ 2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Виннику А.І. факультету інформатики 4 курсу

ТЕМА: Розробка застосунку за допомогою бібліотеки React

Зміст ТЧ до курсової роботи:

1. Індивідуальне завдання
2. Вступ
3. Огляд сфери розробки веб-застосунків
4. Опис бібліотеки React
5. Розробка застосунку
6. Висновки
7. Список літератури
8. Додатки

Дата видачі “ ___ ” _____ 2022 р. Керівник

(підпис)

Завдання отримав

(підпис)

Тема: Розробка застосунку за допомогою бібліотеки React
Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на дипломну роботу.	18.10.2021	
2.	Огляд технічної літератури за темою роботи.	15.11.2021	
3.	Ознайомлення з бібліотекою React	25.01.2022	
4.	Вибір предметної області застосунку	29.04.2022	
5.	Обговорення предметної області з керівником	02.05.2022	
6.	Створення застосунку	06.05.2022	
7.	Написання текстової частини	15.04.2022	
8.	Здача курсової роботи	20.05.2022	

Зміст

Перелік прийнятих скорочень	5
Вступ	6
1. Особливості бібліотеки React	7
1.1. Загальний огляд	7
1.2. Переваги React.....	7
1.2.1. Компоненти.....	7
1.2.2. Віртуальний DOM	8
1.2.3. JSX.....	9
2. React Native	10
1.3. Безшовна крос-платформа.....	10
3. Створення веб-застосунку за допомогою бібліотеки React.....	11
3.1. Опис застосунку.....	11
3.2. Інтерфейс застосунку	12
Висновок	17
Список використаних джерел.....	18
Додатки	19

Перелік прийнятих скорочень

HTML – HyperText Markup Language

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

ООП – Об'єктно-орієнтоване програмування

DOM – Document Object Model/

JSX – JavaScript extension

JS - JavaScript

UI – User interface

API – Application Programming Interface

UWP – Universal Windows Platform

Вступ

Перший веб-сайт був створений влітку 1991 року британським спеціалістом. Його творцем був інформатик Тім Бернес-Лі. Власне кажучи, ним же було створено протокол HTTP, мову розмітки HTML, URI, та багато інших технологій. Перший у світі сайт мав наступний URL: “http://info.cern.ch.” та розміщував на собі інформацію про Всесвітню павутину.

Після цієї події веб-технології почали стрімко зростати, з’являлись нові мови програмування, застосунки ставали все складнішими. Так у 1995 році було винайдено одразу дві значні мови програмування веб-додатків PHP та JavaScript. Ці дві події дали величезний поштовх для розвитку всесвітньої павутини. Досить довгий час усі веб-застосунки створювались за допомогою PHP, JavaScript та HTML. Проте з розвитком технологій, з’являлась потреба у створенні все складніших веб-сайтів з великою кількістю елементів, удосконаленням дизайном та функціоналом.

У 2013 році компанія Meta випускає в реліз бібліотеку React, яка дозволяє створювати складний дизайн за допомогою UI компонентів, які створюються на базі JavaScript. Таким чином, з’явилась можливість програмувати великі веб-додатки набагато простіше, створюючи окремі компоненти багаторазового використання, з котрих складається цілісний застосунок. Грубо кажучи, можна провести аналогію з парадигмою ООП, у якій застосунок складається з багатьох об’єктів.

За допомогою React стало простіше створювати нові застосунки, та оновлювати вже існуючі, оскільки все розбито на багато компонентів, котрі легко модифікувати, та впроваджувати в нові проекти.

Ціллю моєї курсової роботи є ознайомлення з бібліотекою React, її особливостями та написання показового застосунку за її допомогою.

1. Особливості бібліотеки React

1.1. Загальний огляд

React – це безкоштовна JavaScript бібліотека з відкритим вихідним кодом для створення інтерфейсу веб-застосунків. Підтримується компанією Meta та спільнотою окремих розробників. React займається лише керуванням станом інтерфейсу і відтворенням цього стану в DOM, тому створення додатків React зазвичай вимагає використання додаткових бібліотек для маршрутизації, а також певної функціональності на стороні клієнта.

Дана бібліотека є декларативною, що дозволяє без складнощів програмувати комплексні інтерфейси. Розробник тільки має вказати, як певні складові застосунку будуть відображатись в різних станах, а React сам визначить які компоненти слід оновлювати, після зміни даних.

1.2. Переваги React

1.2.1. Компоненти

Одна з основних переваг React це – компоненти, котрі є незалежними один від одного, та відповідають тільки за власний стан. Це дозволяє створювати комплексні веб-застосунки, які складаються з багатьох самостійних компонентів. Таким чином, можна легко підтримувати, оновлювати версії додатку та повторно використовувати власний код в інших проектах.

React дозволяє створювати компоненти двома способами: як клас, або як функцію-компонент.

```
class Greetings extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```

Рисунок 1.1 – приклад React класу

```
function Greetings(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

Рисунок 1.2 - приклад функціонального React компонента

На рисунках 1.1 та 1.2 можна побачити приклад компоненту “Greetings”, який повертає повідомлення з привітанням для певного користувача. Обидва приклади є еквівалентними, проте можна побачити, що навіть на такому простому прикладі функціональний підхід дозволяє зменшити об’єм написаного коду. У оновленні React 16.8, у функціонального компонента з’явилась можливість використовувати так звані «хуки», які до цього були доступні тільки у класах. Це дало змогу використовувати стани за допомогою `useState()`, `useEffect()` та інших методів.

1.2.2. Віртуальний DOM

В React ми не працюємо з браузерним DOM, натомість ми взаємодіємо з віртуальним DOM, який аналогом звичайного, тільки знаходить він у пам’яті. Таким чином кожен React компонент, який ми створюємо вставляється у віртуальний DOM.

