

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Звіт з лабораторної роботи № 8
з «Веб-програмування»

Виконав(ла): студент(ка) групи
6.1212 - 1 : Горпинич Валерія
Перевірив: Панасенко Є. В.

Запоріжжя 2024

Виконання завдань лабораторної роботи

1. Використовуючи Create React App створити проект з назвою simple-react-app.

```
PS D:\University5\Web-programming\Repository\WebProgramming\lab8> npx create-react-app simple-react-app

Creating a new React app in D:\University5\Web-programming\Repository\WebProgramming\lab8\simple-react-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1472 packages in 2m

258 packages are looking for funding
  run `npm fund` for details

Installing template dependencies using npm...
added 69 packages, and changed 1 package in 9s

262 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...

removed 1 package, and audited 1541 packages in 6s

262 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Success! Created simple-react-app at D:\University5\Web-programming\Repository\WebProgramming\lab8\simple-react-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

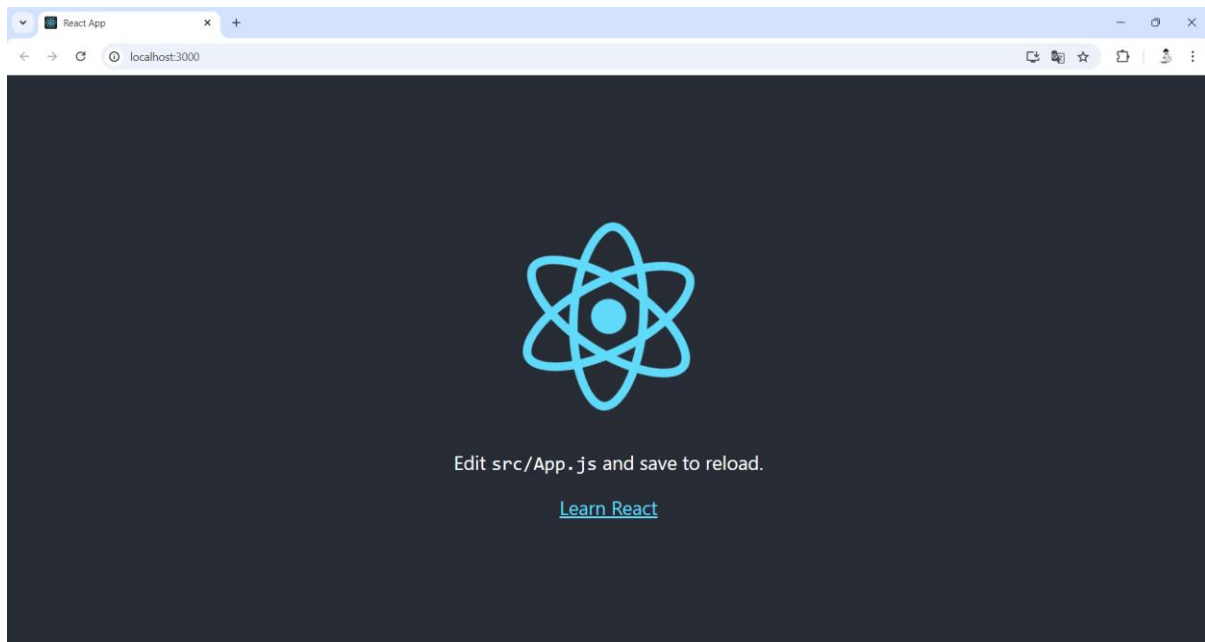
  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd simple-react-app
  npm start

Happy hacking!
PS D:\University5\Web-programming\Repository\WebProgramming\lab8> |
```

npm start



2. Використовуючи наданий шаблон створити набір React компонентів. Окремо повинні бути представлені компоненти (Expenses, ExpenseItem, ExpenseDate, Card).

```
PS D:\University5\Web-programming\Repository\WebProgramming\lab8\simple-react-app> npm install sass
added 8 packages, and audited 1561 packages in 8s

269 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS D:\University5\Web-programming\Repository\WebProgramming\lab8\simple-react-app>
```

```
PS D:\University5\Web-programming\Repository\WebProgramming\lab8\simple-react-app> npm install styled-components
added 12 packages, and audited 1553 packages in 7s

265 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS D:\University5\Web-programming\Repository\WebProgramming\lab8\simple-react-app>
```

