

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування та спеціалізованих комп’ютерних систем**

**Лабораторна робота №3**

з дисципліни **Web-дизайн**

*на тему: “Розробка Web-додатка засобами Javascript/React”*

Виконала: студентка ІІI курсу

ФПМ групи КВ-11

Зубенко Марія Олексіївна

Перевірив:

Київ – 2024

*Мета роботи*: ознайомитись із засобами фреймворка React та навчитись створювати асинхронні запити до Web-сервера.

*Загальне завдання:* адаптувати програмний код ЛР№2 до вимог фреймворка React та забезпечити завантаження необхідних даних з Web-сервера.

Інструменти розробки: Javascript, React, NodeJS, Express.

*Посилання на Github-репозиторій*: https://github.com/MariiaZubenko/WEB\_Zubenko

*Контакт студента у месенджері Телеграм:* @one\_of\_vips

**Опис додатка «Облік робочого часу»**

Додаток передбачає 5 сторінок:

1. Реєстрація користувача (поля: ім’я, email, пароль, дата народження);

2. Вхід до сайту (поля: email, пароль);

3. Профіль користувача (поля у табличному вигляді);

4. Про додаток (короткий опис додатка);

5. Робоча сторінка додатка на якій знаходяться кнопки для запуску, паузи і зупинки таймера, а також циферблат;

**Список React-компонентів та схема їх взаємодії**

***App:*** головний компонент додатку, який містить в собі всі інші компоненти.

***Header:*** компонент верхньої частини сторінки (хедеру) для навігації по додатку.

***MainPage:*** компонент, який відображає головну сторінку із таймером та кнопками для управління ним. Наприклад, можна запускати таймер, ставити його на паузу та зупиняти. Усі таймери користувача та їх загальний час також можна бачити на сторінці.

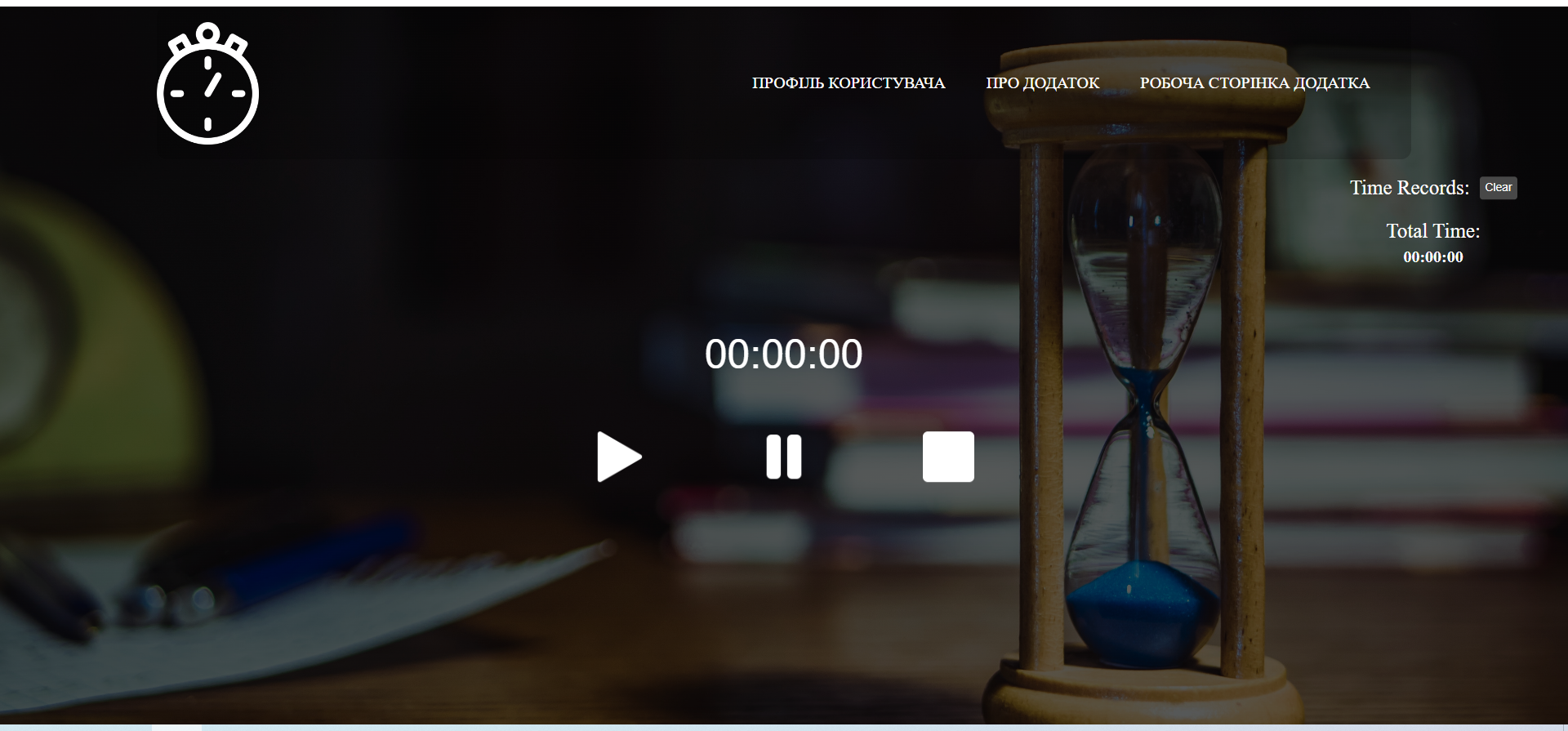
***AboutPage:*** компонент, який відображає сторінку із описом додатку.