Це було зроблено з тої причини, що звичайний браузерний DOM працює не дуже ефективно з великою кількістю елементів на сайті. Так наприклад, якщо в застосунку є список зі ста елементів і нам необхідно змінити дані тільки на одному з них, звичайний DOM оновлює всю сторінку заради одного елемента. Враховуючи, що сучасні веб-застосунки мають велику кількість елементів, зміна декількох з них є досить неефективним для швидкодії роботи сайту.

При використанні React спочатку оновлюється віртуальний DOM, далі він порівнюється з DOM браузера і якщо є якісь зміни у віртуальному,

тоді оновлюється і головний DOM. Це значно прискорює роботу веб-застосунку.

1.2.3. JSX

JSX – це є розширення для мови JavaScript, яке дозволяє писати HTML код у симбіозі з JS, що дуже спрощує створення нових компонентів. JSX дозволяє не розділяти інтерфейс та логіку відображення у різних файлах, а дозволяє створювати компоненти у котрих буде і те, і інше. Дане розширення не обов'язково використовувати, проте більшість веб-розробників надають перевагу JSX, бо це дає можливість працювати з JS в UI частині та бачити цілісну картину компонента.

```
src > components > Team.jsx > ...
1  export const Team = (props) => {
2    return (
3      <div id='team' className='text-center'>
4        <div className='container'>
5          <div className='col-md-8 col-md-offset-2 section-title'>
6            <h2>Meet the Team</h2>
7            <p>
8              Lorem ipsum dolor sit amet, consectetur adipiscing elit dui sed
9              dapibus leonec.
10           </p>
11         </div>
12         <div id='row'>
13           {props.data
14             ? props.data.map((d, i) => (
15                 <div key={` ${d.name}-${i}`} className='col-md-3 col-sm-6 team'>
16                   <div className='thumbnail'>
17                     { ' ' }
18                     <img src={d.img} alt='...' className='team-img' />
19                     <div className='caption'>
20                       <h4>{d.name}</h4>
21                       <p>{d.job}</p>
22                     </div>
23                   </div>
24                 </div>
25             )
26             : 'loading'}
27         </div>
28       </div>
29     </div>
30   )
31 }
```

Рисунок 1.3 – приклад JSX

2. React Native

React Native – це фреймворк на базі бібліотеки React, який був створений для розробки застосунків на наступних платформах: Windows, iOS, Android, AndroidTV, macOS, tvOS, UWP. Основна відмінність принципу роботи React від React Native полягає у тому, що останній не використовує віртуальний DOM. Він працює у фоновому режимі та взаємодіє з нативною, тобто оригінальної платформи за допомогою серіалізації та пакентного моста.

1.3. Безшовна крос-платформа

Компоненти React обгортають наявний рідний код і взаємодіють з рідними API за допомогою декларативної парадигми інтерфейсу React і JavaScript. Це дає змогу розробляти нативні програми для цілих нових команд розробників і може дозволити існуючим нативним командам працювати набагато швидше.

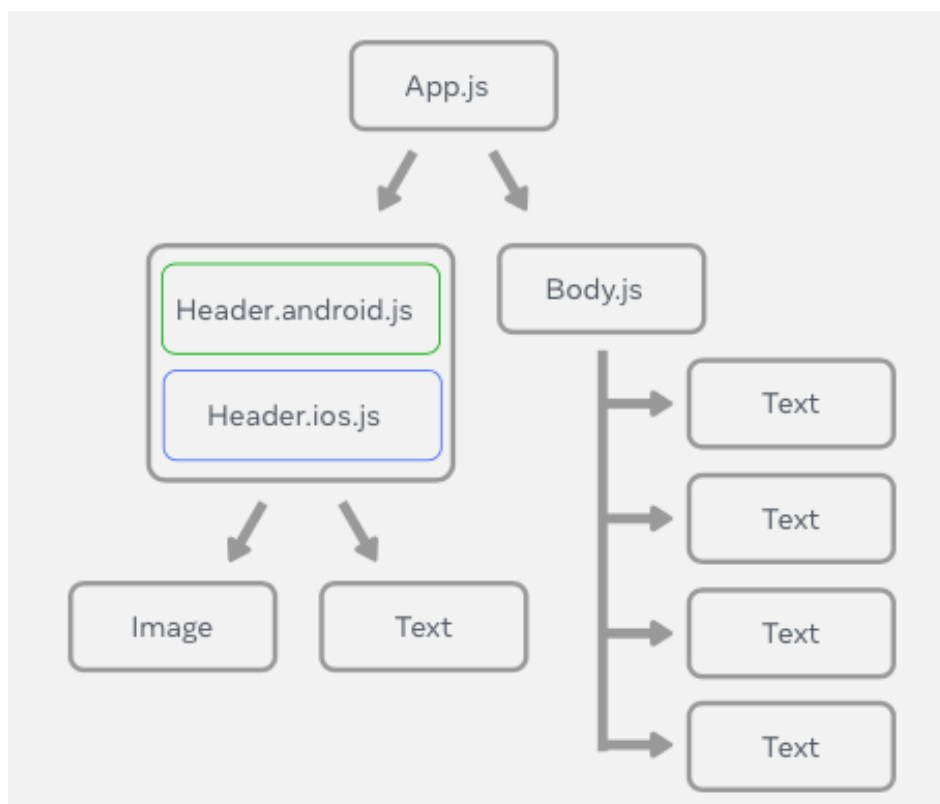


Рисунок 2.1 – Seamless cross-platform

3. Створення веб-застосунку за допомогою бібліотеки React

3.1 Опис застосунку

Для практичної частини курсової роботи був створений застосунок за допомогою бібліотеки React. Додаток є шаблоном головної сторінки компанії, який можна з легкістю переробити під будь-які потреби та під будь-яку сферу діяльності. Для створення застосунку я використовував усі вище наведені переваги бібліотеки React.

