**Міністерство освіти і науки України**

**Національний технічний університет України**

**“Київський політехнічний інститут”**

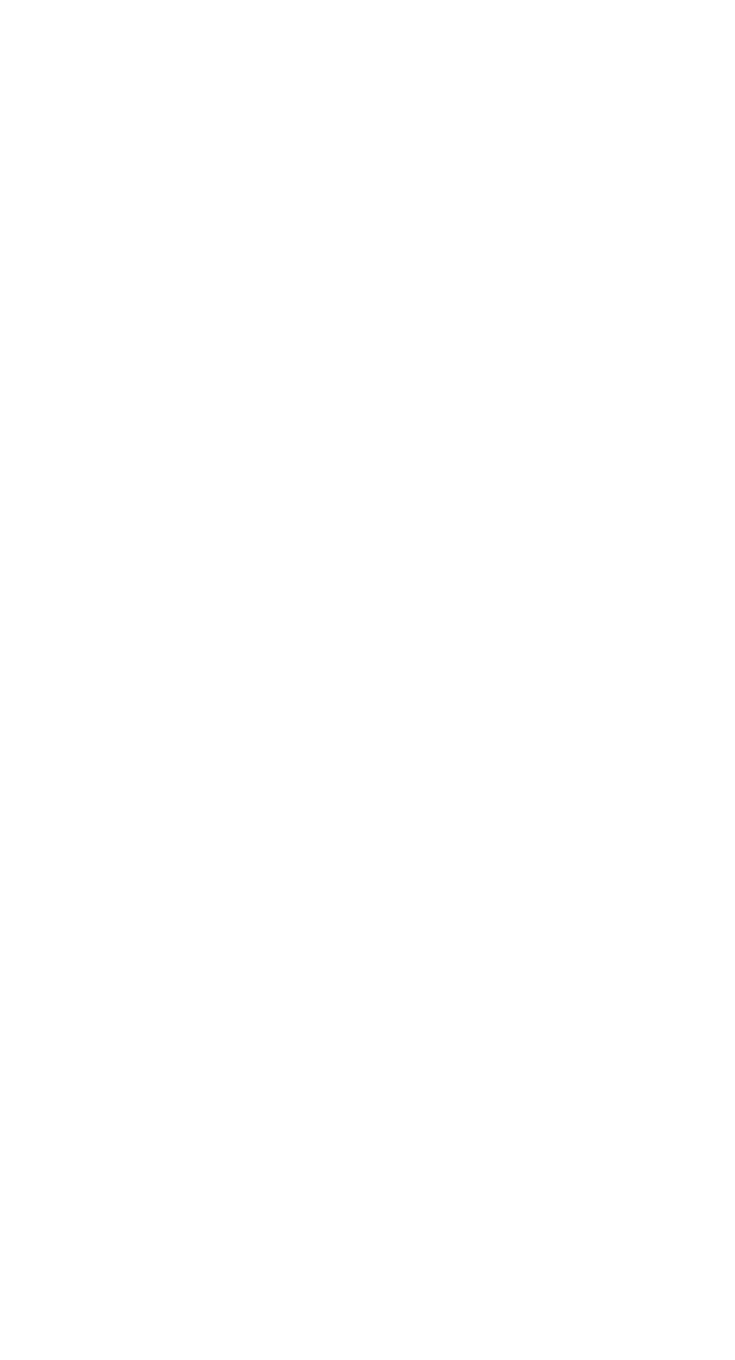
**Факультет прикладної математики**

**Кафедра спеціалізованих комп’ютерних систем**

**Лабораторна робота № 2**

з дисципліни «WEB-дизайн»

«**Розробка статичного інтерфейсу Web-додатка**»



Виконав:

студент групи КВ-84

**Мулярчук Микола**

Київ 2021

**Завдання**

‌**Мета‌**:‌ ‌ознайомитись‌ ‌із‌ ‌засобами‌ ‌мови‌ ‌Javascript‌ ‌та‌ ‌навчитись‌ ‌їх‌ ‌

застосовувати‌ ‌побудови‌ ‌Web-інтерфейсу‌ ‌користувача.‌ ‌

**Загальне‌ ‌завдання‌**:‌ ‌розробити‌ ‌функціональність‌ ‌для‌ ‌статичних‌ ‌сторінок‌ ‌

Web-додатку‌ ‌першої‌ ‌лабораторної‌ ‌роботи‌ ‌із‌ ‌використанням‌ ‌шаблону‌ ‌MVC‌ ‌