ExpenseDate.js:

```
import {styled} from 'styled-components';

const DateStyle = styled.div `
```

```

display: flex;
flex-direction: column;
width: 5.5rem;
height: 5.5rem;
border: 1px solid #ececec;
background-color: #2a2a2a;
color: white;
border-radius: 12px;
align-items: center;
justify-content: center;
`,

const MonthStyle = styled.div `
  font-size: 0.75rem;
  font-weight: bold;
`,

const YearStyle = styled.div `
  font-size: 0.75rem;
`,

const DayStyle = styled.div `
  font-size: 1.5rem;
  font-weight: bold;
`,

export default function ExpenseDate(props) {
  const month = props.date.toLocaleString('en-US', { month: 'long' });
  const day = props.date.toLocaleString('en-US', { day: '2-digit' });
  const year = props.date.getFullYear();

  return (
    <DateStyle>
      <MonthStyle>{month}</MonthStyle>
      <YearStyle>{year}</YearStyle>
      <DayStyle>{day}</DayStyle>
    </DateStyle>
  );
}

```

ExpenseItem.js:

```

import ExpenseDate from './ExpenseDate';
import Card from './Card';
import {styled} from 'styled-components';

const ItemStyle = styled(Card) `
  display: flex;
  justify-content: space-between;

```

```
    align-items: center;
    padding: 0.5rem;
    margin: 1rem 0;
    background-color: #4b4b4b;
  },
  `
const DescriptionStyle = styled.div `
  display: flex;
  flex-direction: row;
  grid-gap: 1rem;
  gap: 1rem;
  align-items: flex-end;
  flex-flow: column-reverse;
  justify-content: flex-start;
  flex: 1 1;

  @media (min-width: 580px) {
    flex-direction: row;
    align-items: center;
    justify-content: flex-start;
    flex: 1 1;
  }
  `

const TitleH2Style = styled.h2 `
  color: #3a3a3a;
  font-size: 1rem;
  flex: 1 1;
  margin: 0 1rem;
  color: white;

  @media (min-width: 580px) {
    font-size: 1.25rem;
  }
  `

const PriceStyle = styled.div `
  font-size: 1rem;
  font-weight: bold;
  color: white;
  background-color: #40005d;
  border: 1px solid white;
  padding: 0.5rem;
  border-radius: 12px;

  @media (min-width: 580px) {
```

```

    font-size: 1.25rem;
    padding: 0.5rem 1.5rem;
  }
},

export default function ExpenseItem(props) {
  return (
    <ItemStyle>
      <ExpenseDate date={props.date} />
      <DescriptionStyle>
        <TitleH2Style>{props.title}</TitleH2Style>
        <PriceStyle>${props.amount}</PriceStyle>
      </DescriptionStyle>
    </ItemStyle>
  );
}

```

Expenses.js:

```

import Card from './Card';
import ExpenseItem from './ExpenseItem';
import {styled} from 'styled-components';

const ExpenseStyle = styled(Card) `
  padding: 1rem;
  background-color: rgba(117, 118, 118, 0.46);
  margin: 2rem auto;
  width: 50rem;
  max-width: 95%;
`;

export default function Expenses(props) {
  return (
    <ExpenseStyle>
      {props.items.map((expense) => (
        <ExpenseItem
          key={expense.id}
          title={expense.title}
          amount={expense.amount}
          date={expense.date}
        />
      ))}
    </ExpenseStyle>
  );
}

```

Card.js:

```

import {styled} from 'styled-components';

```

```
const CardStyle = styled.div `
  border-radius: 12px;
  box-shadow: 0 1px 8px rgba(0, 0, 0, 0.25);
`;

export default function Card(props) {
  return <CardStyle
className={props.className}>{props.children}</CardStyle>;
}
```

App.js:

```
import React from 'react';
import Expenses from './components/Expenses';
import './scss/styles.scss';
import styled from 'styled-components';

const expenses = [
  { id: 'e1', title: 'Toilet Paper', amount: 94.12, date: new Date(2024, 7, 14) },
  { id: 'e2', title: 'New TV', amount: 799.49, date: new Date(2024, 2, 12) },
  { id: 'e3', title: 'Car Insurance', amount: 294.67, date: new Date(2024, 2, 28) },
  { id: 'e4', title: 'New Desk (Wooden)', amount: 450, date: new Date(2024, 5, 12) },
];

const AppStyle = styled.body `
  margin: 0;
  background-color: #ffffff;
  font-family: 'Noto Sans JP', sans-serif;
`;

export default function App() {
  return (
    <AppStyle>
      <h2>My Expenses tempalte</h2>
      <Expenses items={expenses} />
    </AppStyle>
  );
}
```

styles.scss:

```
@import
url('https://fonts.googleapis.com/css2?family=Noto+Sans+JP:wght@400;700&display=swap');
```