Для кожного елемента сторінки я створив окремий компонент, що дало мені змогу легко маніпулювати зовнішнім виглядом кожного компонента. У всіх компонентах використовуються функції, а не класи, що також значно полегшило процес написання коду, оскільки так його значно менше.

```
import { Image } from "../image";

export const Gallery = (props) => {
  return (
    <div id='portfolio' className='text-center'>
      <div className='container'>
        <div className='section-title'>
          <h2>Gallery</h2>
          <p>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit dui sed
            dapibus leonec.
          </p>
        </div>
        <div className='row'>
          <div className='portfolio-items'>
            {props.data
              ? props.data.map((d, i) => (
                  <div key={` ${d.title}-${i}`} className='col-sm-6 col-md-4 col-lg-4'>
                    <Image title={d.title} largeImage={d.largeImage} smallImage={d.smallImage} />
                  </div>
                ))
              : 'Loading...'}
          </div>
        </div>
      </div>
    </div>
  )
}
```

Рисунок 3.1 – Компонент “Gallery”

Також зазначу, що я використовував виключно синтаксис JSX, що дозволило спростити створення компонентів

```

1 import { useState } from 'react'
2 import emailjs from 'emailjs-com'
3
4 const initialState = {
5   name: '',
6   email: '',
7   message: '',
8 }
9 export const Contact = (props) => {
10   const [{ name, email, message }, setState] = useState(initialState)
11

```

Рисунок 3.2 – React Hooks

Окремо хочу продемонструвати використання «хуків», які самі по собі дозволяють підключатись до стану компонента та до методів життєвого циклу.

На рисунку 5 можна побачити приклад мого використання «хука» `useState()`, котрий дозволяє керувати станом таких полів як: “name, email,message”.

Також, у більшості компонентів я використовую “props”, таким чином передаю аргументи всередину дочірнього компонента з головного компонента.

3.2 Інтерфейс застосунку

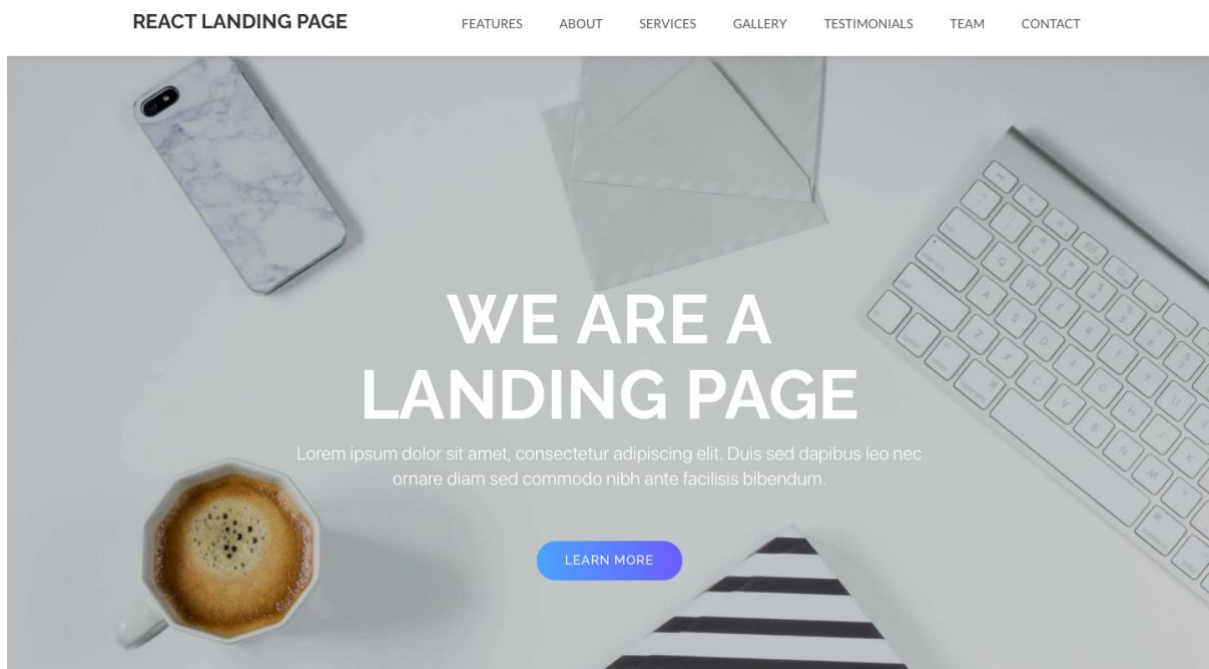


Рисунок 3.3 – головна сторінка

При запуску застосунку, користувач побачить головну сторінку застосунку, на якій зверху є інтерактивне меню для пересування по сторінці та початкова інформація, яку можна при необхідності змінити під будь-які потреби.

