**Лабораторная работа**

**«Создание клиент-серверного приложения на Flask и React»**

**Порядок выполнения лабораторной работы**

1. Создать проект, например, с помощью Create React App.

npx create-react-app my-app

2. Инсталлировать Flask:

pip install flask

3. Создать в директории проекта (my-app) папку backend.

4. Скопировать в папку backend flask.exe из директории:

C:\Users\<имя\_пользователя>\AppData\Local\Programs\Python\Python310\Scripts

5. Перенести в папку backend файл БД (**games.sqlite**).

6. Создать в директории backend файл приложения (**app.py**):

from flask import Flask

app = Flask(\_\_name\_\_)

import controllers.data

7. Создать в директории backend файл **utils.py**:

import sqlite3

def get\_db\_connection():

return sqlite3.connect('games.sqlite')

8. Создать в директории backend/controllers файл **data.py**:

from app import app

from flask import request, session

from flask import jsonify

from utils import get\_db\_connection

from models.data\_model import \*

@app.route('/data', methods=['get'])

def index():

conn = get\_db\_connection()

data = get\_games(conn)

arr\_data = data.to\_json(orient='records')

return jsonify(arr\_data)

9. Создать в директории backend/models файл **data\_model.py**:

import pandas

def get\_games(conn):

return pandas.read\_sql(

'''

SELECT \* FROM games

''', conn)

10. Создать в директории backend файл для запуска сервера Flask (**.flaskenv**):

FLASK\_APP=app

FLASK\_ENV=development

11. Запустить сервер Flask из директории backend:

flask run

Если зайти по адресу localhost:5000/data, то будет отображена страница с JSON, содержащем данные из БД.

12. Реализуем простую страницу для отображения всех данных из БД в одной таблице. Для этого в директории src создадим папку components и создадим в ней файл Table.js:

**import** React from 'react';

**function** TableRow(props) {

**const** cells = Object.values(props.row).map((item, index) => item === undefined ? <></> : <td key={index}> {item} </td>);

**return** (

<tr>

{cells}

</tr>

);

}

**export function** Table(props) {

**const** [tableData, setTableData] = React.useState(props.dataset);

**const** updateTableData = (value) => {

setTableData(value);

};

**const** rows = tableData.map((item, index) => <TableRow key={index} row={item}/>);

**return** (

<>

<table className="table-data">

<thead>

<TableRow row={Object.keys(props.dataset[0])} />

</thead>

<tbody>

{rows}

</tbody>

</table>

</>

);

}

13. Создадим компонент для загрузки данных с сервера Flask. Для этого в директории src/components создадим файл DataLoader.js:

**import** React, { useEffect, useState } from 'react';

**import** {Table} from "./Table";

**export function** DataLoader() {

**const** [dataset, setDataset] = useState(0);

**const** [isLoading, setLoading] = useState(**true**);

**const** [isError, setError] = useState(**false**);

useEffect(() => {

fetch('/data').then((response) => {

**if** (response.ok) {

**return** response.json();

}

})

.then(data => {

setDataset(JSON.parse(data));

setLoading(**false**);

})

.catch((error) => {

setError(**true**);

});

}, []);

**if** (isError) {

**return** <div className="App">Internal server error</div>;

}

**else if** (isLoading) {

**return** <div className="App">Загрузка...</div>;

}

**return** (

<>

<h3>Рейтинг компьютерных игр</h3>

<Table dataset={dataset}/>

</>

);

}

14. Изменить файл index.js (в директории src) следующим образом:

**import** React from 'react';

**import** ReactDOM from 'react-dom/client';

**import** App from './App';

**import** './index.css';

**import** {DataLoader} from './components/DataLoader';

**const** root =

ReactDOM.createRoot(**document**.getElementById('root'));

root.render(

<React.StrictMode>

<App />

</React.StrictMode>

);

root.render(

<>

<DataLoader />

</>

);

15. Добавить в package.json прокси на сервер Flask:

…

"private": true,

**"proxy": "http://localhost:5000/",**

"dependencies":

…

16. Убедиться, что сервер на Flask работает, и запустить проект на React из корневой директории (**my-app**):

npm start

В браузере откроется страница с таблицей, заполненной данными из БД, полученными с сервера Flask. Если сервер на Flask не включен, то на странице будет отображаться сообщение “Internal server error”.