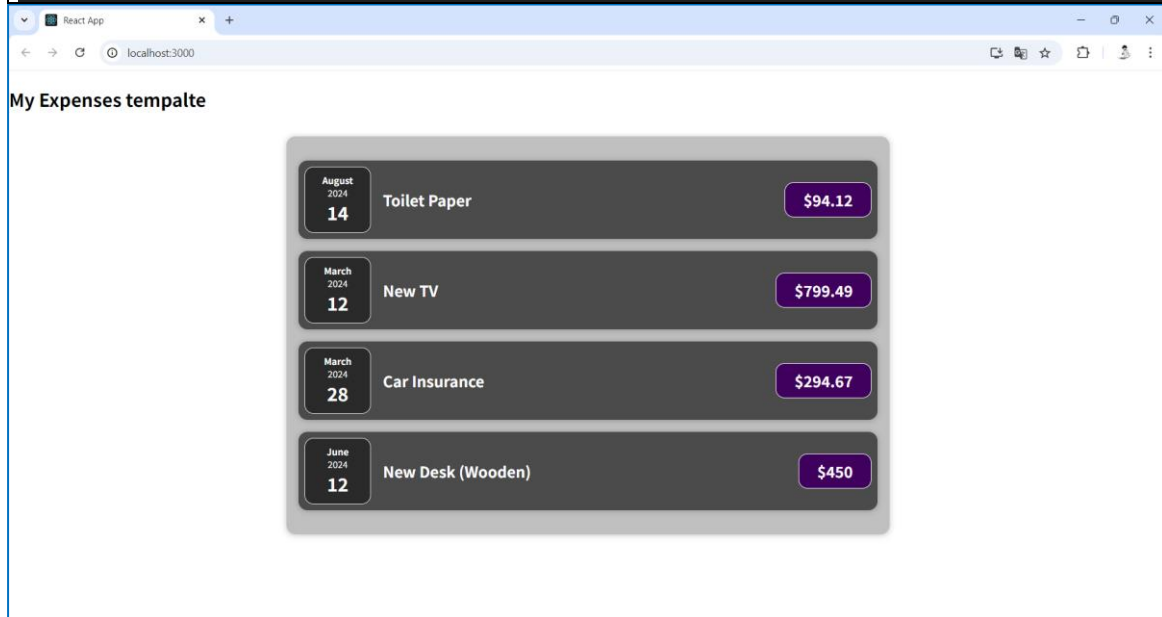
```

$font-family: 'Noto Sans JP', sans-serif;

* {
  box-sizing: border-box;
}

html {
  font-family: $font-family;
}