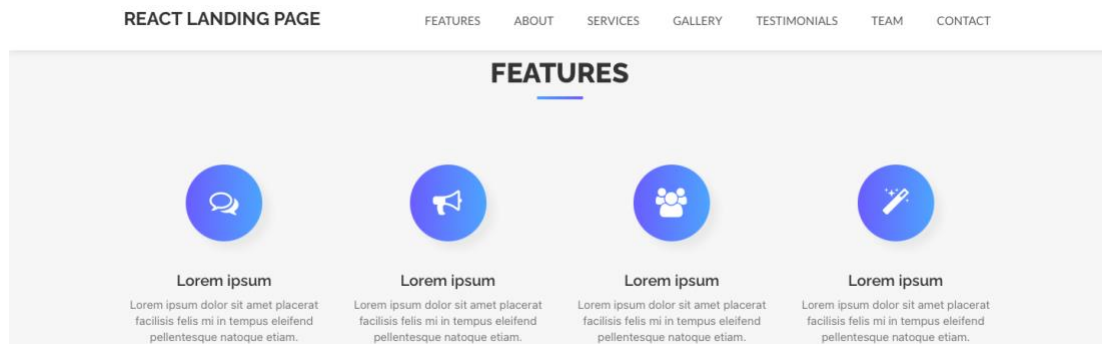


Рисунок 3.4 – Сторінка «Особливості»

На даній сторінці можна побачити перелік шаблон переваг, який легко переробити для будь-яких цілей.

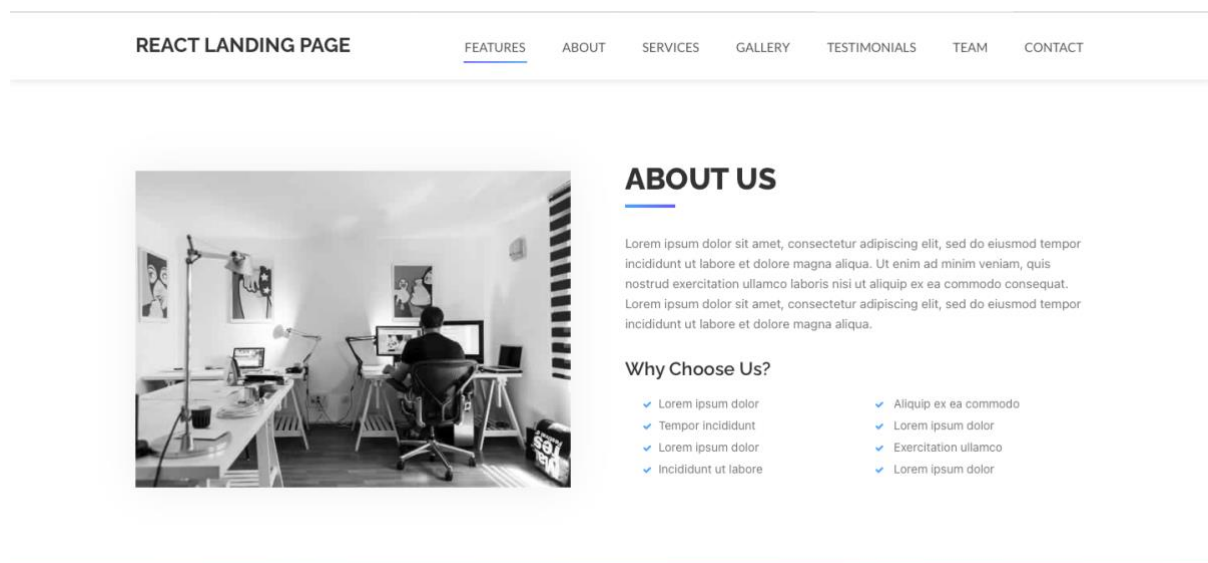


Рисунок 3.5 – Сторінка «Про нас»

Ця сторінка відображує коротку інформацію про компанію.

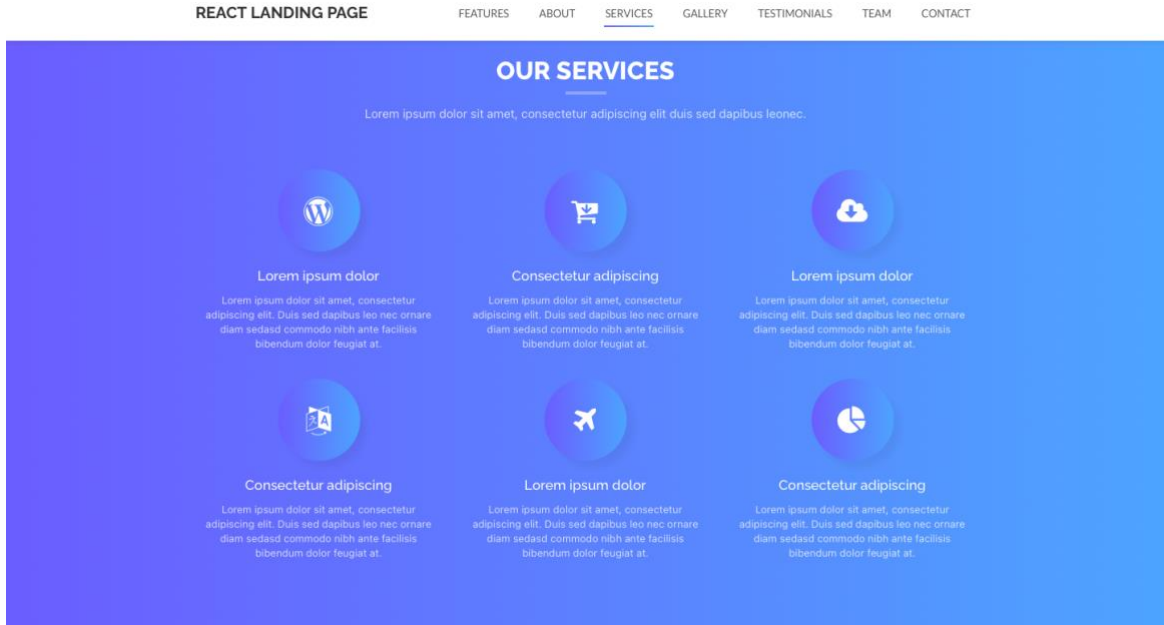


Рисунок 3.6 – сторінка «Наші сервіси»

Ця сторінка слугує для ознайомлення користувача з наявними сервісами, які надаються.

