**Міністерство освіти і науки України**

**Національний технічний університет України**

**“Київський політехнічний інститут”**

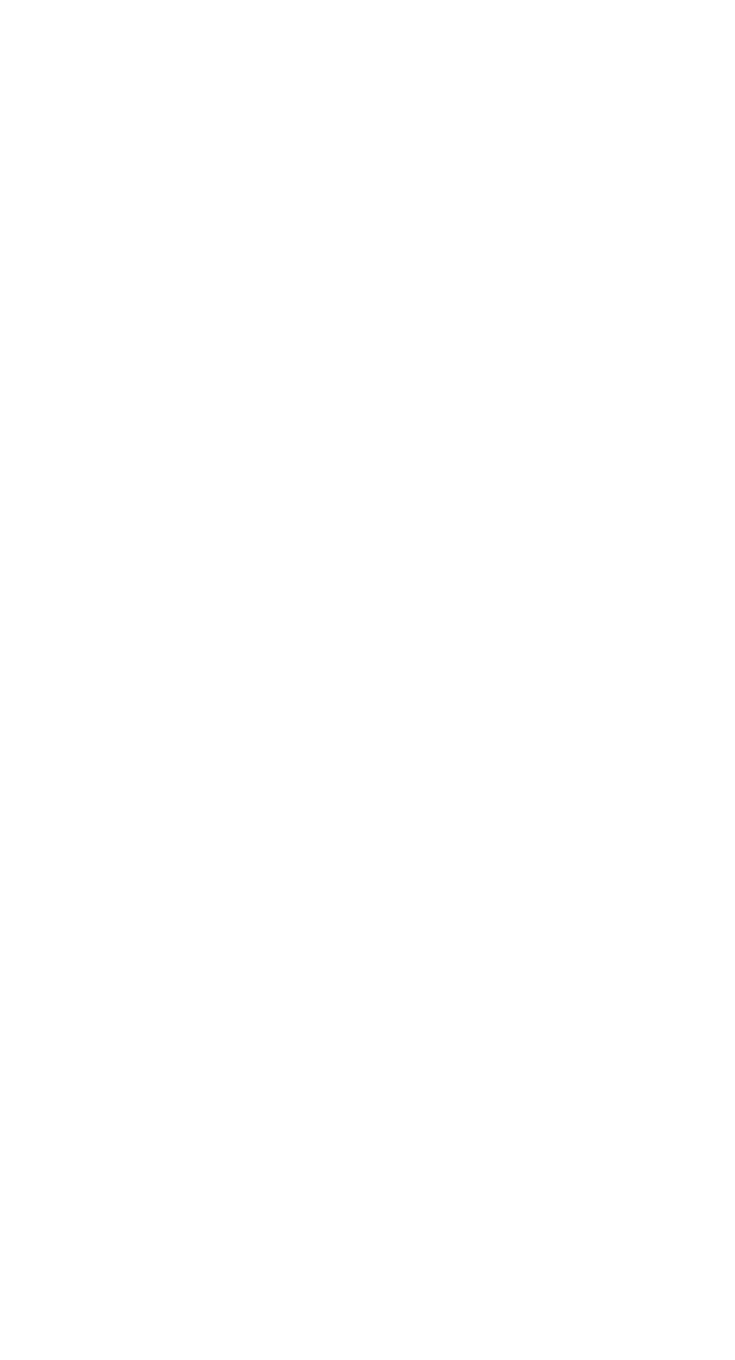
**Факультет прикладної математики**

**Кафедра спеціалізованих комп’ютерних систем**

**Лабораторна робота №2**

з дисципліни «ВЕБ-дизайн»

«**Розробка функціональності Web-додатка мовою Javascript**»



Виконав:

студент групи КВ-83

**Цесля Антон**

Київ 2021

**Зміст**

**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

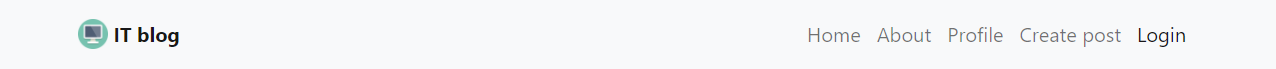
**Інструменти розробки**: мови HTML5, CSS3, Javascript.

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

**Завдання**

**Тема:** “Блог про інформаційні технології”.   
**Посилання на Github-репозиторій:** https://github.com/deusantonious/Web  
**Телеграм-контакт:** @deus\_antonious

**Опис додатку:**  
У ході виконання даної лабораторної работи мною був створений статичний інтерфейс ВЕБ-додатку на тему “Блог про інформаційні технології”. Для кожної веб сторінки був розроблений header з навігацією по веб-сторінкам.