**Інструменти‌ ‌розробки‌**:‌ ‌мови‌ ‌HTML5,‌ ‌CSS3,‌ ‌Javascript.‌ ‌

**Зауваження‌**:‌ ‌Web-фреймворки‌ ‌(ReactJS,‌ ‌AngularJS,‌ ‌VueJS)‌ ‌не‌ ‌

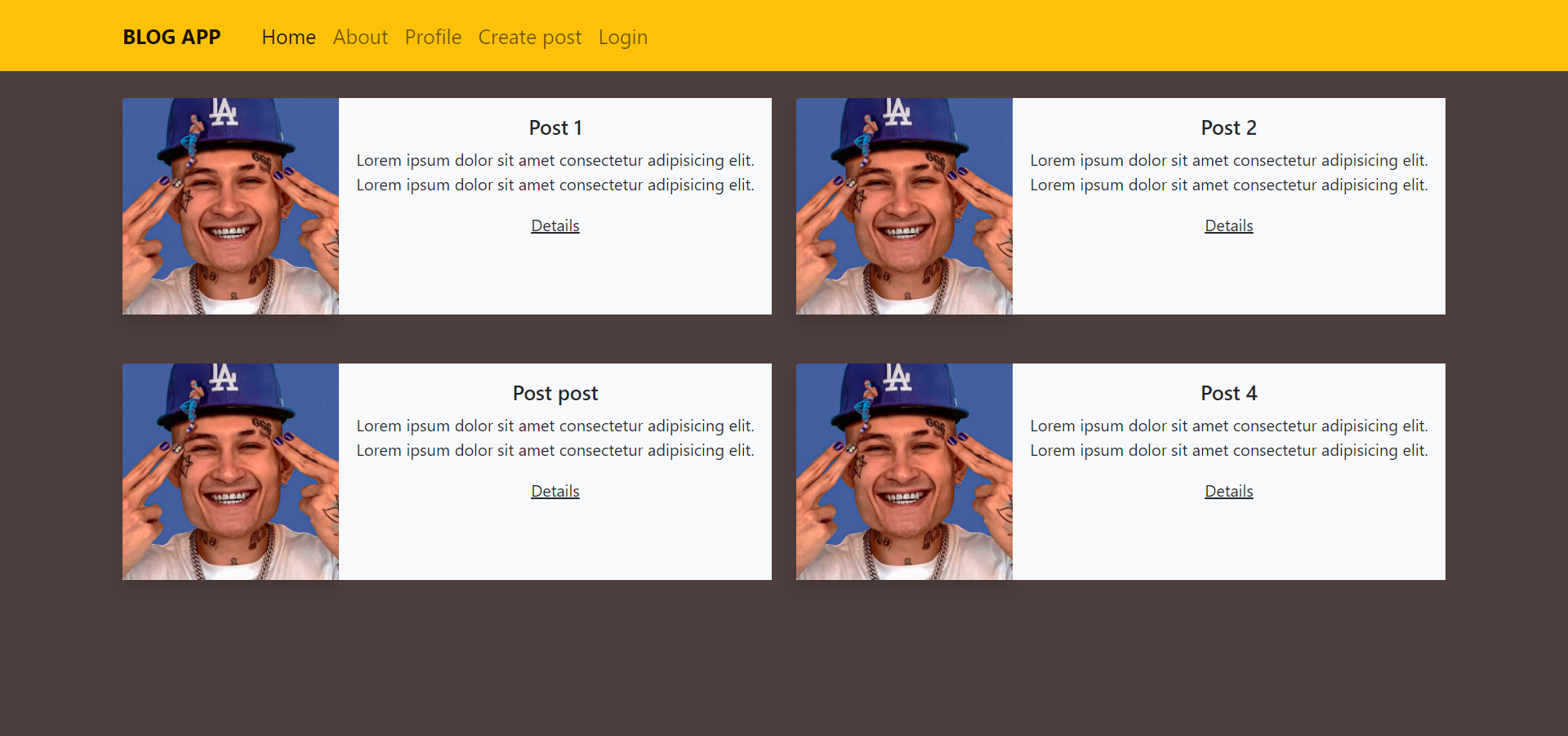