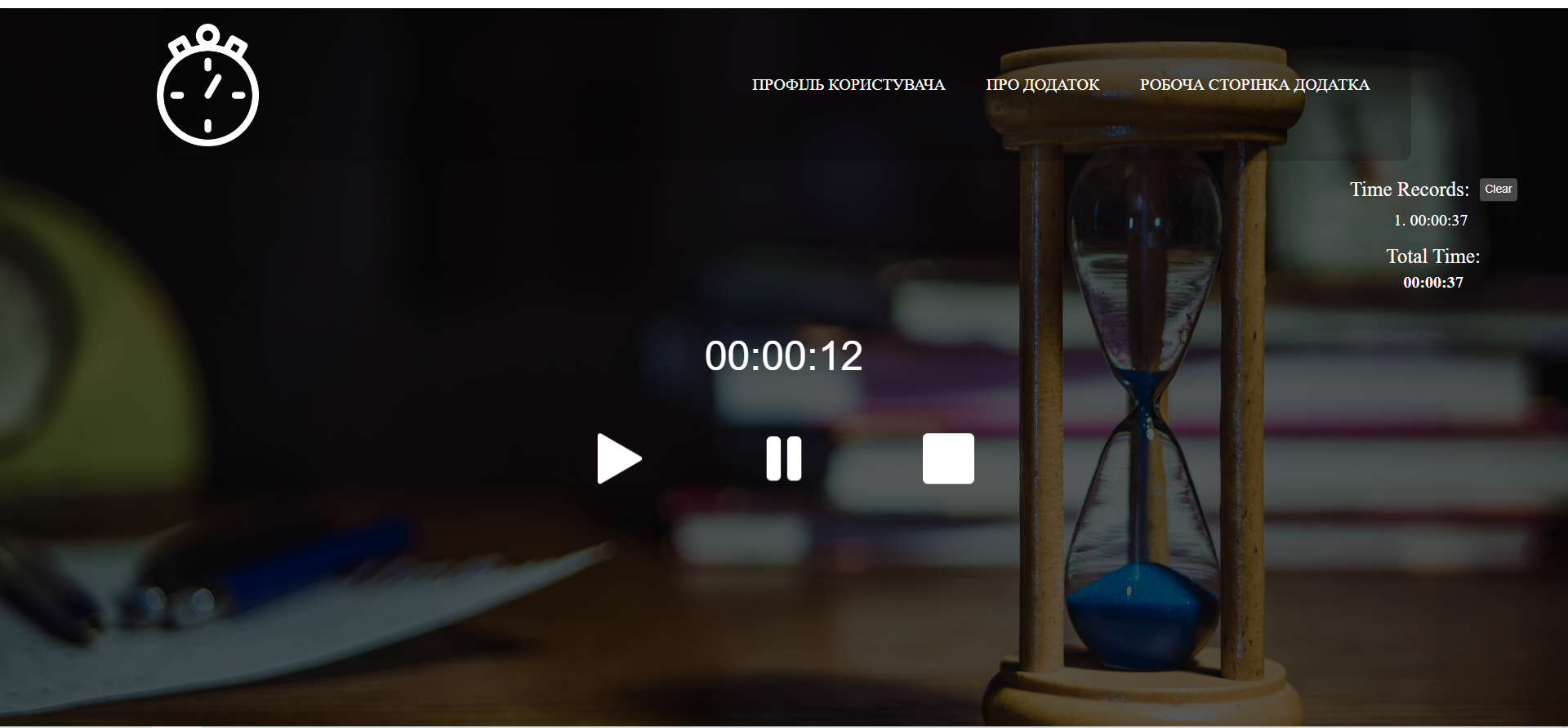
***RegisterPage:*** компонент, що відображає сторінку реєстрації. Реєстрація відбувається за іменем, поштою, паролем та датою народження.