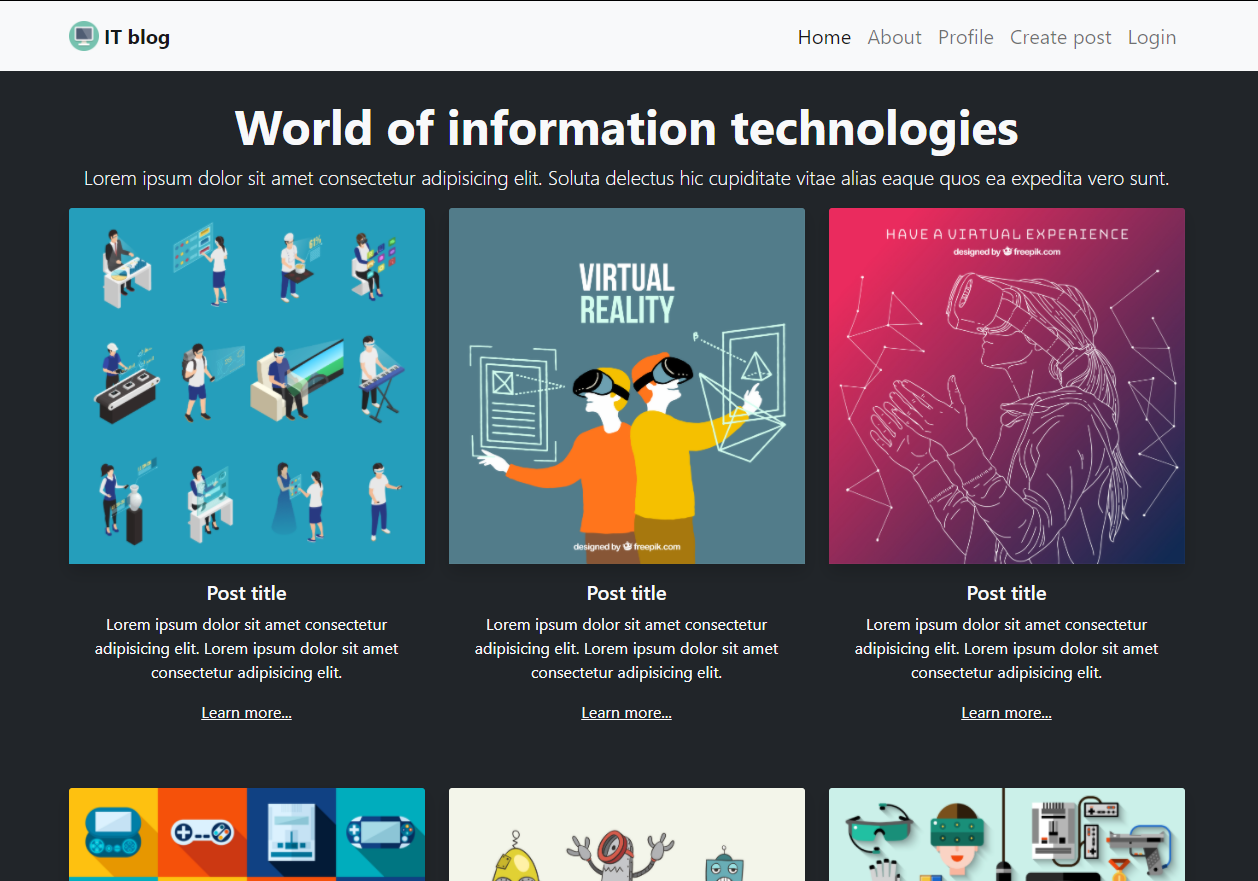


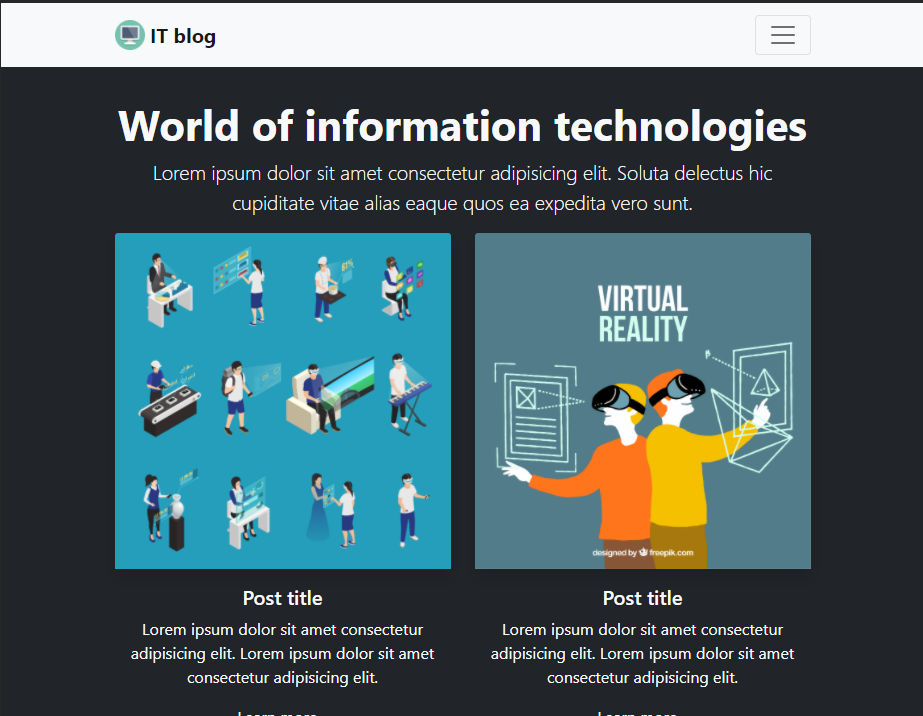
Зовнішній вигляд на планшетах та телефонах:

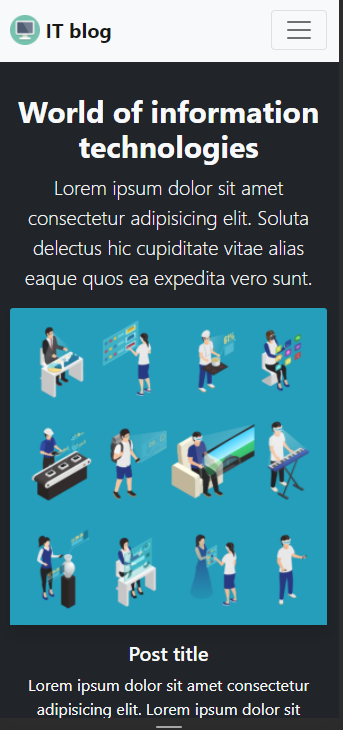
  
Були створені наступні html-сторінки:

* **index.html**Сторінка є “домашньою”. Вона має відкриватися першою, як тільки користувач переходить на наш сайт. На цій сторінці відображаються всі пости.

Зовнішній вигляд на компьютерах:

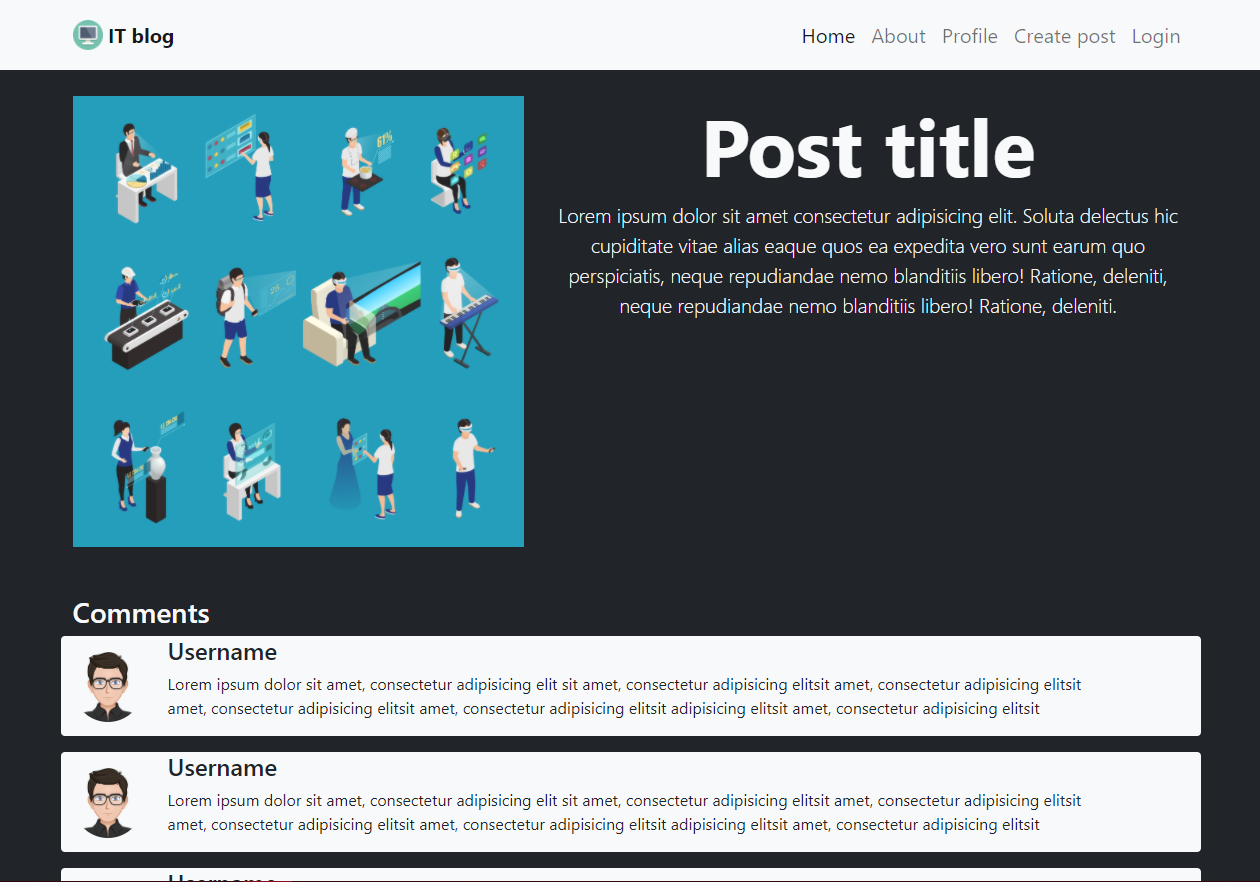
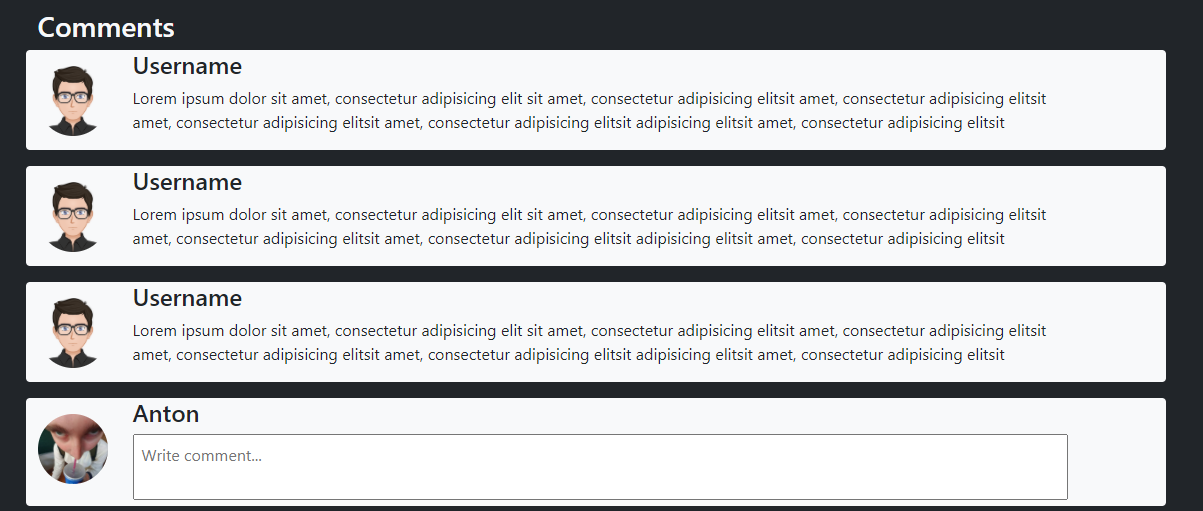




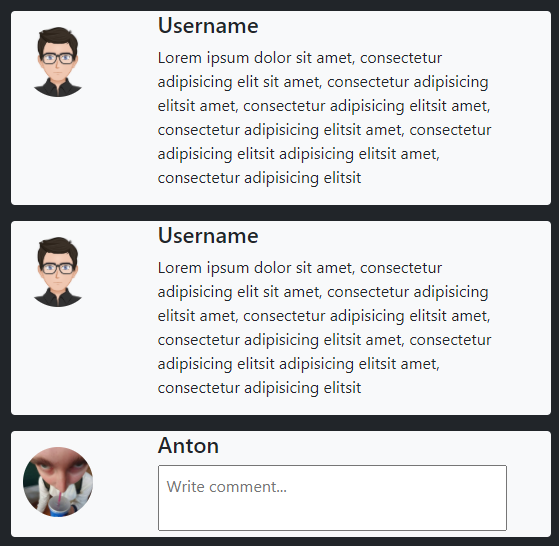
Зовнішній вигляд на телефонах:

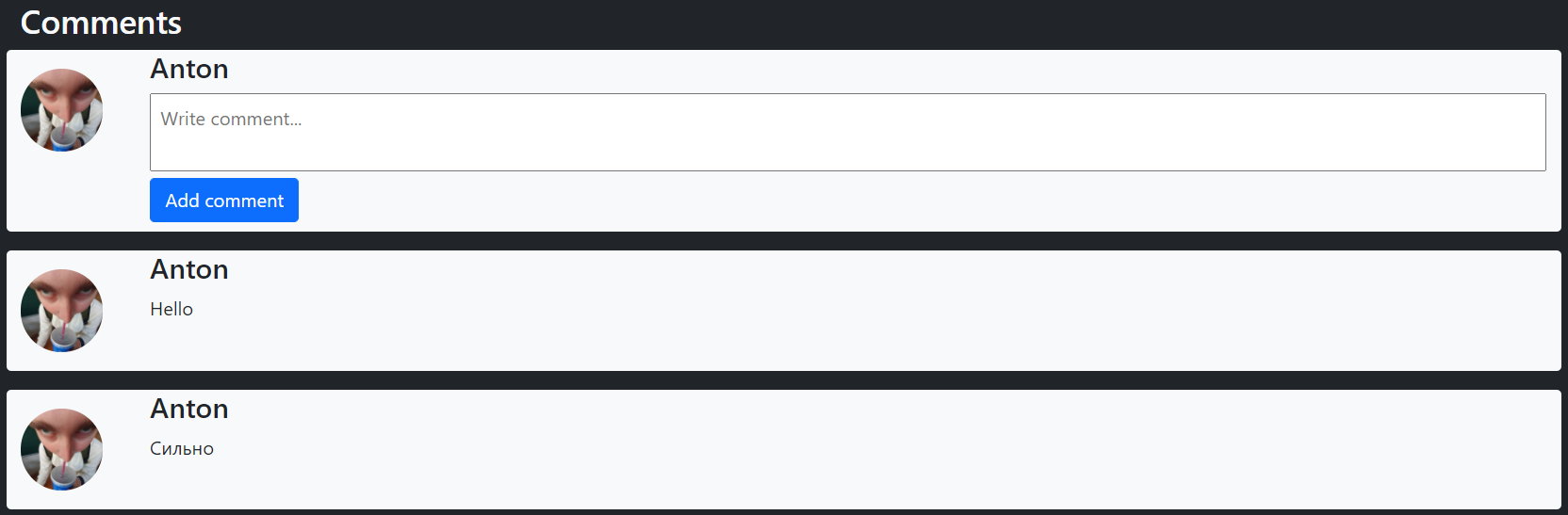
**UPDATE**При відкритті сторінки пости завантажуються з local storage.