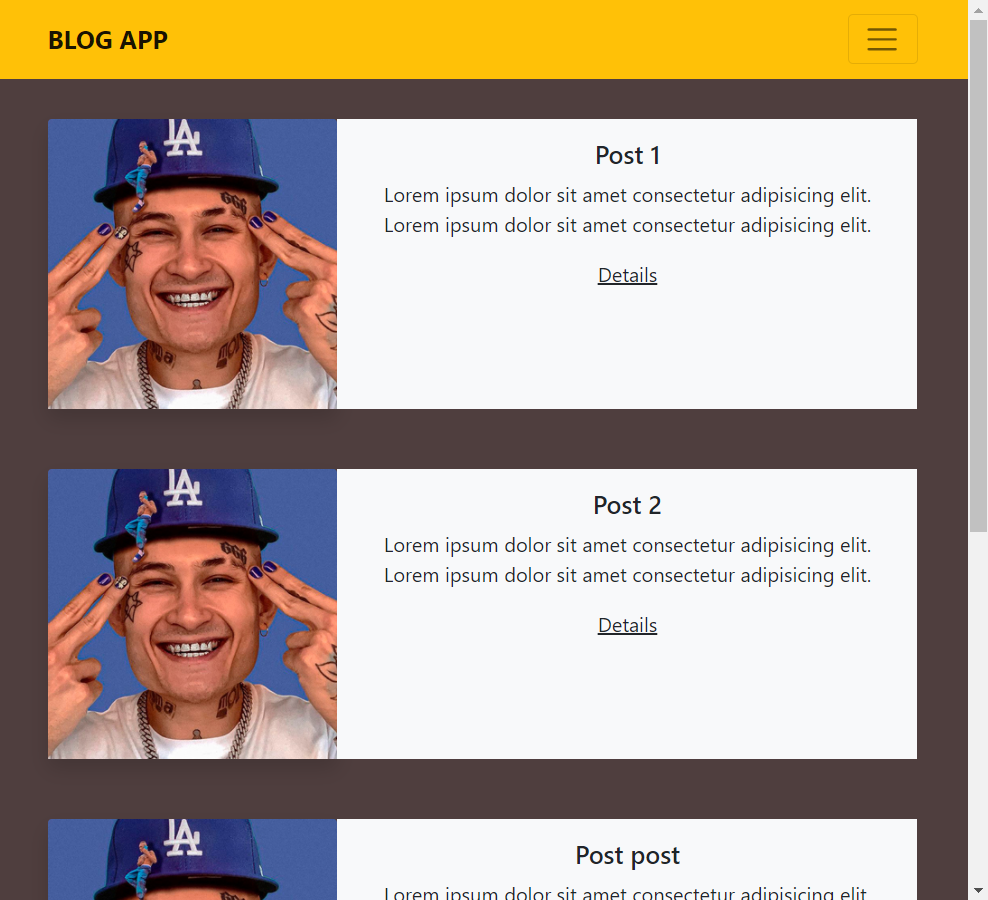
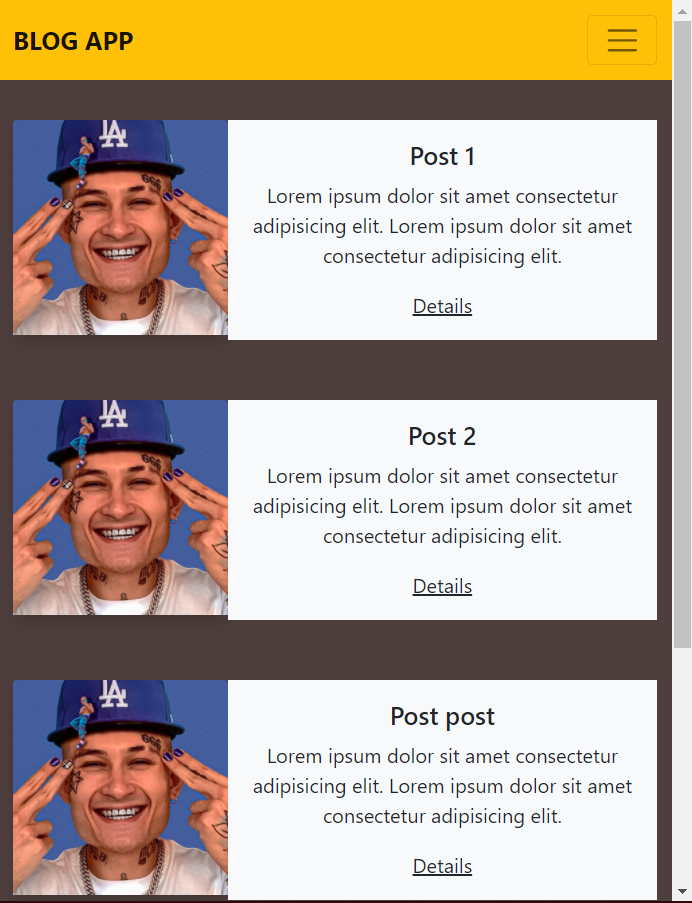
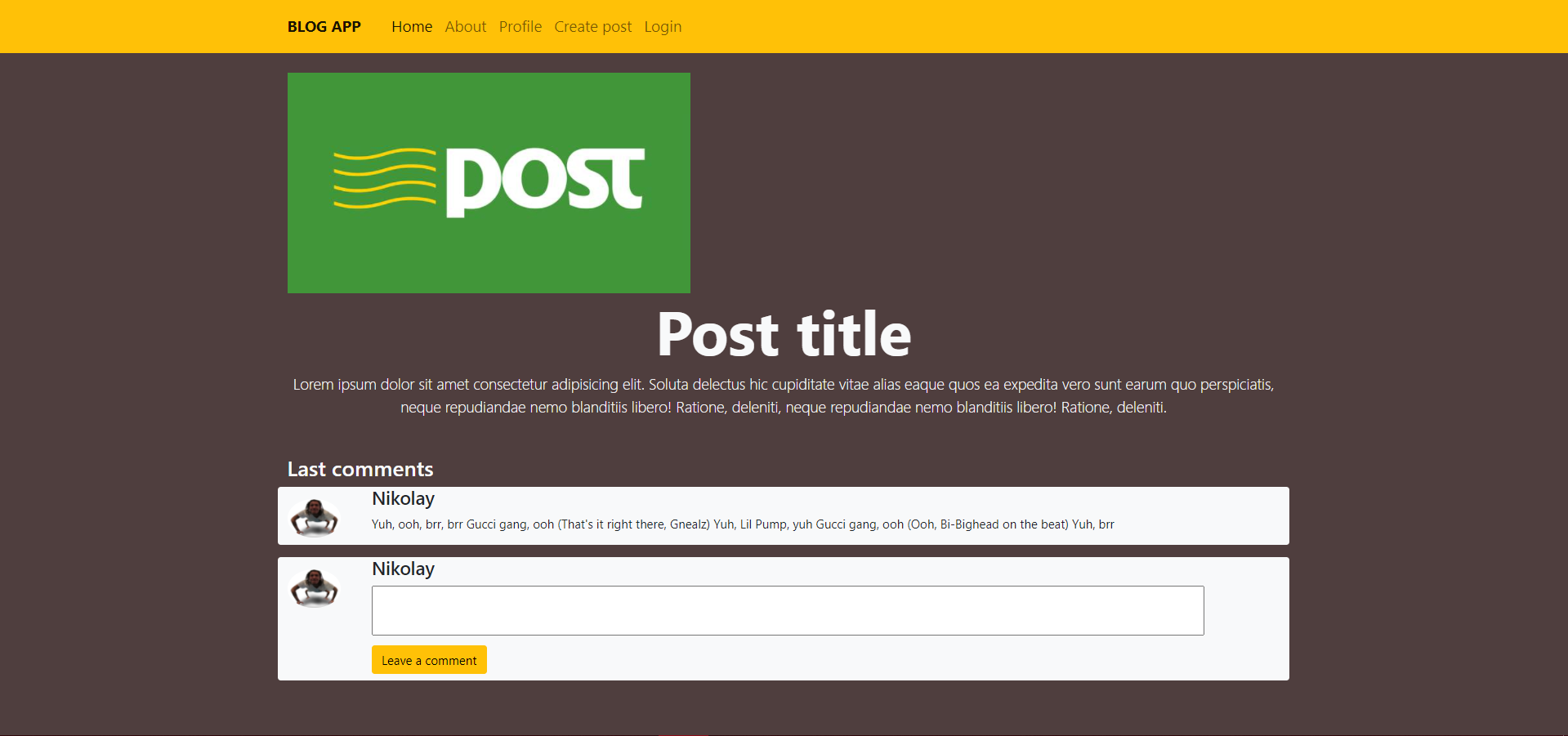
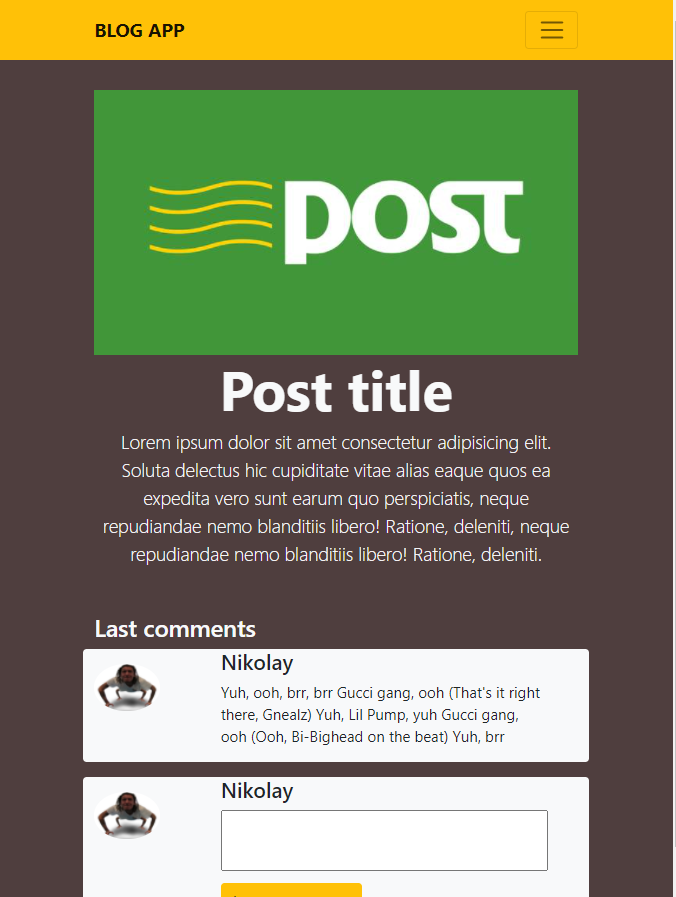
використовувати!‌ ‌