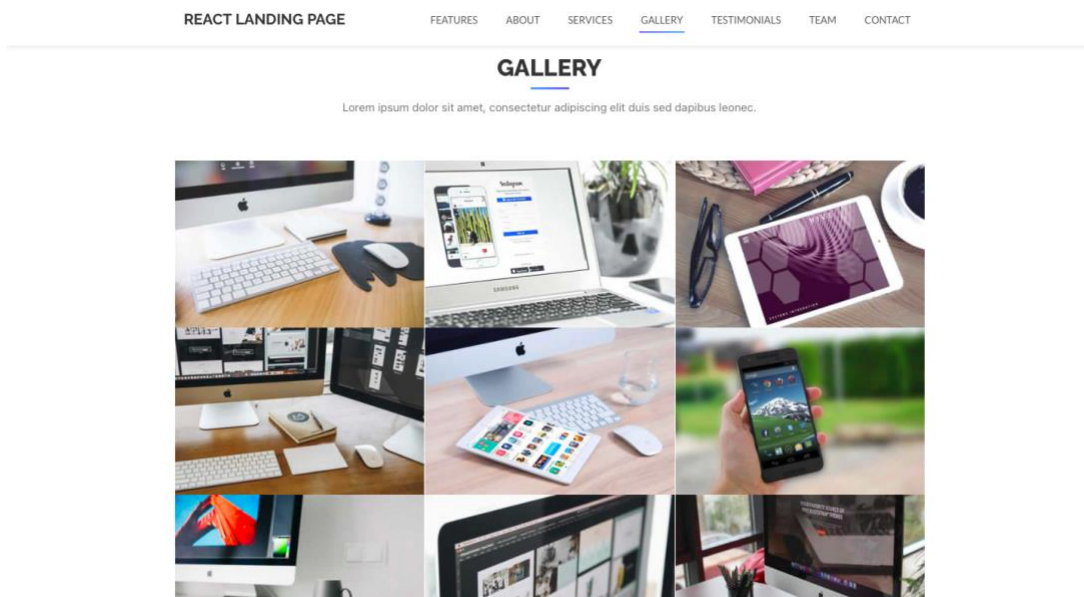


Рисунок 3.7 – сторінка «Галерея»

Ця сторінка може слугувати шаблоном для відображення певних зображень, це можуть бути картинки проектів, фотографії певних об'єктів, тощо.

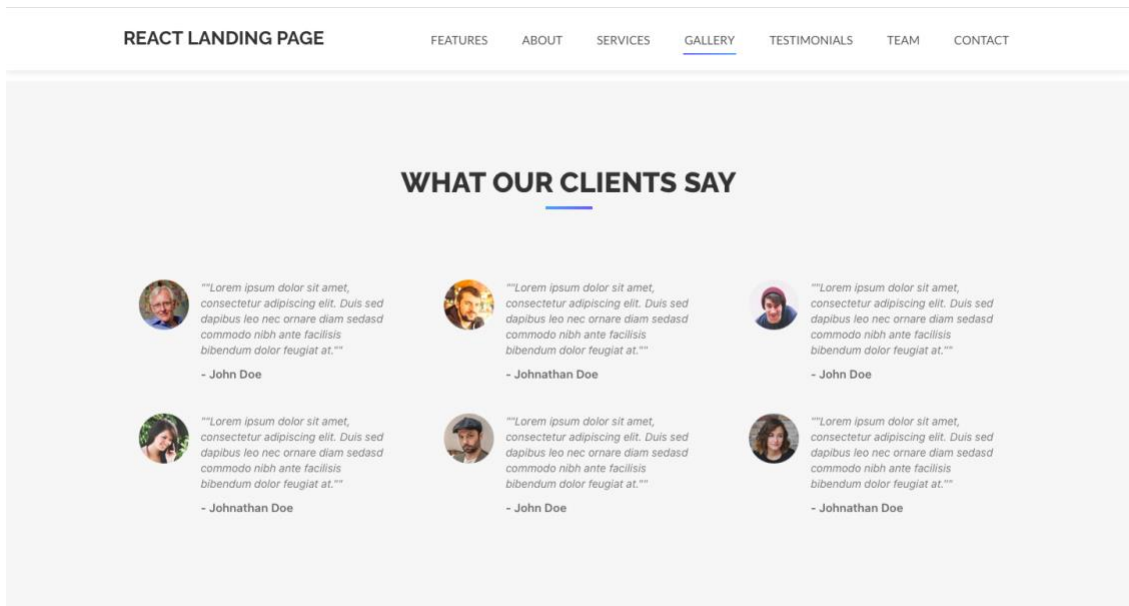


Рисунок 3.8 – сторінка з відгуками клієнтів
Дана сторінка слугує для відображення відгуків клієнтів.