* **post.html**Якщо на сторінці index.html під постом перейти за посиланням “Learn more…”, то відкривається сторінка з описом поста та можливістю залишити коментар.

  
  
Зовнішній вигляд на телефонах та планшетах:



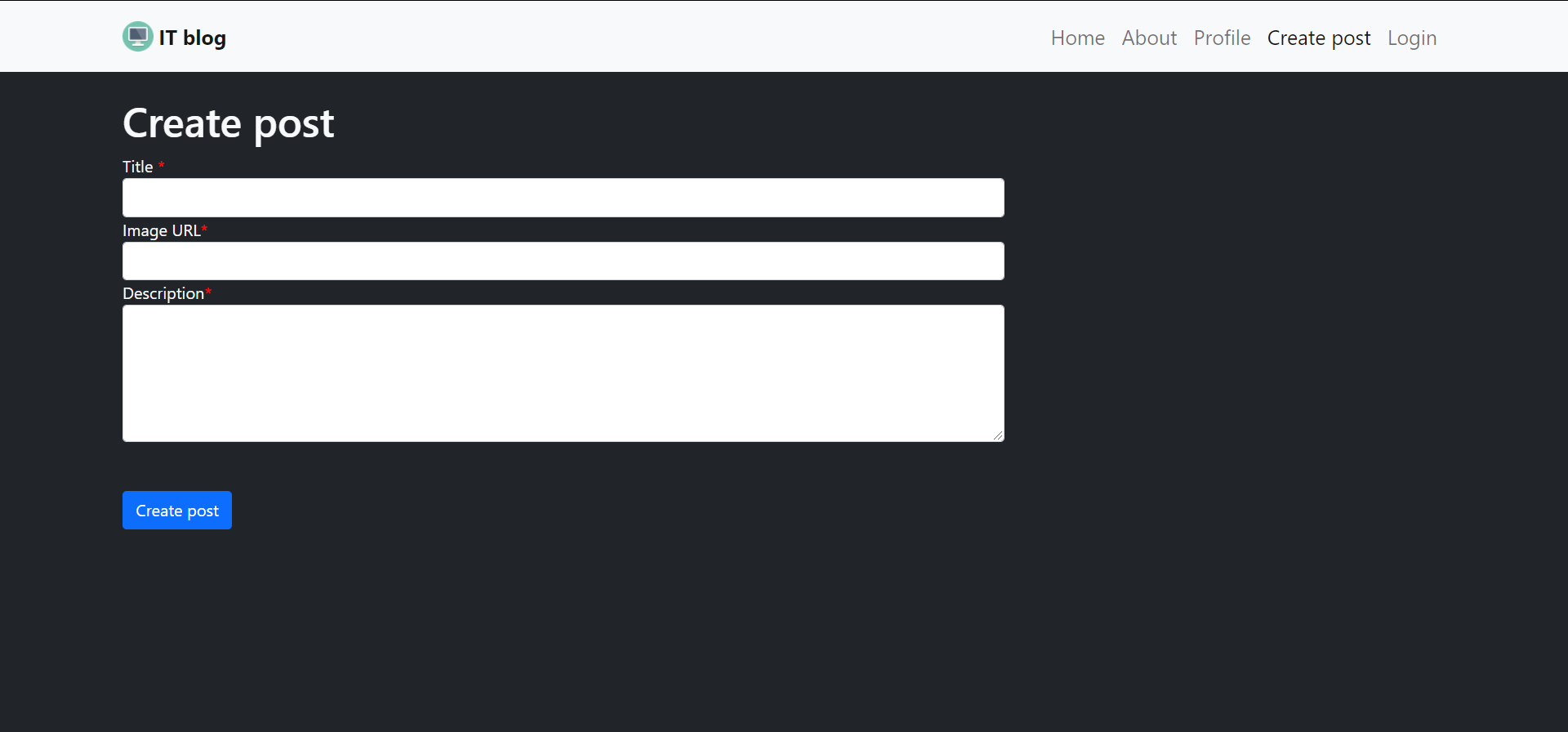
  
**UPDATE**Реалізовано можливість додавати коментарі до поста.

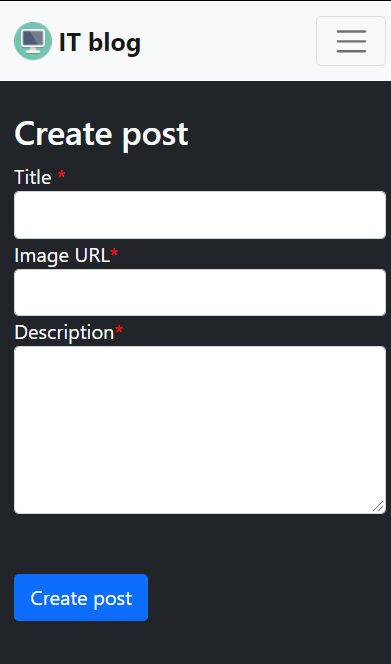


* **about.html**Сторінка з описом сайти та інформацією про розробника.  
    
  На телефонах та планшетах:

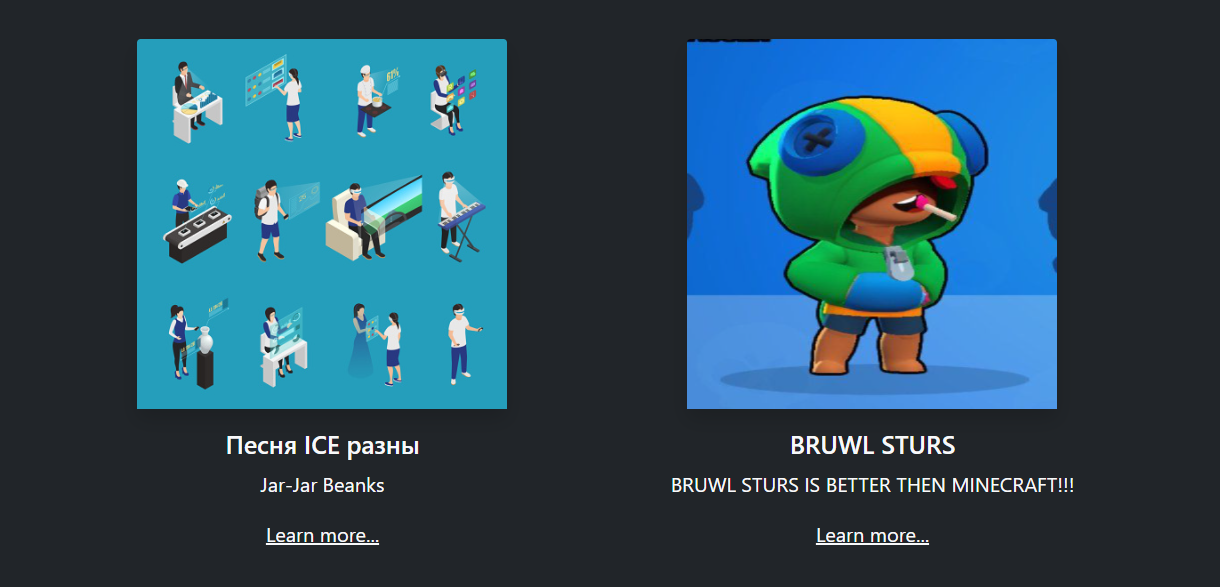


* **create-post.html**Сторінка з має звичайну форму з полями “title” та “description” для того щоб створювати пости.

