НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

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

**систем**

# Лабораторна робота №2

з дисципліни

**«ВЕБ дизайн»**

Тема: «Розробка функціональності Web-додатка мовою Javascript»

Виконав: студент III курсу

ФПМ групи КВ-81

Гей Антон

Перевірив: Петрашенко А. В.

Київ – 2021

## Завдання на роботу

**Мета**: ознайомитись із засобами мов HTML5 та CSS3 та навчитись застосовувати бібліотеки Twitter Bootstrap/Material UI щодо швидкої побудови Web-інтерфейсу користувача.

**Загальне завдання**: розробити функціональність для статичних сторінок Web-додатку першої лабораторної роботи із використанням шаблону MVC.

Github репозиторій:   
https://github.com/amdota/uni-web-labs/tree/main/lab2

Telegram: @tryplaybetter

Статичний Web-сайт має містити наступні сторінки:

* реєстрація користувача (поля: прізвище, ім’я, email, пароль)
* вхід до сайту (поля: email, пароль)
* профіль користувача (поля у табличному вигляді)
* про додаток (емблема додатку, короткий опис додатка)
* робоча сторінка додатка (розробляється самостійно студентом відповідно до обраної тематики)

## Time Tracker

Тематика додатка - Облік робочого часу.  
Додаток має наступний функціонал:

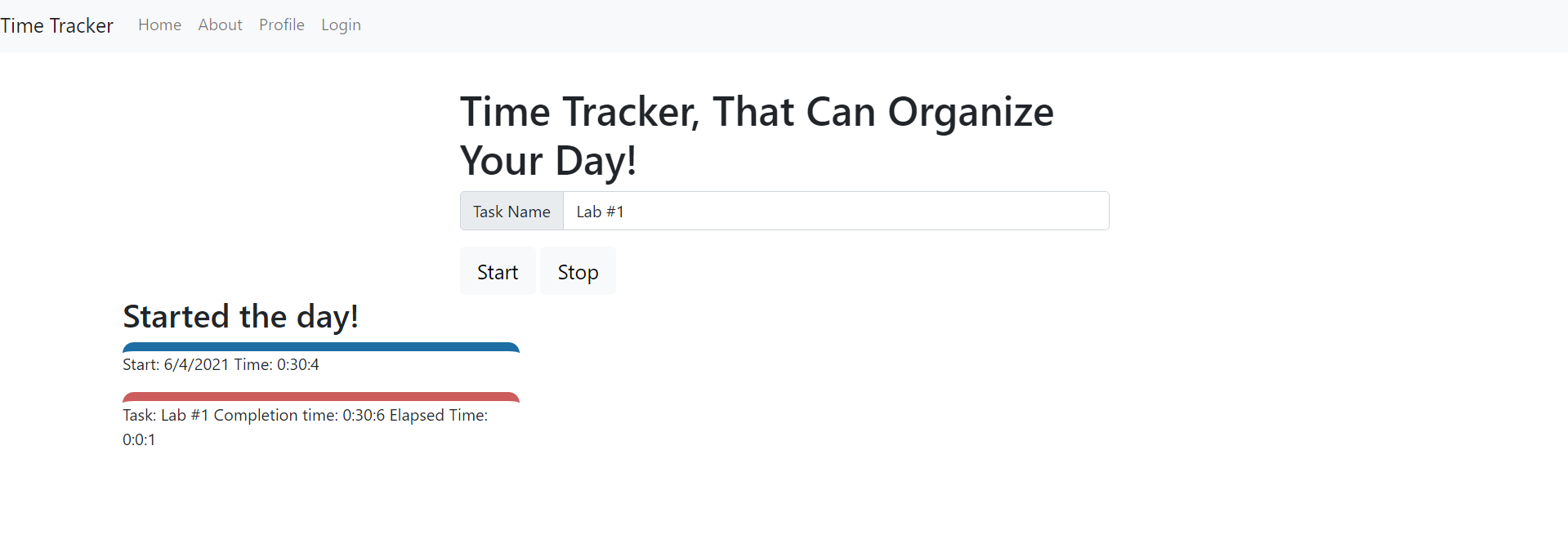
* запуск таймера
* призупинення/продовження, зупинка
* збереження назви, часу початку і завершення сеансу роботи

Наразі додаток має наступні сторінки:

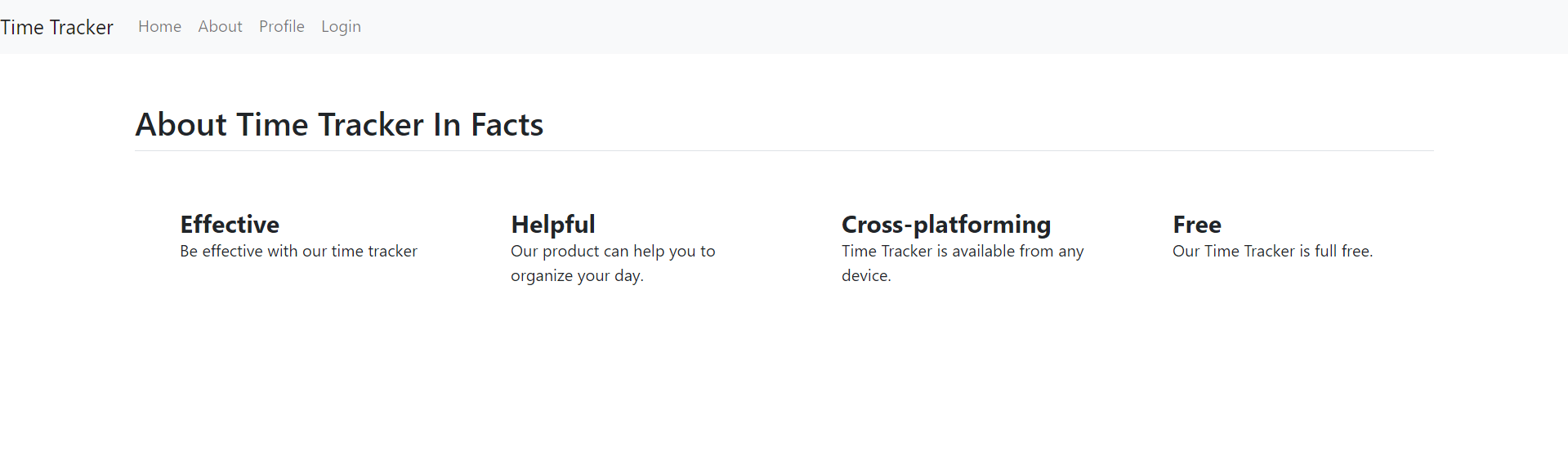
* Home - де і розташований додаток
* About - сторінка з основними фактами про додаток
* Profile - сторінка юзера, з його особистою інформацією
* Login - сторінка для входу у додаток
* Register - сторінка для реєстрації в додаток

Скріншоти усіх статичних сторінок додатка:

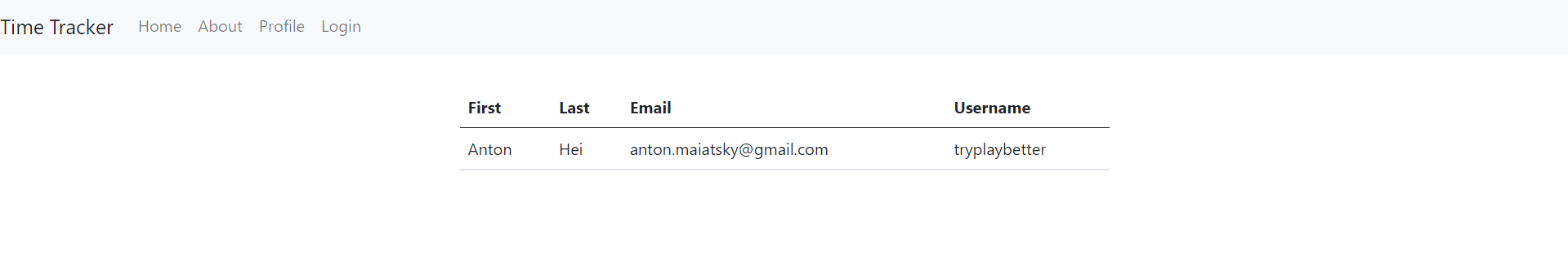
### Home



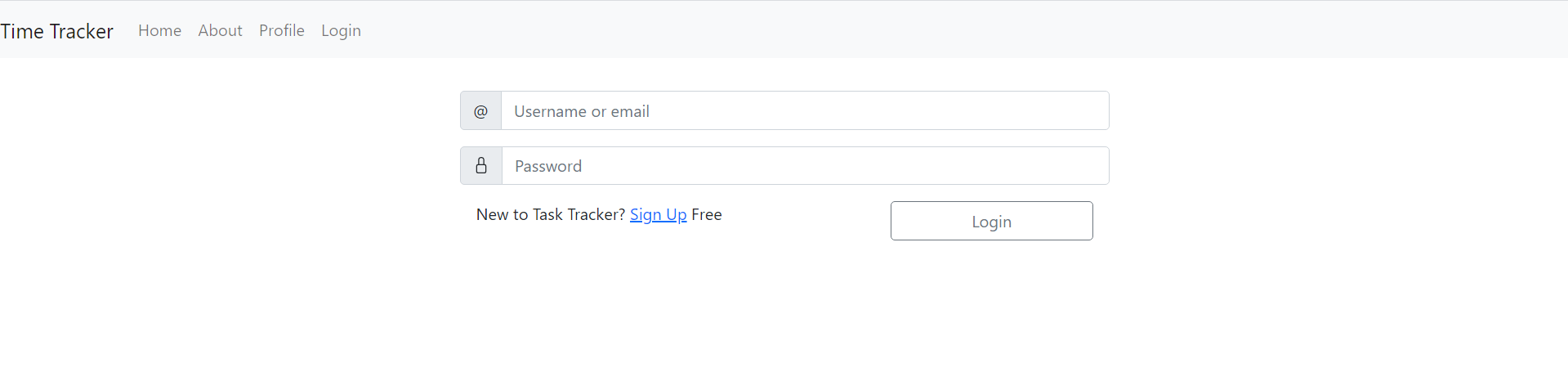
### About



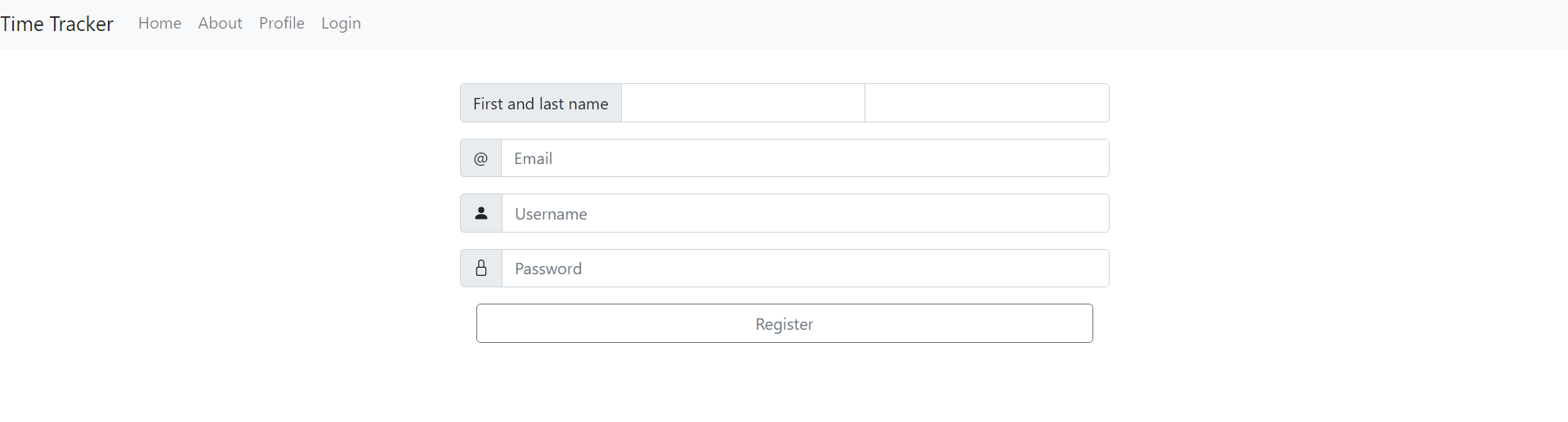
### Profile



### Login



### Register



### timeTracker.js

var lastTaskTime = Date.now();

function startButton() {

var dateNow = new Date();

var getList = document.getElementById("list");

var listItem = document.createElement("P");

listItem.classList.add("startedTask");

var getDate = dateNow.getMonth() + "/" + dateNow.getDate() + "/" + dateNow.getFullYear();

var getTime = dateNow.getHours() + ":" + dateNow.getMinutes() + ":" + dateNow.getSeconds();

var strDate = "Start: " + getDate + " Time: " + getTime;

lastTaskTime = Date.now();

listItem.innerText = strDate;

getList.appendChild(listItem);

}

function stopTask() {

var textValue = document.getElementById("taskInput").value;

var getList = document.getElementById("list");

var newTask = document.createElement("P");

newTask.classList.add("newTask")

var dateNow = new Date();

var getTime = dateNow.getHours() + ":" + dateNow.getMinutes() + ":" + dateNow.getSeconds();

document.getElementById("taskInput").innerText = "";

var elaspedSeconds = elapsedTime(lastTaskTime);

var elaspedTime = bunchOfSeconds(elaspedSeconds);

newTask.innerText = "Task: " + textValue + " Completion time: " + getTime + " " + elaspedTime;

getList.appendChild(newTask);

lastTaskTime = Date.now();

}

// utility function

function elapsedTime(preTime) {

var getNowTime = Date.now();

console.log(getNowTime);

console.log(preTime);

return (getNowTime - preTime) / 1000

}

// utility function

function bunchOfSeconds(secs) {

var seconds = 0;

var hours = 0;

var minutes = 0

console.log(secs)

while (secs > 3600) {

hours += 1;

secs -= 3600;

}

while (secs > 60) {

minutes += 1;

secs -= 60;

}

seconds = secs

return "Elapsed Time: " + hours + ":" + minutes + ":" + Math.floor(seconds)

}