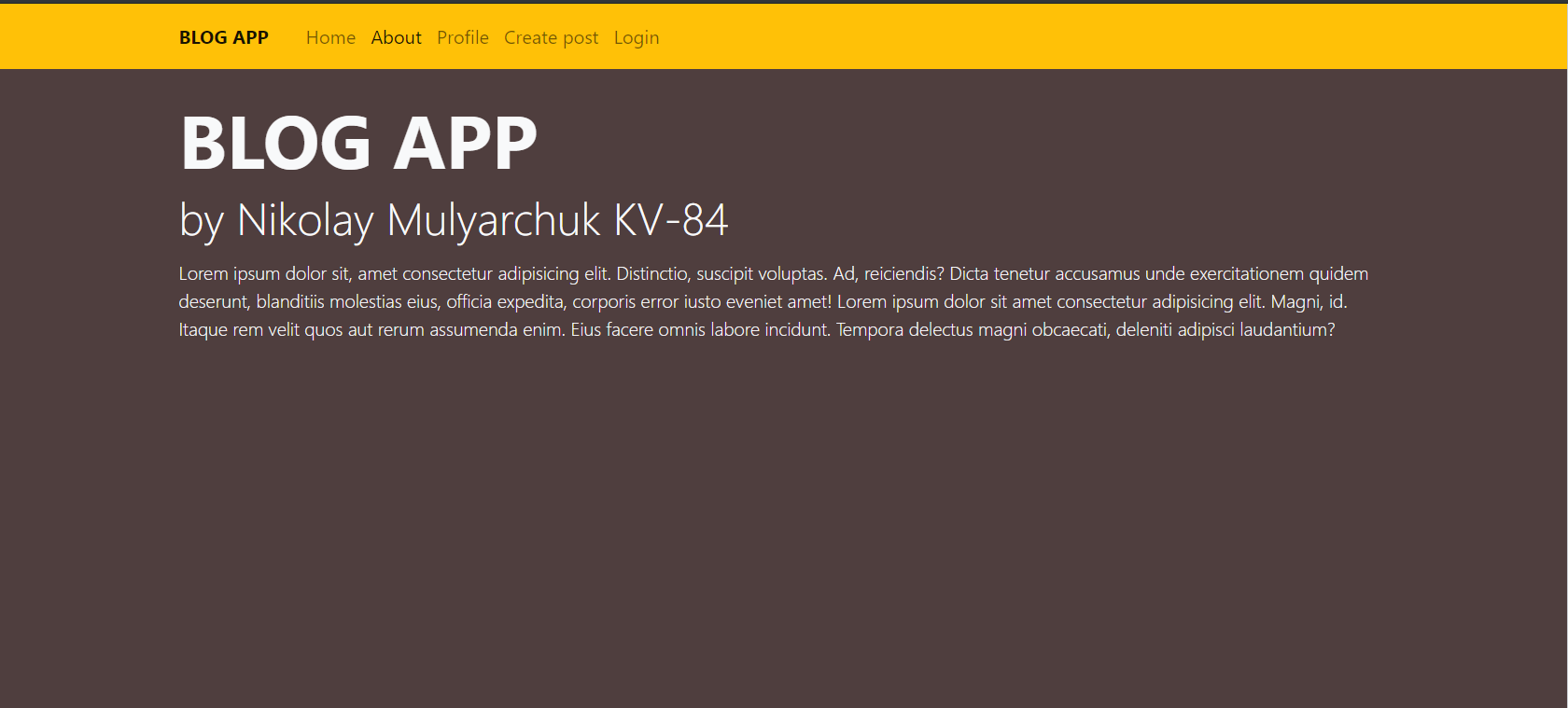
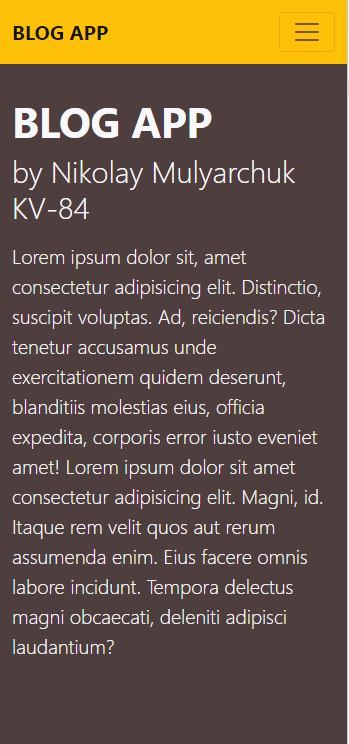
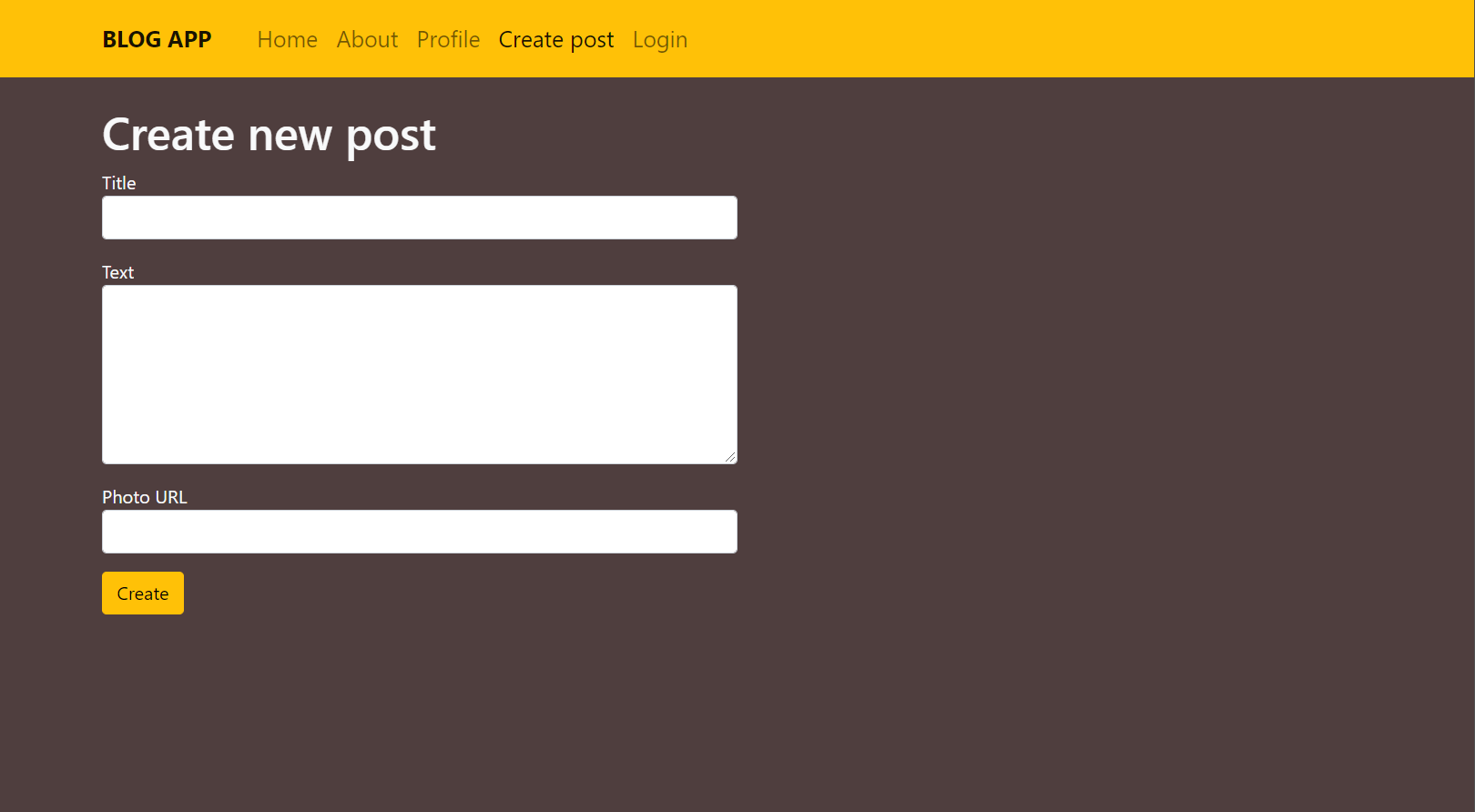
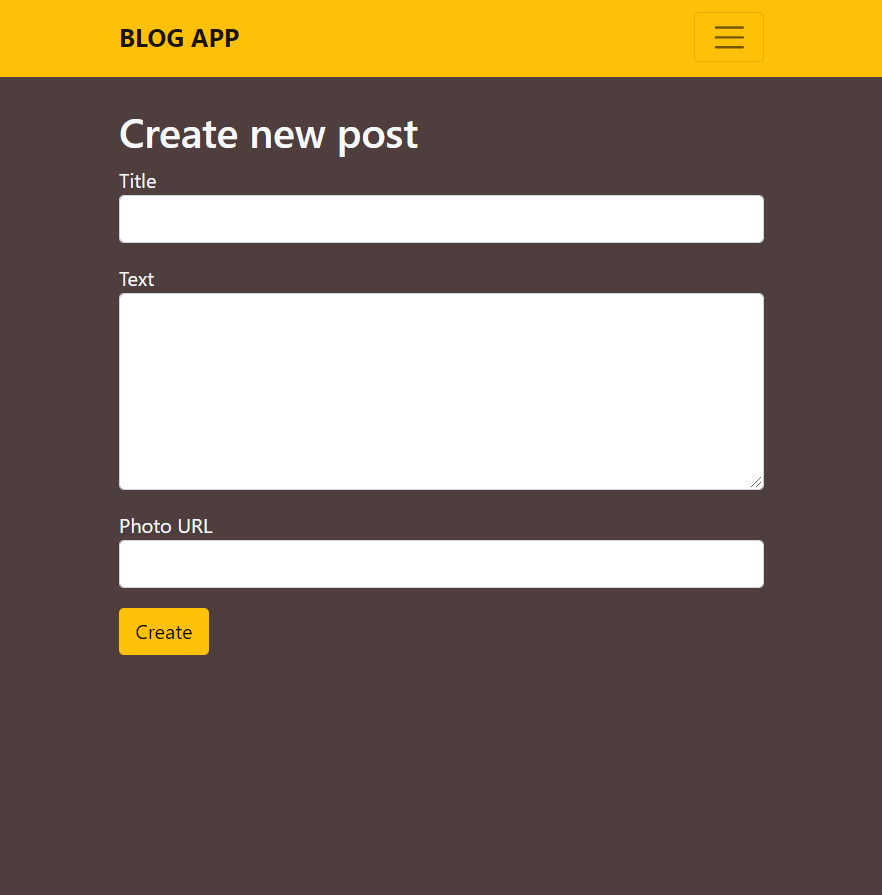
**Зміст**

