1. Устанавливаем WebStorm

2. Устанавливаем Node.js

3. Из командной строки (Node.js command prompt):

- установить npm i create-react-app -g

- перейти в исходную папку, куда будет устанавливаться проект (D:)

- запускаем create-react-app diplom\_react

4. Запускам WebStorm -> Открыть -> 'D:\diplom\_react'

5. Первоначальная настройка проекта:

- тестирование проекта какое либо на данном этапе не предполагается, удалим строки из package.json:

"@testing-library/jest-dom": "^4.2.4",

"@testing-library/react": "^9.5.0",

"@testing-library/user-event": "^7.2.1",

"test": "react-scripts test",

- нам не нужно ручное настраивание и ручной контроль всех зависимостей (Webpack, Babel, ESLint), удалим строки из package.json:

"eject": "react-scripts eject"

- настроим webpack чтобы мы могли подключать стили как объекты

В файлах

D:\Refer\_React\node\_modules\react-scripts\config\webpack.config.js

Изменим строку:

options: cssOptions,

на

options: {...cssOptions, modules: **true**},

6. Удалим ненужные файлы:

- \src\App.test.js

- \src\logo.svg

- \src\setupTests.js

- \src\serviceWorker.js

- \public\logo192.png

- \public\logo512.png

- \public\manifest.json

- \public\robots.txt

7. Уберем все ненужное в следующих файлах:

- \public\index.html, сразу переименуем заголовок

<!DOCTYPE html>  
<html lang="en">  
 <head>  
 <meta charset="utf-8" />  
 <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />  
 <meta name="viewport" content="width=device-width, initial-scale=1" />  
 <meta name="theme-color" content="#000000" />  
 <title>Курсовая работа по React</title>  
 </head>  
 <body>  
 <noscript>You need to enable JavaScript to run this app.</noscript>  
 <div id="root"></div>  
 </body>  
</html>

- \src\App.css

.App {  
 text-align: center;  
}

- \src\App.js

* **import** React **from** 'react';  
  **import** './App.css';  
    
  **function** App() {  
   **return** (  
   <div className="App">  
   Hello World!!!  
   </div>  
   );  
  }  
    
  **export default** App;

- \src\index.js

**import** React **from** 'react';  
**import** ReactDOM **from** 'react-dom';  
**import** './index.css';  
**import** App **from** './App';  
  
ReactDOM.render(  
 <React.StrictMode>  
 <App />  
 </React.StrictMode>,  
 document.getElementById('root')  
);

Мы настроили проект, удалили из него все лишнее и теперь готовы к разработке. По-хорошему, нам нужно еще настроить Git. Но это не входит в программу курсовой.

8. Установим в наш проект Material-UI и SVG иконки, Axios

- npm i @material-ui/core

- npm i @material-ui/icons

- npm i axios –S

- npm install firebase –save

- npm install redux react-redux –save

- npm install redux-thunk --save

9. Структура проекта:

- Layout. Здесь расположены файлы, которые задают общую разметку всему сайту. Содержат хедер, футер, майн теги, а также стили для них

- Containers. Здесь расположены файлы, которые содержат собственный стейт, т.е. корневые компоненты

- Components. Здесь будут располагаться чисто функциональные компоненты

10. По структуре, у нас будет 5 контейнеров и один лайоут. Создадим их:

В папке Layout:

* D:\Refer\_React\src\Layout\Layout.js

**import** React, {Component} **from** 'react'  
  
**class** Layout **extends** Component {  
 render() {  
 **return** (  
 <div>  
 <main>  
 { **this**.props.children }  
 </main>  
 </div>  
 )  
 }  
}  
  
**export default** Layout

* D:\Refer\_React\src\Layout\Layout.css

.Layout {  
 height: 100vh;  
 display: flex;  
 flex-direction: column;  
}  
  
.Layout main {  
 display: flex;  
 flex-direction: column;  
 flex-grow: 1;  
}

В папке Containers:

D:\Refer\_React\src\Containers

D:\Refer\_React\src\Containers\Account

D:\Refer\_React\src\Containers\Account\Account.css

D:\Refer\_React\src\Containers\Account\Account.js

D:\Refer\_React\src\Containers\Log

D:\Refer\_React\src\Containers\Log\Log.css

D:\Refer\_React\src\Containers\Log\Log.js

D:\Refer\_React\src\Containers\Main

D:\Refer\_React\src\Containers\Main\Main.css

D:\Refer\_React\src\Containers\Main\Main.js

D:\Refer\_React\src\Containers\Register

D:\Refer\_React\src\Containers\Register\Register.css

D:\Refer\_React\src\Containers\Register\Register.js

D:\Refer\_React\src\Containers\Test

D:\Refer\_React\src\Containers\Test\Test.css

D:\Refer\_React\src\Containers\Test\Test.js

**import** React, {Component} **from** 'react'  
**import** classes **from** './Main.css'  
  
**class** Main **extends** Component {  
 render() {  
 **return** (  
 <div className={classes.Main}>  
 <h1>Main</h1>  
 </div>  
 )  
 }  
}  
  
**export default** Main

Main.css:

.Main {  
 color: aqua;  
}

Остальные файлы по аналогии.

11.Добавим меню к лояуту

12.прикрутим к проекту react-router

13. Прикрутим к проекту redux

14. Добавим в стор тестовые данные для страницы со входом

15. Сверстаем страницу

16. Добавим в стор тестовые данные для страницы с регистрацией

17. Сверстаем страницу

18. Добавим в стор тестовые данные для страницы с прохождением теста

19. Сверстаем страницу главную и сверстаем страницу с прохождением теста

19.1 Сверстать страницу с результатом прохождения теста

19.2 Продумать над тем, что бы страница с результатами была единой для всех (показать пользователю результат прохождения теста и показать преподу результат прохождения его тестов всеми студентами)

20. Добавим в стор тестовые данные для страницы личным кабинетом

21. Сделаем мультикомпонент с таблицей

22. сверстаем страницу

В качестве бэкенд сервера будем использовать систему Firebase от компании Google. Для того, чтобы создать проект в fb необходимо создать аккаунт в gmail. Заходим на сайт

[firebase.google.com авторизуемся под gmail-аккаунтом и переходим в консоль (верхний правый угол). Жмем на кнопку создать проект. Вводим имя (react-dip). От дополнительных опций (Google Аналитика) отказываемся. Жмем на кнопку Создать проект. После создания убедимся, что был создан проект с бесплатным тарифом Spark.](https://firebase.google.com/" \t "_blank)

Слева, на вкладке Develop выбираем Real Database. Далее жмем на кнопку Создать базу данных.

Выбираем тестовый режим

Переносим конфигурацию в проект

Устанавливаем режим аутентификации

Устанавливаем правила для фильтрации, иначе будет сильно ругаться

Структура базы данных

- ключ: {

"access" : 2, // тип доступа

"key" : "-MK7mB5W2y0YnTZW7N2G", // ключ теста

"name" : "Компьютерная сеть. Топология сети", // наименование теста

"uid" : "H2FfnJfpDKfT2THljCGtNeCbyh73" // идентификатор пользователя, кому принадлежит тест

}

- quises: {

"name" : "Сколько байт в бите?", // Наименование вопроса

"keyQuiz" : "-MK7mB5W2y0YnTZW7N2G", // ключ теста

"key" : "[-MK8DRpzHMXsLiE4fHK3](https://console.firebase.google.com/project/react-dip/database/react-dip/data/-MK8DRpzHMXsLiE4fHK3)", // ключ вопроса

"correctAnswer" : "[-MK8DRpzHMXsLiE4fHK3](https://console.firebase.google.com/project/react-dip/database/react-dip/data/-MK8DRpzHMXsLiE4fHK3)", // правильный ответ

- "answers" : {

"key" : "[-MK8DRpzHMXsLiE4fHK3](https://console.firebase.google.com/project/react-dip/database/react-dip/data/-MK8DRpzHMXsLiE4fHK3)",

"name" : "8 бит",

}

}