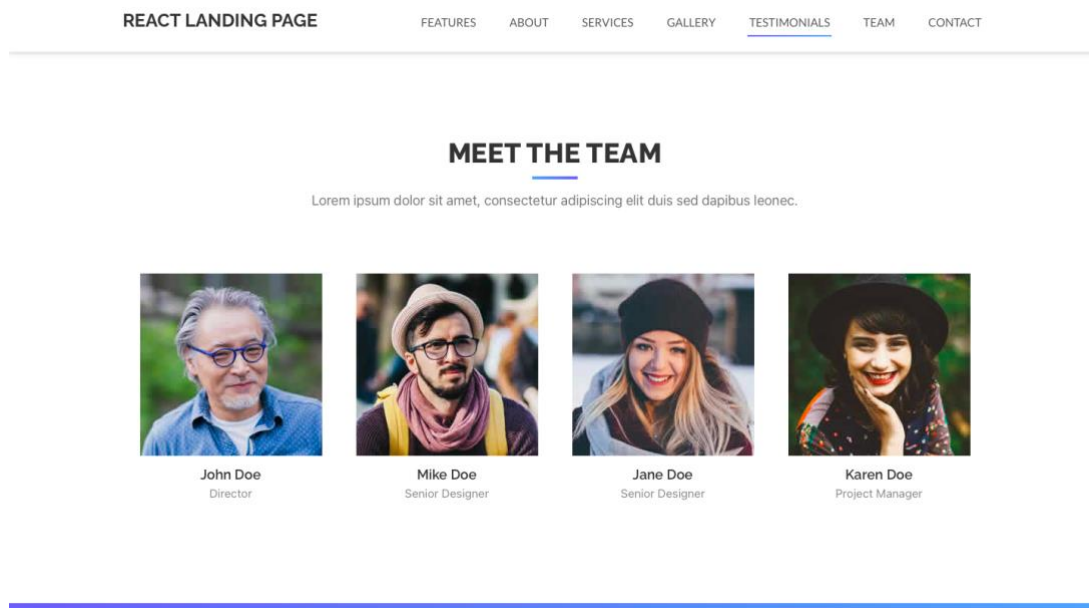


Рисунок 3.9 – сторінка зі працівниками компанії
Цей сектор відображує список наявних працівників з їхніми фото, ім'ям та прізвищем та посадою, яку вони займають.

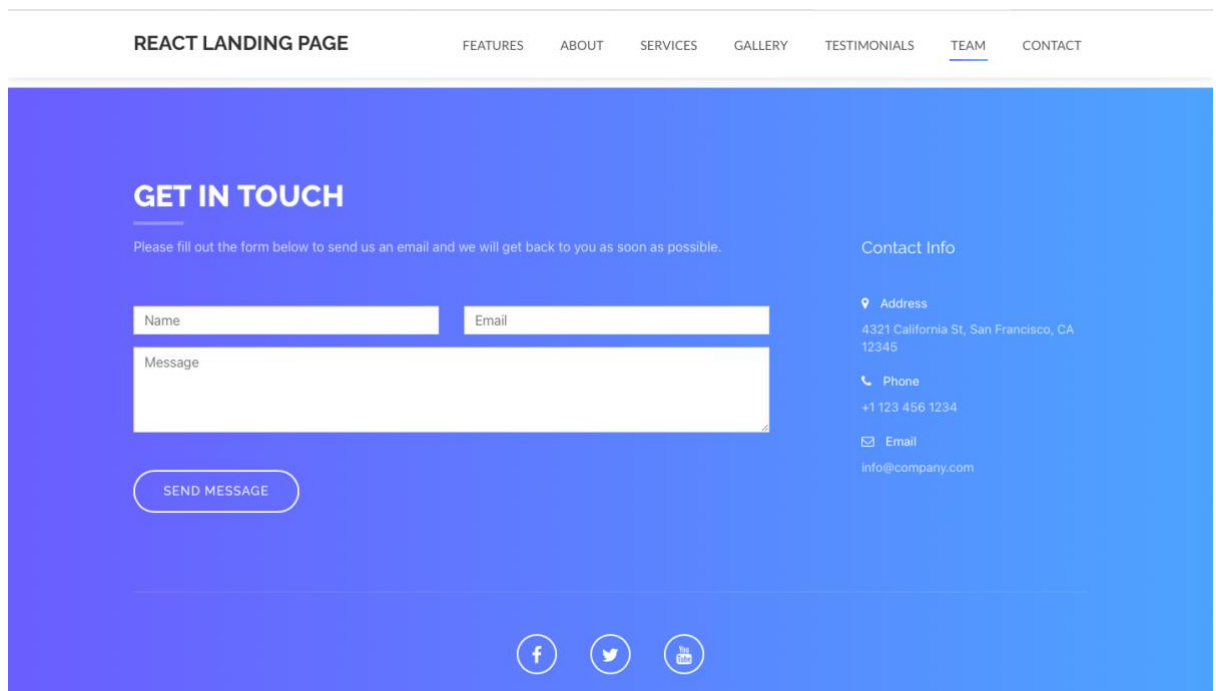


Рисунок 3.10 – сторінка зворотного зв'язку

Ця сторінка слугує для отримання зворотного зв'язку від відвідувачів сайт, де є наступні поля : Ім'я, пошта щоб отримати відповідь, та повідомлення, яке хоче надіслати користувач. Також на сторінці розміщені контактні дані підприємства, її адреса, телефон та електронна адреса для зв'язку.

Висновок

У курсовій роботі був проведений огляд бібліотеки для створення веб-застосунків React та фреймворку для створення крос-платформових застосунків React Native. Бібліотека має низку переваг, такі як використання компонентів, віртуальний DOM та JSX синтаксис, що значно спрощує розробку нових застосунків та редагування вже існуючих.

Для практичної частини курсової роботи був розроблений шаблонний застосунок, який демонструє усі переваги використання React. За допомогою даної бібліотеки розробка застосунку не зайняла надто багато часу та була відносно простою. Створюючи такий застосунок за допомогою PHP чи JS та HTML, я впевнений що це відняло би більше зусиль та часу.

Проаналізувавши бібліотеку React та створивши застосунок за її допомогою, можу зробити висновок, що це одним з найкращих рішень для створення веб-застосунків та крос-платформових додатків.