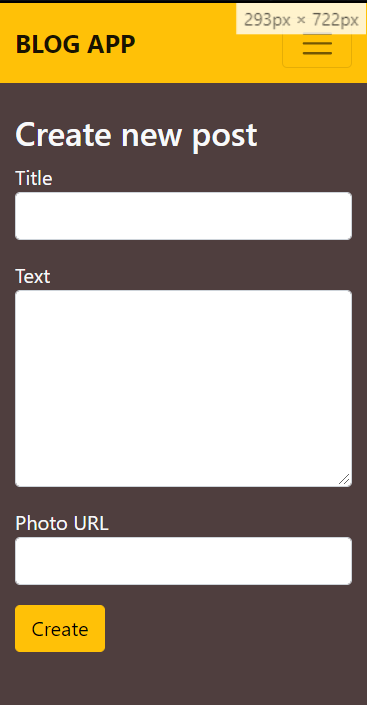
**Github: https://github.com/Bellis-lab/web\_labна drive**  
**Telegram:** @origin9

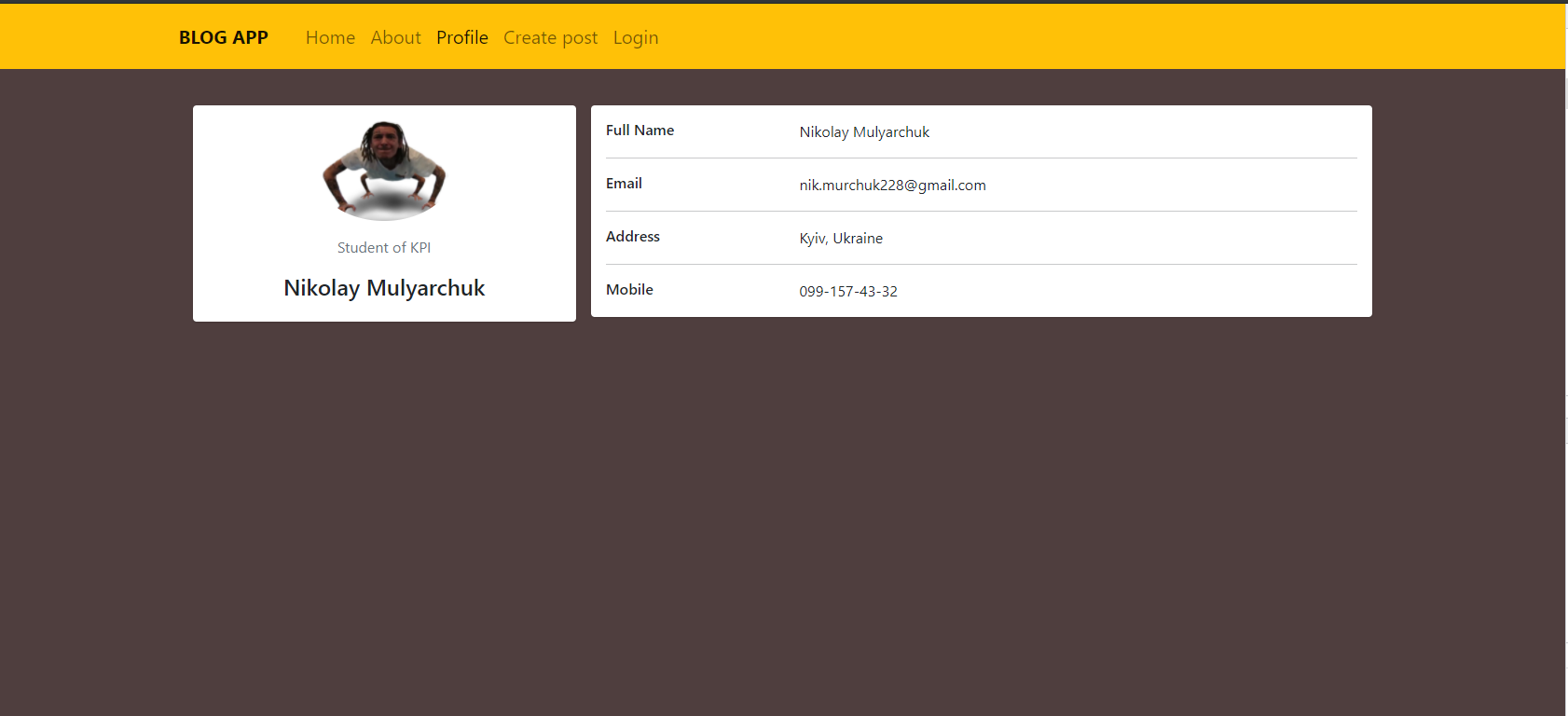
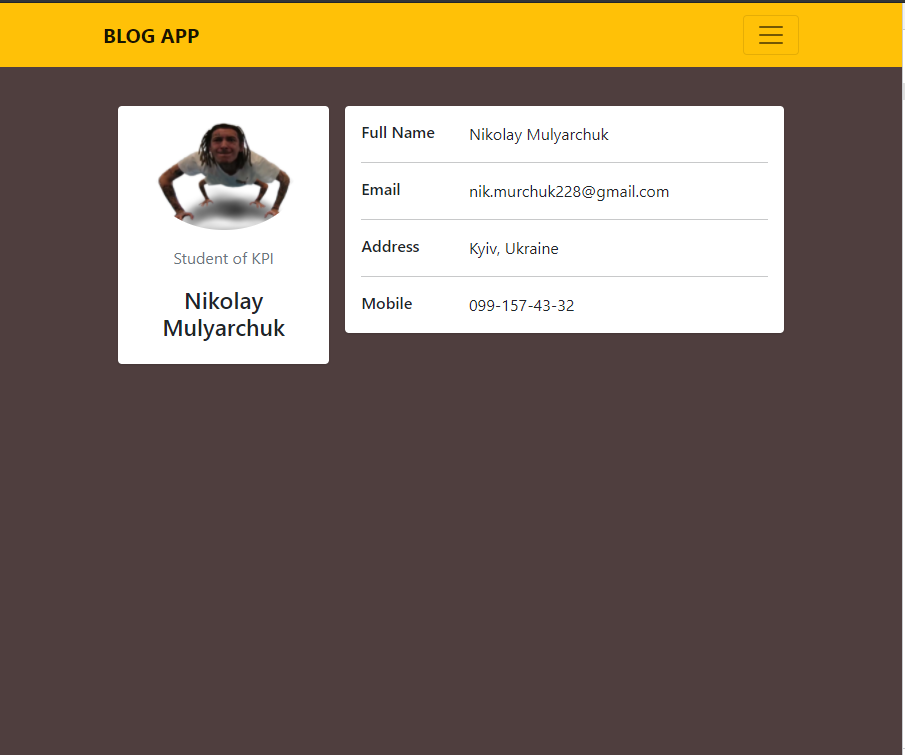
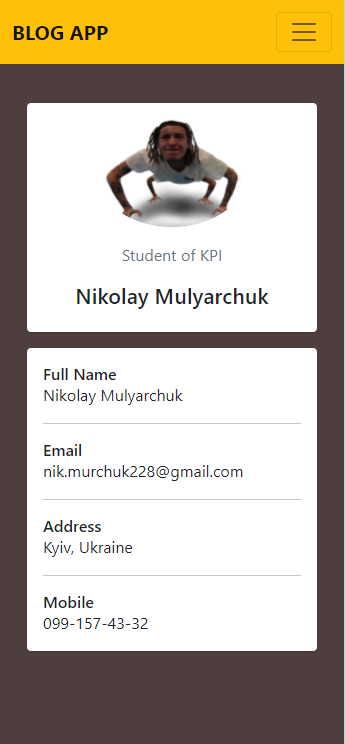
**Опис лабораторної:**Мною був створений “Блог” додаток з використанням HTML, CSS та фреймворку Bootstrap.