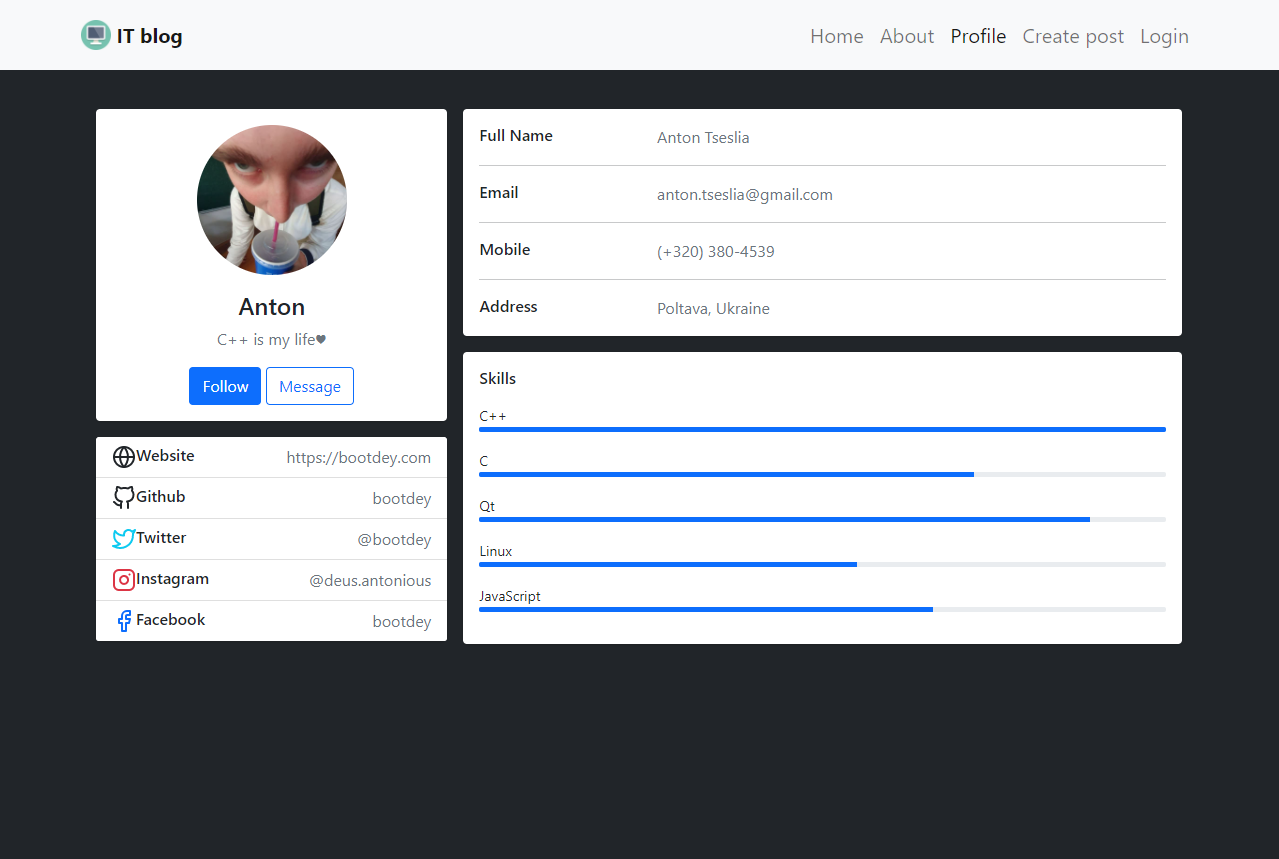
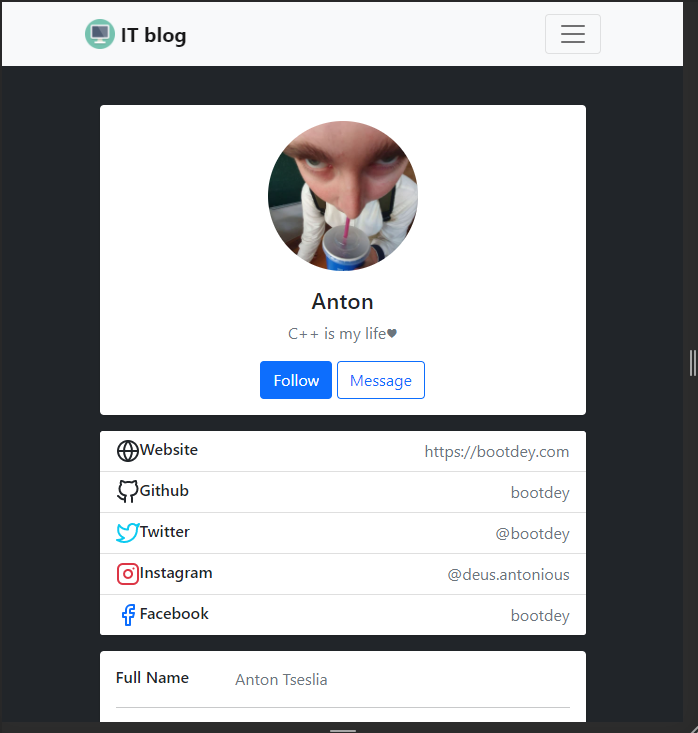
  
  
На телефонах та планшетах:

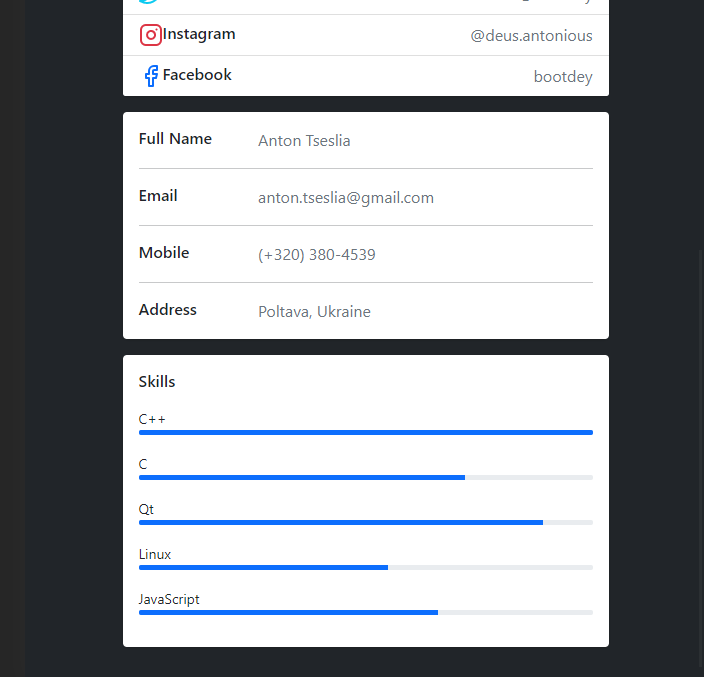


**UPDATE**Реалізовано можливість додавати коментарі до поста.

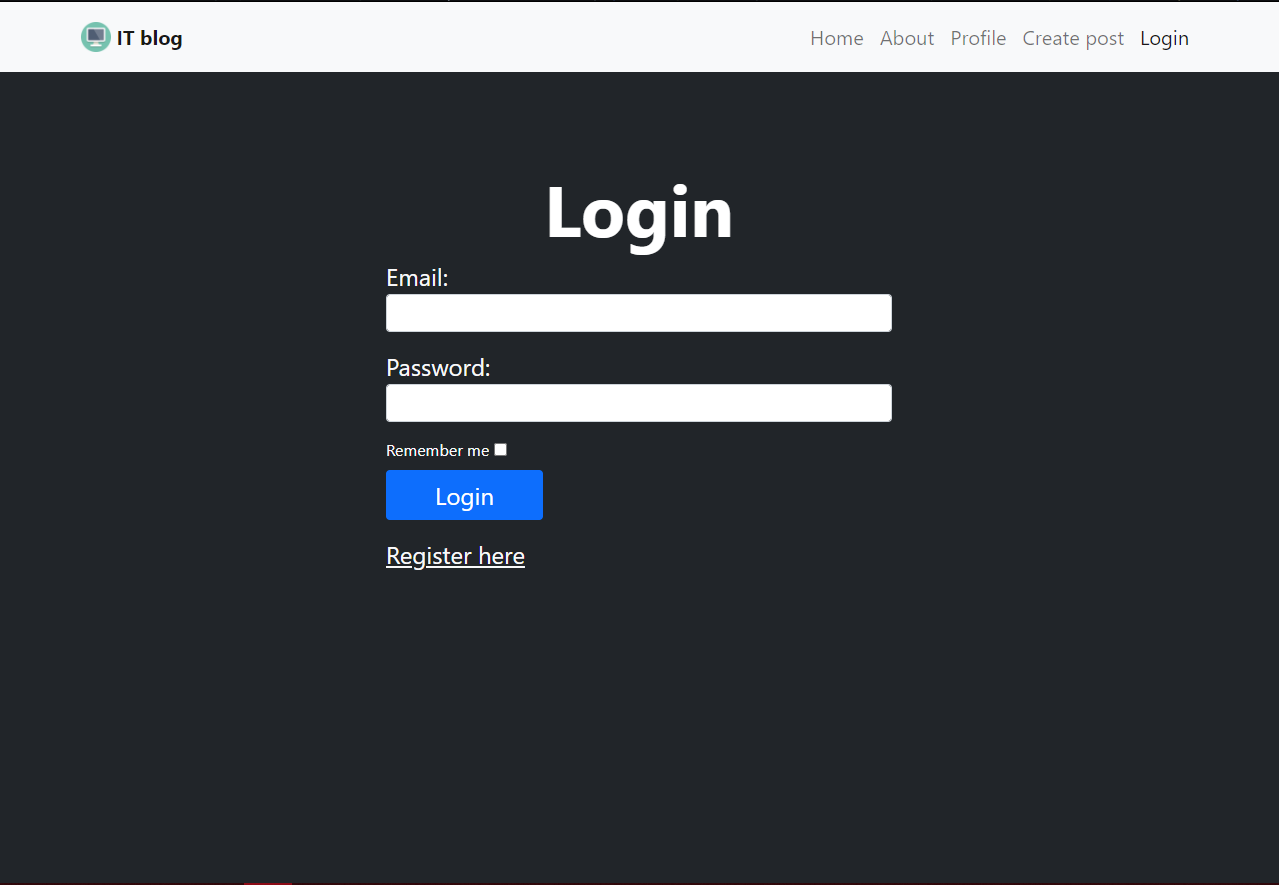


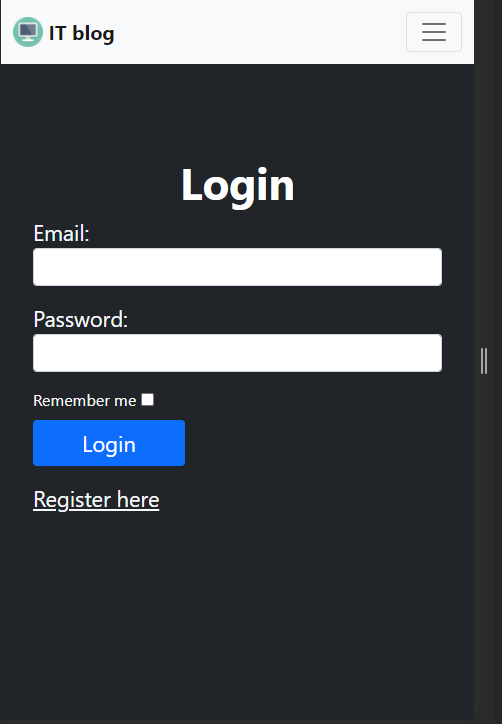
* **profile.html**Сторінка користувача.