```

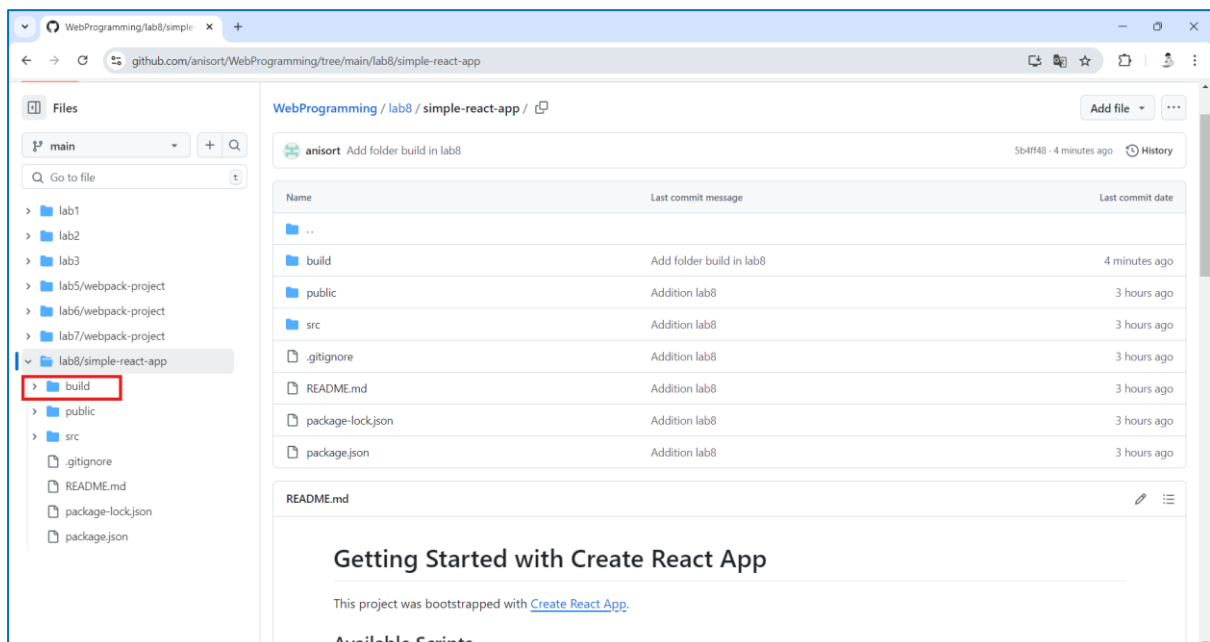


3. Дані для відображення повинні подаватися у якості статичного масива компоненту Expenses у головному компоненті App. Структура та приклад даних наведено у вкладеному файлі.


```
JS App.js  X  JS Card.js  JS Expenses.js  JS ExpenseItem.js  JS ExpenseDate.js  styles.scss

simple-react-app > src > JS App.js > AppStyle
1  import React from 'react';
2  import Expenses from './components/Expenses';
3  import './scss/styles.scss';
4  import styled from 'styled-components';
5
6  const expenses = [
7    { id: 'e1', title: 'Toilet Paper', amount: 94.12, date: new Date(2024, 7, 14) },
8    { id: 'e2', title: 'New TV', amount: 799.49, date: new Date(2024, 2, 12) },
9    { id: 'e3', title: 'Car Insurance', amount: 294.67, date: new Date(2024, 2, 28) },
10   { id: 'e4', title: 'New Desk (Wooden)', amount: 450, date: new Date(2024, 5, 12) },
11 ];
12
13 const AppStyle = styled.body `
14   margin: 0;
15   background-color: #ffffff;
16   font-family: 'Noto Sans JP', sans-serif;
17 `
18
19 export default function App() {
20   return (
21     <AppStyle>
22       <h2>My Expenses template</h2>
23       <Expenses items={expenses} />
24     </AppStyle>
25   );
26 }
27
```

4. Розмістити додаток включаючи директорію dist у каталозі lab-<номер поточної лабораторної роботи>



5. Оновити посилання на робочий проект на персональній сторінці.

<https://github.com/anisort/WebProgramming.git>

Питання

1. Поясніть призначення бібліотеки react-dom.

react-dom — це офіційна бібліотека, яка є частиною екосистеми React. Її основне призначення — керувати відображенням компонентів React в реальному DOM (Document Object Model). Вона виступає як міст між компонентами React і браузерним DOM.

2. Поясніть призначення бібліотеки babel.

Babel — це транспайлер (компілятор), який перетворює сучасний код JavaScript (ES6+, JSX, TypeScript тощо) у код, зрозумілий більшості браузерів. Браузери можуть не підтримувати всі нові можливості JavaScript, тому Babel перетворює новий синтаксис у старіший, сумісний з браузерами.

3. Наведіть приклади інструментів які є альтернативою Create React App

Create React App (CRA) — це інструмент для швидкого старту проектів React, який пропонує готову конфігурацію без необхідності вручну налаштовувати Webpack, Babel та інші інструменти. Проте існують інші альтернативні інструменти, які можна використовувати для створення і налаштування React-проектів:

1. Next.js

- Це фреймворк на базі React, який підтримує рендеринг на сервері (SSR), генерацію статичних сайтів (SSG) та інші оптимізації для SEO.

2. Vite

- Це новий інструмент для фронтенд-розробки, який пропонує дуже швидку збірку і час запуску завдяки використанню ES-модулів (native ES modules).

3. Parcel

- Це альтернативний бандлер, який автоматично настраює проект без явного конфігурування (подібно до CRA, але більш легковажний і швидкий).

4. Razzle

- Це інструмент, схожий на Create React App, але з можливістю серверного рендерингу з мінімумом налаштувань.