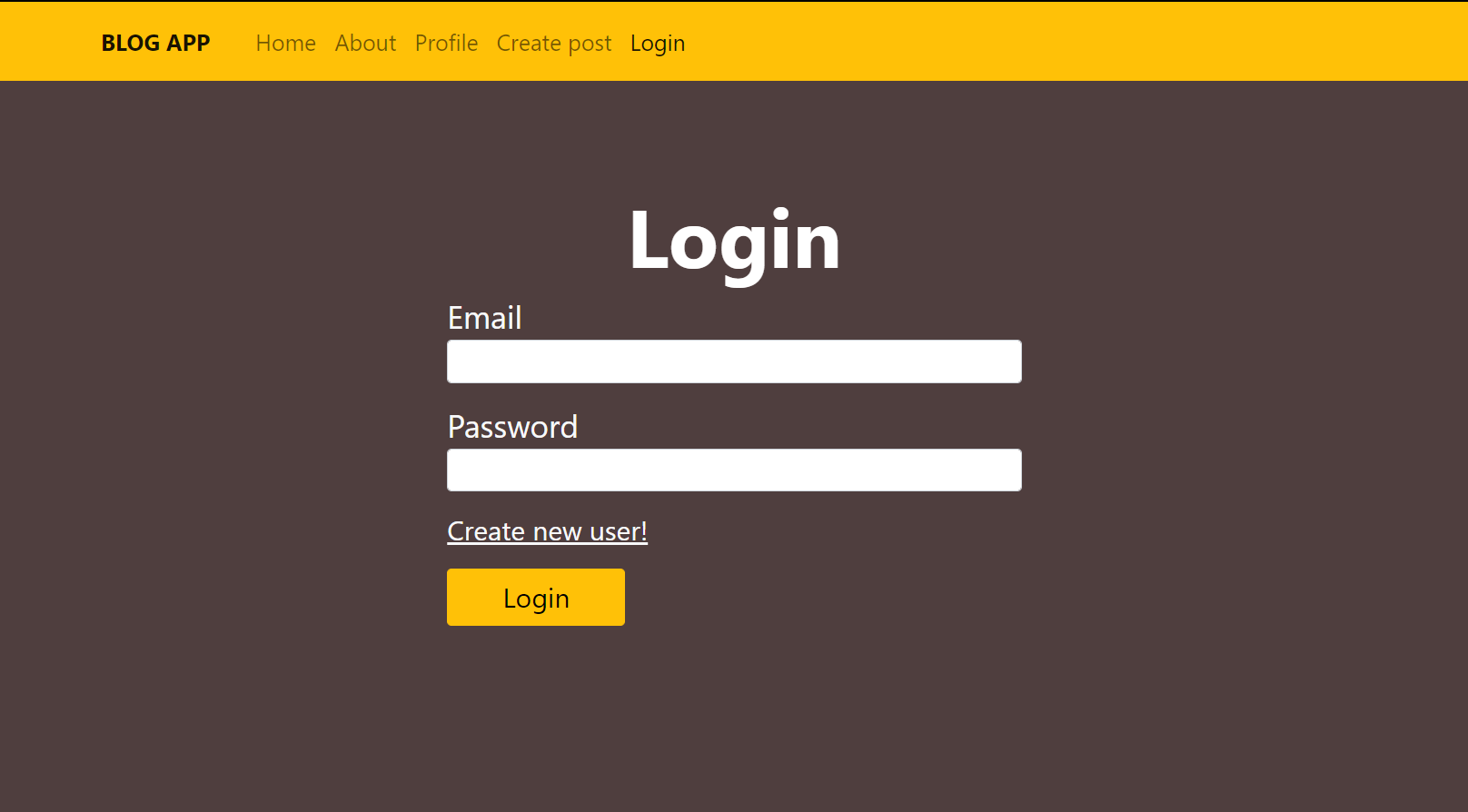
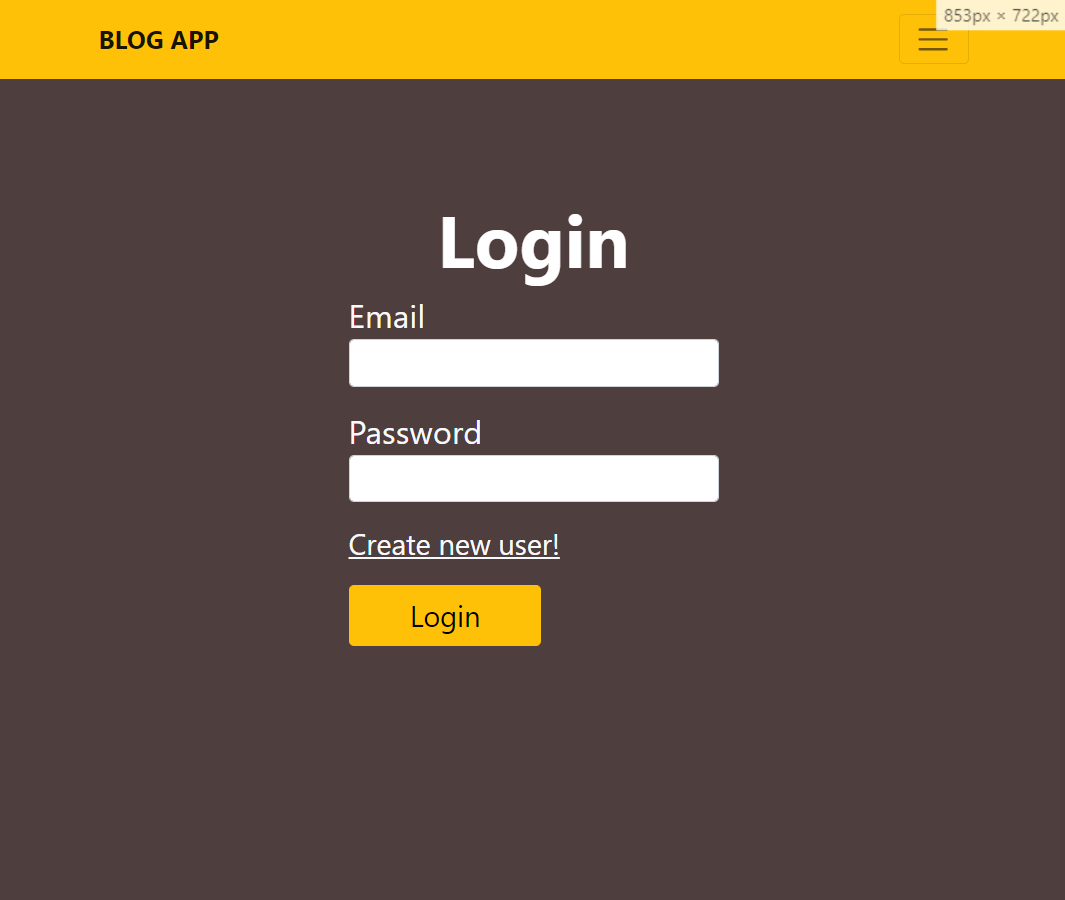
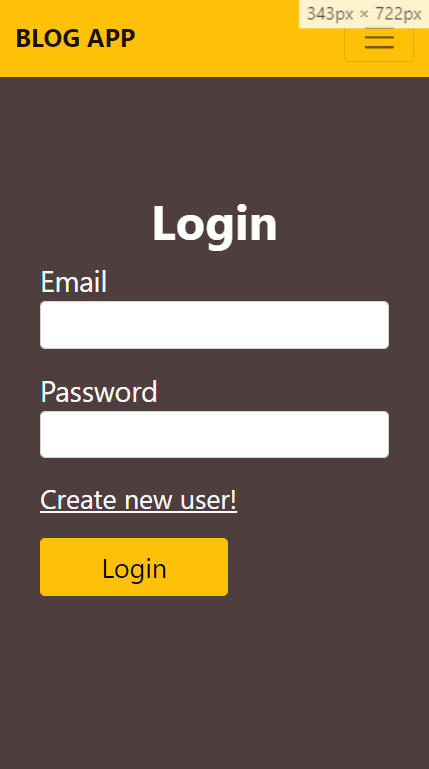
**HTML сторінки додатку**

* **index.html -** головна сторінка. На цій сторінці знаходяться всі останні опубліковані пости.   
  **Реалізовано** можливість виведення постів та перегляду детальної інформації.  
  Desktop  
    
  Tablet  
    
  Mobile  
  
* **post.html** - сторінка посту. Відображається назва, текст та зображення прикріплене до посту. Також під постом відображаються коментарі.  
  **Реалізовано** виведення інформацію про пост, зчитуючі query параметр із URL посилання, а також можливість додавання коментаря.  
    