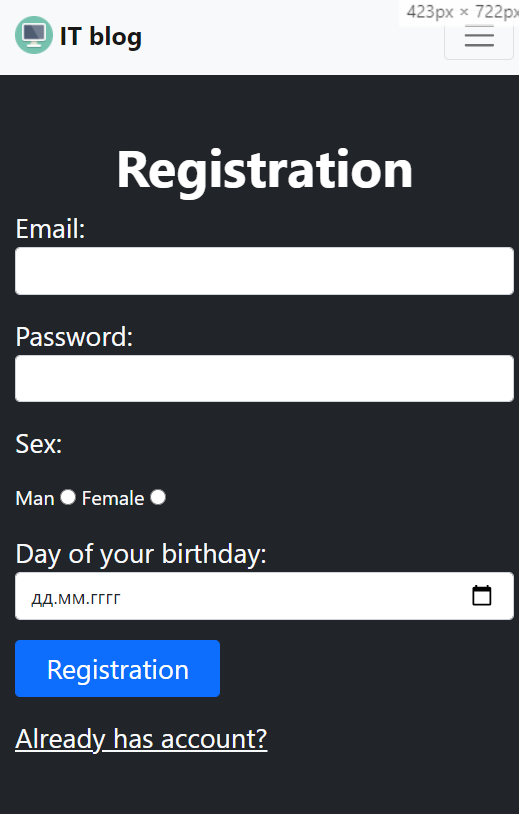
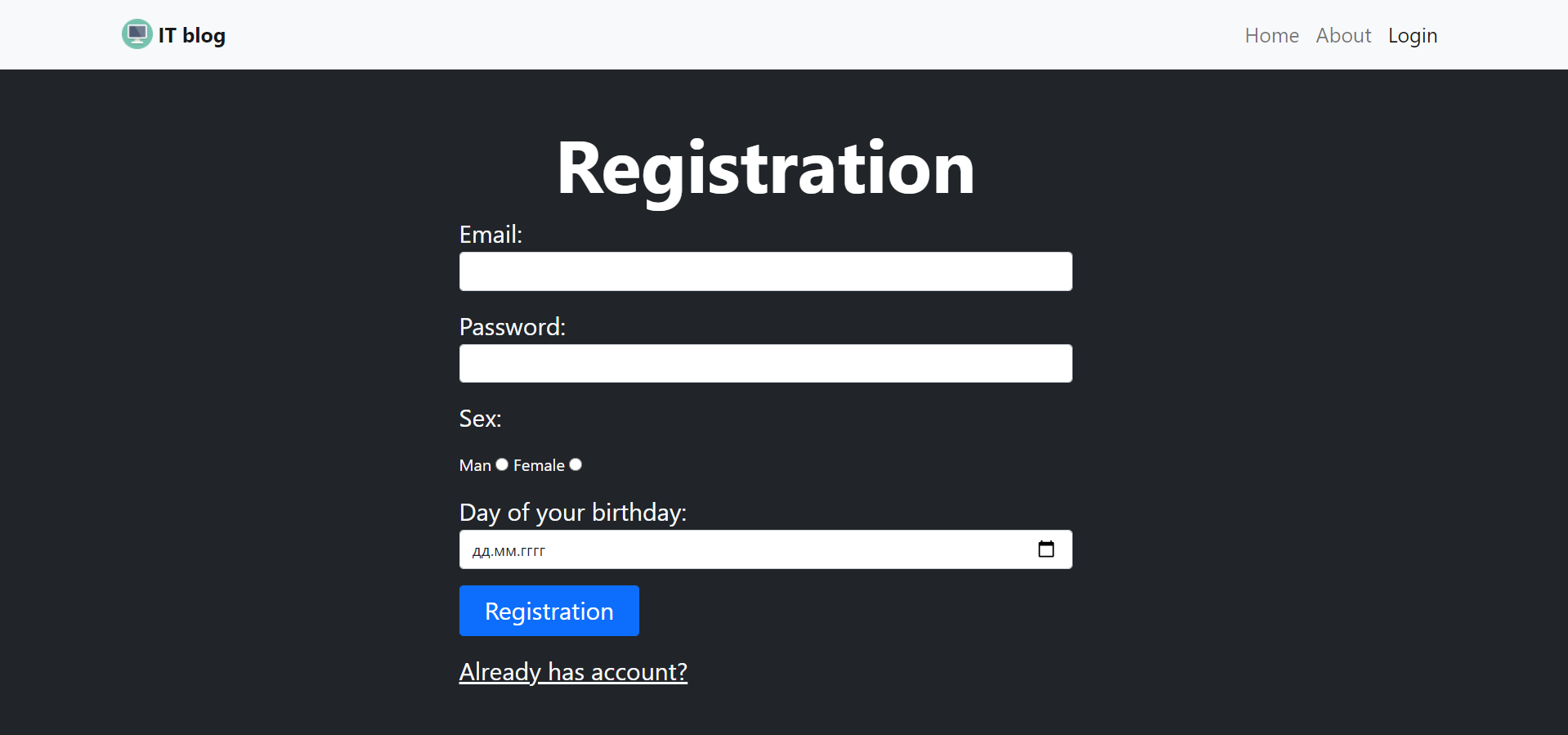
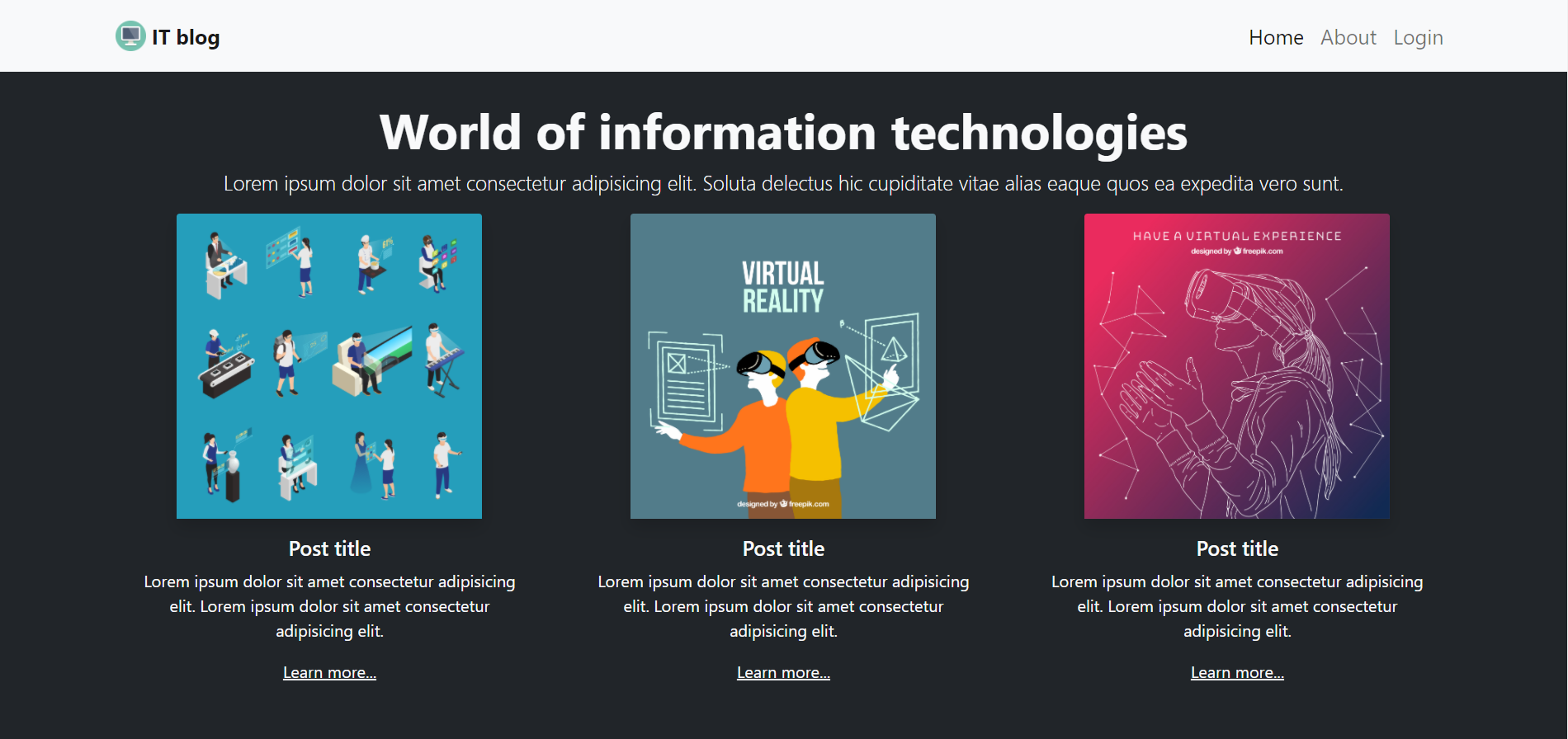