***LoginPage:*** компонент, що відображає сторінку логіну. Логін відбувається за поштою та паролем.

***ProfilePage:*** компонент, що відображає сторінку з інформацією про користувача. Тут показано його ім’я, пошту на загальний час роботи.

**Копії екрана робочої сторінки**







**Код компоненту робочої сторінки**

**import** React, { useState, useEffect, useRef } from "react";

**import** axios from "axios";

**import** play from "../../assets/img/play.png";

**import** pause from "../../assets/img/pause.png";

**import** stop from "../../assets/img/stop.png";

**import** "./style.css";

**const** MainPage = () => {

**const** [time, **setTime**] = useState(0);

**const** [running, setRunning] = useState(false);

**const** [timeRecords, setTimeRecords] = useState([]);

**const** timerRef = useRef(null);

useEffect(() => {

fetchTimeRecords();

}, []);

**const** fetchTimeRecords = async () => {

**const** token = localStorage.**getItem**("token");

**const** response = await axios.get("http://localhost:8080/api/timerecords", {

headers: { Authorization: `Bearer ${token}` },

});

setTimeRecords(response.data);

};

useEffect(() => {

**if** (running) {

timerRef.current = **setInterval**(() => {

**setTime**((prevTime) => prevTime + 1);

}, 1000);

} **else** **if** (!running && time !== 0) {

**clearInterval**(timerRef.current);

}

**return** () => **clearInterval**(timerRef.current);

}, [running]);

**const** formatTime = (seconds) => {

**const** **getSeconds** = `0${seconds % 60}`.**slice**(-2);

**const** minutes = `${Math.**floor**(seconds / 60)}`;

**const** **getMinutes** = `0${minutes % 60}`.**slice**(-2);

**const** **getHours** = `0${Math.**floor**(seconds / 3600)}`.**slice**(-2);

**return** `${**getHours**}:${**getMinutes**}:${**getSeconds**}`;

};

**const** handleStart = () => {

setRunning(true);

};

**const** handlePause = () => {

setRunning(false);

};

**const** handleStop = async () => {

setRunning(false);

**const** token = localStorage.**getItem**("token");

**const** response = await axios.post(

"http://localhost:8080/api/timerecords",

{ time },

{ headers: { Authorization: `Bearer ${token}` } }

);

setTimeRecords([...timeRecords, response.data]);

**setTime**(0);

};

**const** handleClear = async () => {

**const** token = localStorage.**getItem**("token");

await axios.**delete**("http://localhost:8080/api/timerecords", {

headers: { Authorization: `Bearer ${token}` },

});

setTimeRecords([]);

};

**const** getTotalTime = () => {

**return** timeRecords.**reduce**((total, record) => total + record.time, 0);

};

**return** (

<div className="content">

<div className="">

<div className="timer\_counter">

<input

className="timer"

id="timer"

type="text"

value={formatTime(time)}

readOnly

/>

</div>

<div className="icons">

<button id="start" onClick={handleStart}>

<img src={play} className="icon" />

</button>

<button id="pause" onClick={handlePause}>

<img src={pause} className="icon" />

</button>

<button id="stop" onClick={handleStop}>

<img src={stop} className="icon" />

</button>

</div>

</div>

<div className="time-records">

<div className="time-records\_\_title">

Time Records:{" "}

<button className="clear-button" onClick={handleClear}>

Clear

</button>

</div>

<div className="time-records\_\_list">

{timeRecords.**map**((record, index) => (

<div className="time-record" key={index}>

{index + 1}. {formatTime(record.time)}

</div>

))}

</div>

<div className="total-time">

<div className="total-time\_\_title">Total Time:</div>

<span>{formatTime(getTotalTime())}</span>

</div>

</div>

</div>

);

};

**export** **default** MainPage;