Список використаних джерел

[1] React – JavaScript-бібліотека для створення користувацьких інтерфейсів
[Електронний ресурс] – Режим доступу до ресурсу:

<https://reactjs.org>

[2] React-Native – фреймворк для створення користувацьких
інтерфейсів[Електронний ресурс] – Режим до ресурсу:

<https://reactnative.dev>

[3] Використання State Hook – React [Електронний ресурс] – Режим
доступу: <https://reactjs.org/docs/hooks-state.html>

[4] React: Virtual DOM [Електронний ресурс] – Режим доступу:

<https://www.codecademy.com/article/react-virtual-dom>

Додатки

```
1  export const About = (props) => {
2    return (
3      <div id="about">
4        <div className="container">
5          <div className="row">
6            <div className="col-xs-12 col-md-6">
7              {" "}
8              {" "}
9            </div>
10           <div className="col-xs-12 col-md-6">
11             <div className="about-text">
12               <h2>About Us</h2>
13               <p>{props.data ? props.data.paragraph : "loading..."}</p>
14               <h3>Why Choose Us?</h3>
15               <div className="list-style">
16                 <div className="col-lg-6 col-sm-6 col-xs-12">
17                   <ul>
18                     {props.data
19                       ? props.data.Why.map((d, i) => (
20                         <li key={` ${d}-${i}`}>{d}</li>
21                       ))
22                       : "loading"}
23                   </ul>
24                 </div>
25                 <div className="col-lg-6 col-sm-6 col-xs-12">
26                   <ul>
27                     {props.data
28                       ? props.data.Why2.map((d, i) => (
29                         <li key={` ${d}-${i}`}> {d}</li>
30                       ))
31                       : "loading"}
32                   </ul>
33                 </div>
34               </div>
35             </div>
36           </div>
37         </div>
38       </div>
39     </div>
40   );
41 };
42
```

КОМПОНЕНТ “About”

```

1 import { useState } from 'react'
2 import emailjs from 'emailjs-com'
3
4 const initialState = {
5   name: '',
6   email: '',
7   message: '',
8 }
9 export const Contact = (props) => {
10   const [{ name, email, message }, setState] = useState(initialState)
11
12   const handleChange = (e) => {
13     const { name, value } = e.target
14     setState((prevState) => ({ ...prevState, [name]: value }))
15   }
16   const clearState = () => setState({ ...initialState })
17
18   const handleSubmit = (e) => {
19     e.preventDefault()
20     console.log(name, email, message)
21     emailjs
22       .sendForm(
23         'YOUR_SERVICE_ID', 'YOUR_TEMPLATE_ID', e.target, 'YOUR_USER_ID'
24       )
25       .then(
26         (result) => {
27           console.log(result.text)
28           clearState()
29         },
30         (error) => {
31           console.log(error.text)
32         }
33       )
34   }
35   return (
36     <div>
37       <div id='contact'>
38         <div className='container'>
39           <div className='col-md-8'>
40             <div className='row'>
41               <div className='section-title'>
42                 <h2>Get In Touch</h2>
43               <p>
44                 Please fill out the form below to send us an email and we will
45                 get back to you as soon as possible.
46               </p>
47             </div>
48             <form name='sentMessage' validate onSubmit={handleSubmit}>
49               <div className='row'>
50                 <div className='col-md-6'>
51                   <div className='form-group'>
52                     <input
53                       type='text'
54                       id='name'
55                       name='name'

```

```

105         {props.data ? props.data.address : 'loading'}
106     </p>
107 </div>
108 <div className='contact-item'>
109     <p>
110         <span>
111             <i className='fa fa-phone'></i> Phone
112         </span>{ ' '}
113         {props.data ? props.data.phone : 'loading'}
114     </p>
115 </div>
116 <div className='contact-item'>
117     <p>
118         <span>
119             <i className='fa fa-envelope-o'></i> Email
120         </span>{ ' '}
121         {props.data ? props.data.email : 'loading'}
122     </p>
123 </div>
124 </div>
125 <div className='col-md-12'>
126     <div className='row'>
127         <div className='social'>
128             <ul>
129                 <li>
130                     <a href={props.data ? props.data.facebook : '/'}>
131                         <i className='fa fa-facebook'></i>
132                     </a>
133                 </li>
134                 <li>
135                     <a href={props.data ? props.data.twitter : '/'}>
136                         <i className='fa fa-twitter'></i>
137                     </a>
138                 </li>
139                 <li>
140                     <a href={props.data ? props.data.youtube : '/'}>
141                         <i className='fa fa-youtube'></i>
142                     </a>
143                 </li>
144             </ul>
145         </div>
146     </div>
147 </div>
148 </div>
149 </div>
150 <div id='footer'>
151     <div className='container text-center'>
152         <p>
153             &copy; 2022 Andrii Vynnyk
154         </p>
155     </div>
156 </div>
157 </div>
158 )
159 }
160

```

Компонент “Contact”

```

1  export const Features = (props) => {
2    return (
3      <div id='features' className='text-center'>
4        <div className='container'>
5          <div className='col-md-10 col-md-offset-1 section-title'>
6            <h2>Features</h2>
7          </div>
8          <div className='row'>
9            {props.data
10             ? props.data.map((d, i) => (
11               <div key={`-${d.title}-${i}`} className='col-xs-6 col-md-3'>
12                 { ' ' }
13                 <i className={d.icon}></i>
14                 <h3>{d.title}</h3>
15                 <p>{d.text}</p>
16               </div>
17             ))
18             : 'Loading...'}
19          </div>
20        </div>
21      </div>
22    )
23  }

```

КОМПОНЕНТ “Features”

```

1  import { Image } from "./image";
2
3  export const Gallery = (props) => {
4    return (
5      <div id='portfolio' className='text-center'>
6        <div className='container'>
7          <div className='section-title'>
8            <h2>Gallery</h2>
9            <p>
10             Lorem ipsum dolor sit amet, consectetur adipiscing elit dui sed
11             dapibus leonec.
12           </p>
13          </div>
14          <div className='row'>
15            <div className='portfolio-items'>
16              {props.data
17               ? props.data.map((d, i) => (
18                 <div key={`-${d.title}-${i}`} className='col-sm-6 col-md-4 col-lg-4'>
19                   <Image title={d.title} largeImage={d.largeImage} smallImage={d.smallImage} />
20                 </div>
21               ))
22               : 'Loading...'}
23            </div>
24          </div>
25        </div>
26      </div>
27    )
28  }
29