  Desktop   
    
  Tablet  
    
  Mobile  
  

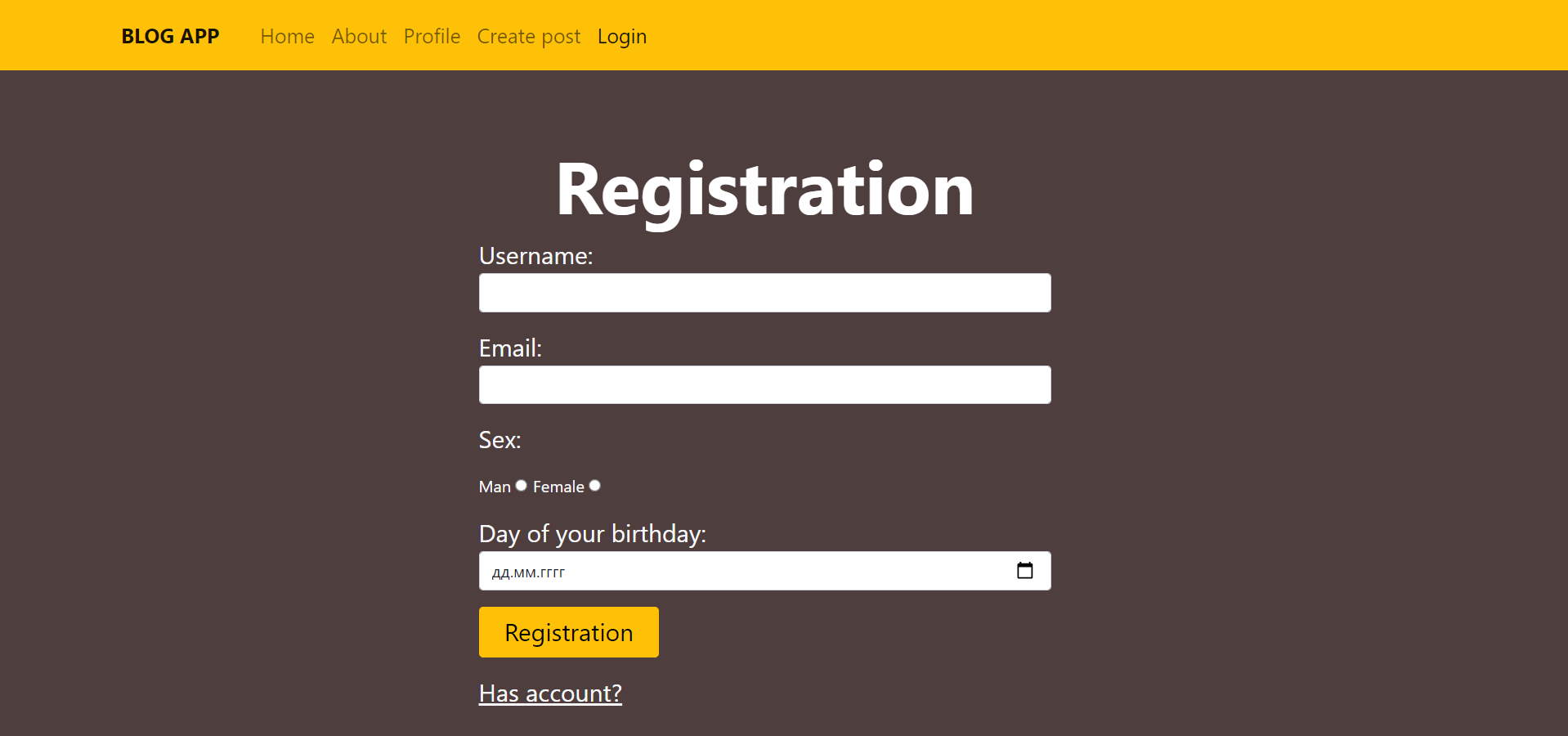
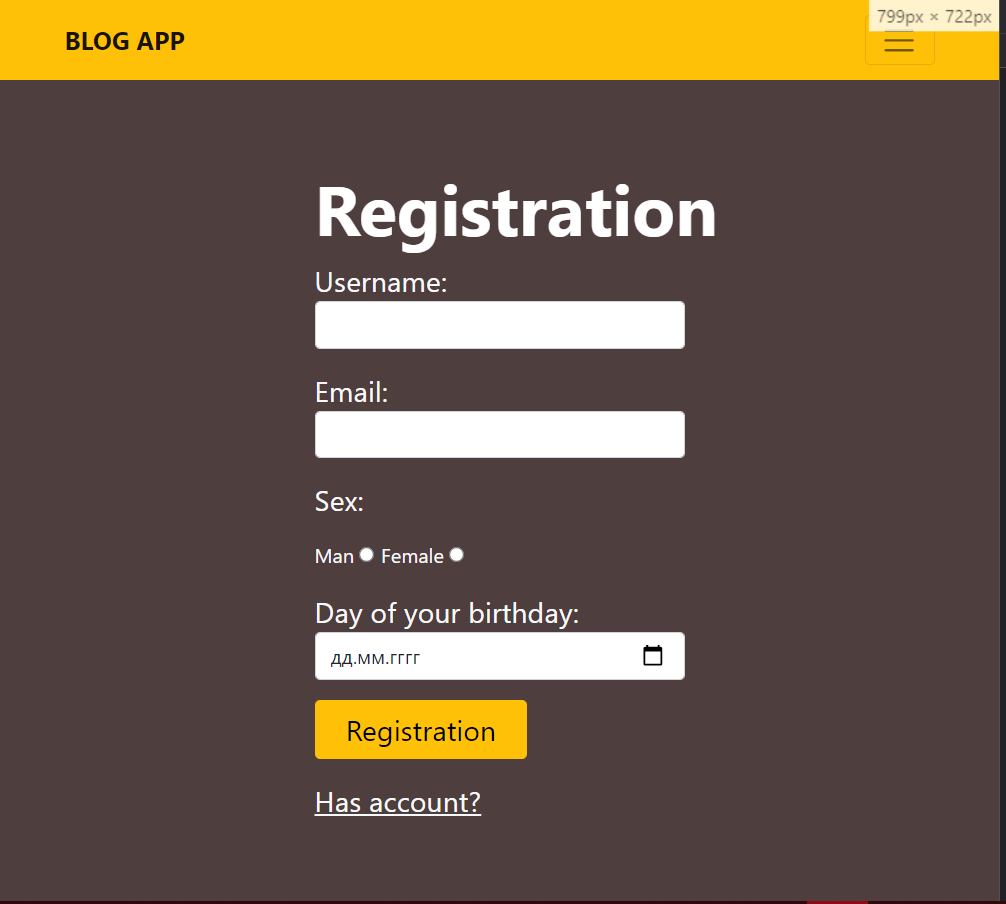
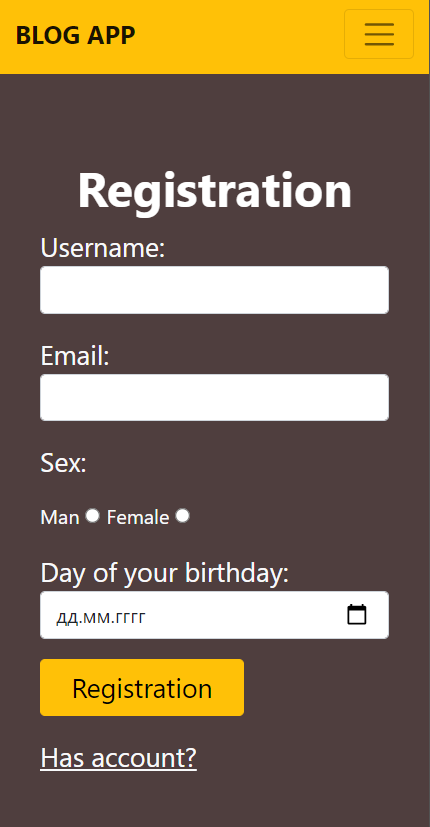
* **about.html -** сторінка з описом додатку.  
  Desktop  
     
  Tablet  
    
  Mobile  
  
* **new\_post.html**  - сторінка для створення нового поста.  
  Desktop  
  **Реалізовано** можливість додавання нового посту.  
    
  Tablet  
    
  Mobile



* **profile.html -** cторінка профілю. Взята з інтернету.   
    
    
  Tablet  
    
  Mobile  
  
* **login.html -** cторінка логіну.   
  **Реалізовано** авторизація користувача.

Desktop  
  
Tablet  
  
Mobile  


* **registration.html -** сторінка регістрації.  
  **Реалізовано** створення нового користувача.

  
  
Tablet   
Mobile  


# **Діаграма класів Model** - зберігає дані додатку та інформацію щодо роботи з ними. **Controller -** зберігає методи для реалізації функціоналу додатку **View -** зберігає методи для створення компонентів інтерфейсу.

# data.js

# let authStartData = {

# currentUser: null,

# allUsers: [

# {

# username: 'test',

# email: 'test@gmail.com',

# password: '000',

# sex: 'man',

# birthday: '01.01.0001'

# }

# ]

# };

# let postStartData = [

# {

# title: 'ABOBA 1',

# description:

# 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit.',

# img: '../images/post2.jpg',

# comments: [

# {

# username: 'Mafioznik from Spain ',

# usernameImg: '../images/kizaru-pauk-shablon.png',

# text: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit sit amet, consectetur adipisicing elitsit amet.'

# }

# ]

# },

# {

# title: 'ABOBA 2',

# description:

# 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit.',

# img: '../images/post2.jpg',

# comments: [

# {

# username: 'Mafioznik from Spain ',

# usernameImg: '../images/kizaru-pauk-shablon.png',

# text: 'Lorem ipsum dolor sit amet, consectetur adipisicing elit sit amet, consectetur adipisicing elitsit.'

# }

# ]

# },

# {

# title: 'ABOBA 3',

# description:

# 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit. Lorem ipsum dolor sit amet consectetur adipisicing elit.',

# img: '../images/post2.jpg',

# comments: [

# {

# username: 'Mafioznik from Spain ',

# usernameImg: '../images/kizaru-pauk-shablon.png',

# text: 'Lorem ipsum dolor sit amet.'

# }

# ]

# }

# ];

# *if* (!localStorage.getItem('authInfo')) {

# localStorage.setItem('authInfo', JSON.stringify(authStartData));

# }

# *if* (!localStorage.getItem('posts')) {

# localStorage.setItem('posts', JSON.stringify(postStartData));

# }

# 

# Model.js class Model {

# constructor() {

# const authInfo = JSON.parse(localStorage.getItem('authInfo'));

# *this*.currentUser = authInfo.currentUser;

# *this*.allUsers = authInfo.allUsers;

# *this*.posts = JSON.parse(localStorage.getItem('posts'));

# }

# getPosts = () => {

# *return* *this*.posts;

# };

# getPost = id => {

# *return* *this*.posts[id];

# };

# createPost = post => {

# *this*.posts.push(post);

# localStorage.setItem('posts', JSON.stringify(*this*.posts));

# *return*;

# };

# createComment = (id, newComment) => {

# *this*.posts[id].comments.push(newComment);

# localStorage.setItem('posts', JSON.stringify(*this*.posts));

# *return*;

# };

# createUser = user => {

# *this*.allUsers.push(user);

# *this*.currentUser = user;

# localStorage.setItem(

# 'authInfo',

# JSON.stringify({

# currentUser: *this*.currentUser,

# allUsers: *this*.allUsers

# })

# );

# *return*;

# };

# authorize = user => {

# const existedUser = *this*.allUsers.find(

# u => user.email === u.email && user.password === u.password

# );

# *if* (existedUser) {

# *this*.currentUser = existedUser;

# localStorage.setItem(

# 'authInfo',

# JSON.stringify({

# currentUser: *this*.currentUser,

# allUsers: *this*.allUsers

# })

# );

# *return* true;

# } *else* *return* false;

# };

# logout = () => {

# *this*.currentUser = null;

# localStorage.setItem(

# 'authInfo',

# JSON.stringify({

# currentUser: *this*.currentUser,

# allUsers: *this*.allUsers

# })

# );

# *return*;

# };

# }

# 

# View.js class View {

# LogoutBtn = () => {

# let btn = document.createElement('a');

# btn.classList.add('nav-link');

# btn.innerHTML = 'Logout';

# *return* btn;

# };

# Post = (id, post) => {

# let postComponent = document.createElement('div');

# postComponent.classList.add('col-12', 'col-xl-6', 'mb-5');

# postComponent.innerHTML = `

# <div class="row g-0">

# <div class="col-4">

# <img class="card-img-top shadow" src="${post.img}"

# alt="Card image cap" />

# </div>

# <div class="col-8 bg-light">

# <div class="card-body">

# <h5 class="card-title">${post.title}</h5>

# <p class="card-text">

# Lorem ipsum dolor sit amet consectetur

# adipisicing elit. Lorem ipsum dolor sit amet

# consectetur adipisicing elit.

# </p>

# <a href="./post.html?post=${id}" class="text-dark py-2">Details</a>

# </div>

# </div>

# </div>`;

# *return* postComponent;

# };

# PostInfo = (title, img, description) => {

# let postInfo = document.createElement('div');

# postInfo.innerHTML = `

# <div class="row flex-column flex-lg-row ">

# <img src="${img}" class="col-12 col-lg-5 img-fluid order-1"

# alt="Responsive image" />

# </div>

# <div class="col-12 text-light">

# <h1 class="display-1 fw-bold">${title}</h1>

# <p class="lead">

# ${description}</p>

# </div>

# `;

# *return* postInfo;

# };

# Comment = (username, usernameImg, text) => {

# let comment = document.createElement('div');

# comment.classList.add(

# 'rounded',

# 'row',

# 'justify-content-md-center',

# 'bg-light',

# 'text-dark',

# 'mb-3'

# );

# comment.innerHTML = `

# <div class="col-1 pb-3">

# <img src="${usernameImg}" alt="avatar" class="border border-dark mt-3 rounded-circle" width="70" />

# </div>

# <div class="col-11 text-start">

# <div class="row">

# <div class="col mt-2 mx-2">

# <h4>${username}</h4>

# </div>

# </div>

# <div class="row">

# <div class="col">

# <p class="col text-wrap">

# ${text}

# </p>

# </div>

# </div>

# </div>`;

# *return* comment;

# };

# }

# 

# Controller.js class Controller {

# constructor() {

# *this*.model = new Model();

# *this*.view = new View();

# }

# renderHome = () => {

# const posts = *this*.model.getPosts();

# posts.forEach((post, id) => {

# const postElem = *this*.view.Post(id, post);

# document.querySelector('#posts-container').appendChild(postElem);

# });

# *return*;

# };

# postForm = () => {

# document.querySelector('#post-form').onsubmit = () => {

# const title = document.querySelector('#title').value;

# const img = document.querySelector('#image').value;

# const description = document.querySelector('#description').value;

# const post = { title, img, description, comments: [] };

# *this*.model.createPost(post);

# };

# *return*;

# };

# renderInfoPost = id => {

# const post = *this*.model.getPost(id);

# const postElem = *this*.view.PostInfo(

# post.title,

# post.img,

# post.description

# );

# document.querySelector('#post-container').appendChild(postElem);

# post.comments.forEach(comment => {

# const commentElem = *this*.view.Comment(

# comment.username,

# comment.usernameImg,

# comment.text

# );

# document

# .querySelector('#comments-container')

# .appendChild(commentElem);

# });

# document.querySelector('#comment-form').onsubmit = event => {

# event.preventDefault();

# const text = document.querySelector('#text').value;

# const comment = {

# username: 'Nikolay',

# text,

# usernameImg: '../images/kizaru-pauk-shablon.png'

# };

# *this*.model.createComment(id, comment);

# location.reload();

# };

# *return*;

# };

# loginForm = () => {

# document.querySelector('#login-form').onsubmit = () => {

# const email = document.querySelector('#email').value;

# const password = document.querySelector('#password').value;

# const user = { email, password };

# *this*.model.authorize(user);

# !*this*.model.currentUser

# ? alert('Incorrect login or password')

# : null;

# };

# *return*;

# };

# registerForm = () => {

# document.querySelector('#registration-form').onsubmit = () => {

# const email = document.querySelector('#email').value;

# const password = document.querySelector('#password').value;

# const sex = [...document.getElementsByName('sex')].find(

# btn => btn.checked

# ).value;

# const birthday = document.querySelector('#birthday').value;

# const newUser = { email, password, sex, birthday };

# *this*.model.createUser(newUser);

# };

# *return*;

# };

# auth = () => {

# *if* (!*this*.model.currentUser) {

# [...document.querySelectorAll('.profile')].map(

# link => (link.style.display = 'none')

# );

# *if* (window.location.href.includes('/new\_post')) {

# document.getElementById('comment-form').style.display = 'none';

# }

# } *else* {

# const logoutBtn = *this*.view.LogoutBtn();

# logoutBtn.onclick = () => {

# *this*.model.logout();

# location.replace('/index.html');

# };

# document.querySelector('#login-logout').innerHTML = '';

# document.querySelector('#login-logout').appendChild(logoutBtn);

# }

# *return*;

# };

# }

# const appController = new Controller();

# appController.auth();

# const url = window.location.href;

# *if* (url.includes('/new\_post')) {

# appController.postForm();

# }

# *if* (url.includes('/post')) {

# const urlParams = new URLSearchParams(window.location.search);

# const id = urlParams.get('post');

# appController.renderInfoPost(id);

# }

# *if* (url.includes('/login')) {

# appController.loginForm();

# }

# *if* (url.includes('/registration')) {

# appController.registerForm();

# }

# *if* (url.includes('/index.html')) {

# 

# appController.renderHome();

# }

# 