```

КОМПОНЕНТ “Gallery”

```

1  export const Header = (props) => {
2    return (
3      <header id='header'>
4        <div className='intro'>
5          <div className='overlay'>
6            <div className='container'>
7              <div className='row'>
8                <div className='col-md-8 col-md-offset-2 intro-text'>
9                  <h1>
10                     {props.data ? props.data.title : 'Loading'}
11                   <span></span>
12                 </h1>
13                 <p>{props.data ? props.data.paragraph : 'Loading'}</p>
14                 <a
15                   href='#features'
16                   className='btn btn-custom btn-lg page-scroll'
17                 >
18                   Learn More
19                 </a>{' '}
20               </div>
21             </div>
22           </div>
23         </div>
24       </div>
25     </header>
26   )
27 }
28

```

Компонент “Header”

```

1  export const Navigation = (props) => {
2    return (
3      <nav id='menu' className='navbar navbar-default navbar-fixed-top'>
4        <div className='container'>
5          <div className='navbar-header'>
6            <button
7              type='button'
8              className='navbar-toggle collapsed'
9              data-toggle='collapse'
10             data-target='#bs-example-navbar-collapse-1'
11            >
12              {' '}
13             <span className='sr-only'>Toggle navigation</span>{' '}
14             <span className='icon-bar'></span>{' '}
15             <span className='icon-bar'></span>{' '}
16             <span className='icon-bar'></span>{' '}
17           </button>
18           <a className='navbar-brand page-scroll' href='#page-top'>
19             React Landing Page
20           </a>{' '}
21         </div>
22
23         <div
24           className='collapse navbar-collapse'
25           id='bs-example-navbar-collapse-1'
26         >
27 > <ul className='nav navbar-nav navbar-right'>...
63 </ul>
64 </div>
65 </div>
66 </nav>
67 )
68 }

```

Компонент “Navigation”

```

export const Services = (props) => {
  return (
    <div id='services' className='text-center'>
      <div className='container'>
        <div className='section-title'>
          <h2>Our Services</h2>
          <p>
            Lorem ipsum dolor sit amet, consectetur adipiscing elit dui sed
            dapibus leonec.
          </p>
        </div>
        <div className='row'>
          {props.data
            ? props.data.map((d, i) => (
                <div key={` ${d.name}-${i}`} className='col-md-4'>
                  { ' ' }
                  <i className={d.icon}></i>
                  <div className='service-desc'>
                    <h3>{d.name}</h3>
                    <p>{d.text}</p>
                  </div>
                </div>
              ))
            : 'loading'}
        </div>
      </div>
    </div>
  )
}

```

КОМПОНЕНТ – “Services”

```

1  export const Team = (props) => {
2    return (
3      <div id='team' className='text-center'>
4        <div className='container'>
5          <div className='col-md-8 col-md-offset-2 section-title'>
6            <h2>Meet the Team</h2>
7            <p>
8              Lorem ipsum dolor sit amet, consectetur adipiscing elit dui sed
9              dapibus leonec.
10           </p>
11         </div>
12         <div id='row'>
13           {props.data
14             ? props.data.map((d, i) => (
15                 <div key={` ${d.name}-${i}`} className='col-md-3 col-sm-6 team'>
16                   <div className='thumbnail'>
17                     { ' ' }
18                     <img src={d.img} alt='...' className='team-img' />
19                     <div className='caption'>
20                       <h4>{d.name}</h4>
21                       <p>{d.job}</p>
22                     </div>
23                   </div>
24                 </div>
25               ))
26             : 'loading'}
27         </div>
28       </div>
29     </div>
30   )
31 }

```

КОМПОНЕНТ “Team”

```

1  export const Testimonials = (props) => {
2    return (
3      <div id='testimonials'>
4        <div className='container'>
5          <div className='section-title text-center'>
6            <h2>What our clients say</h2>
7          </div>
8          <div className='row'>
9            {props.data
10             ? props.data.map((d, i) => (
11               <div key={`-${d.name}-${i}`} className='col-md-4'>
12                 <div className='testimonial'>
13                   <div className='testimonial-image'>
14                     {' '}
15                     <img src={d.img} alt=' />{' '}
16                   </div>
17                   <div className='testimonial-content'>
18                     <p>{d.text}</p>
19                     <div className='testimonial-meta'> - {d.name} </div>
20                   </div>
21                 </div>
22               </div>
23             )
24             : 'loading'}
25          </div>
26        </div>
27      </div>
28    )
29  }
30

```

КОМПОНЕНТ “Tesimonials”

```

export const scroll = new SmoothScroll('a[href*="#"]', {
  speed: 1000,
  speedAsDuration: true,
});

const App = () => {
  const [landingPageData, setLandingPageData] = useState({});
  useEffect(() => {
    setLandingPageData(JsonData);
  }, []);

  return (
    <div>
      <Navigation />
      <Header data={landingPageData.Header} />
      <Features data={landingPageData.Features} />
      <About data={landingPageData.About} />
      <Services data={landingPageData.Services} />
      <Gallery data={landingPageData.Gallery}/>
      <Testimonials data={landingPageData.Testimonials} />
      <Team data={landingPageData.Team} />
      <Contact data={landingPageData.Contact} />
    </div>
  );
};

export default App;

```

КОМПОНЕНТ “App”