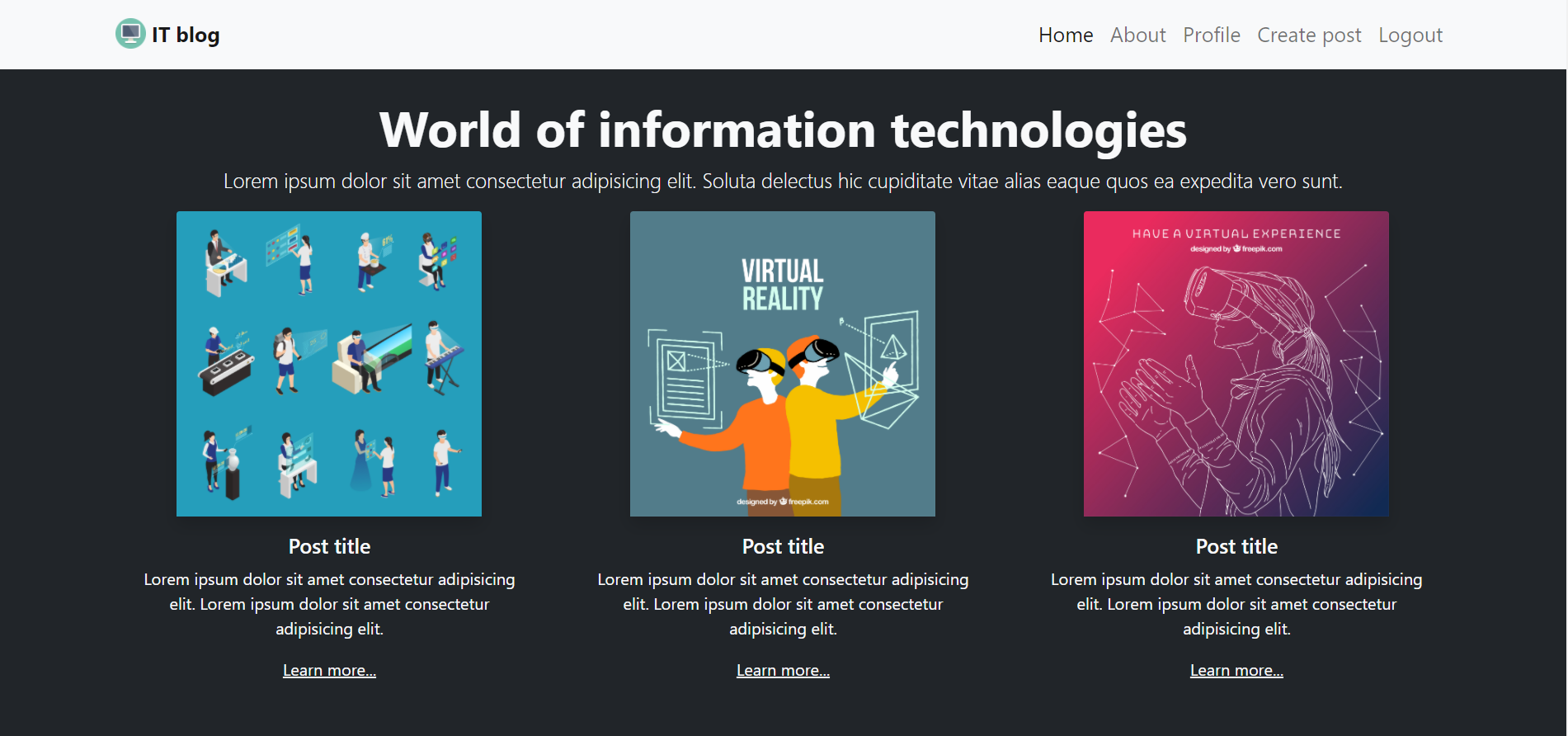
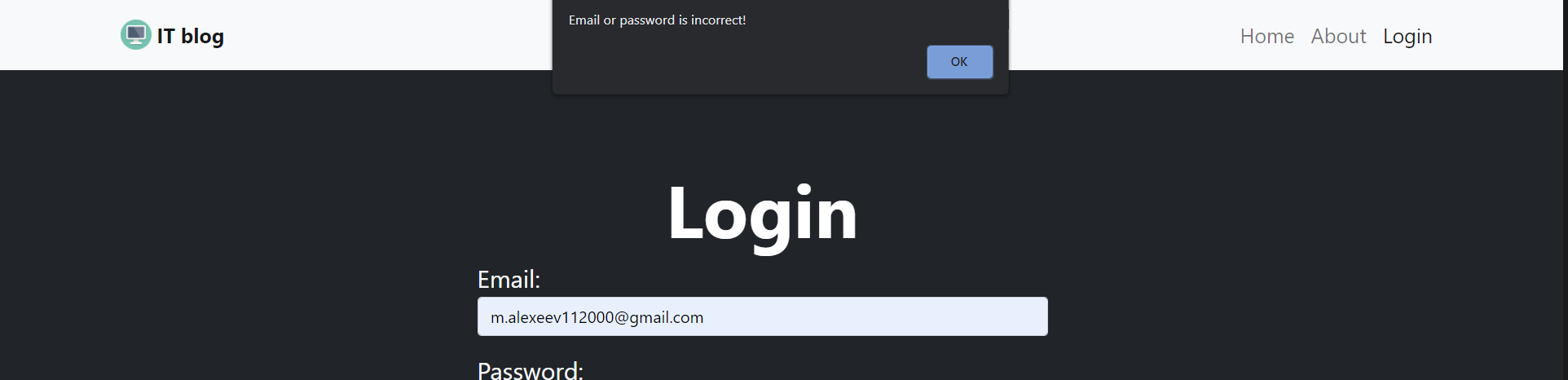
  
На планшетах та телефонах.  


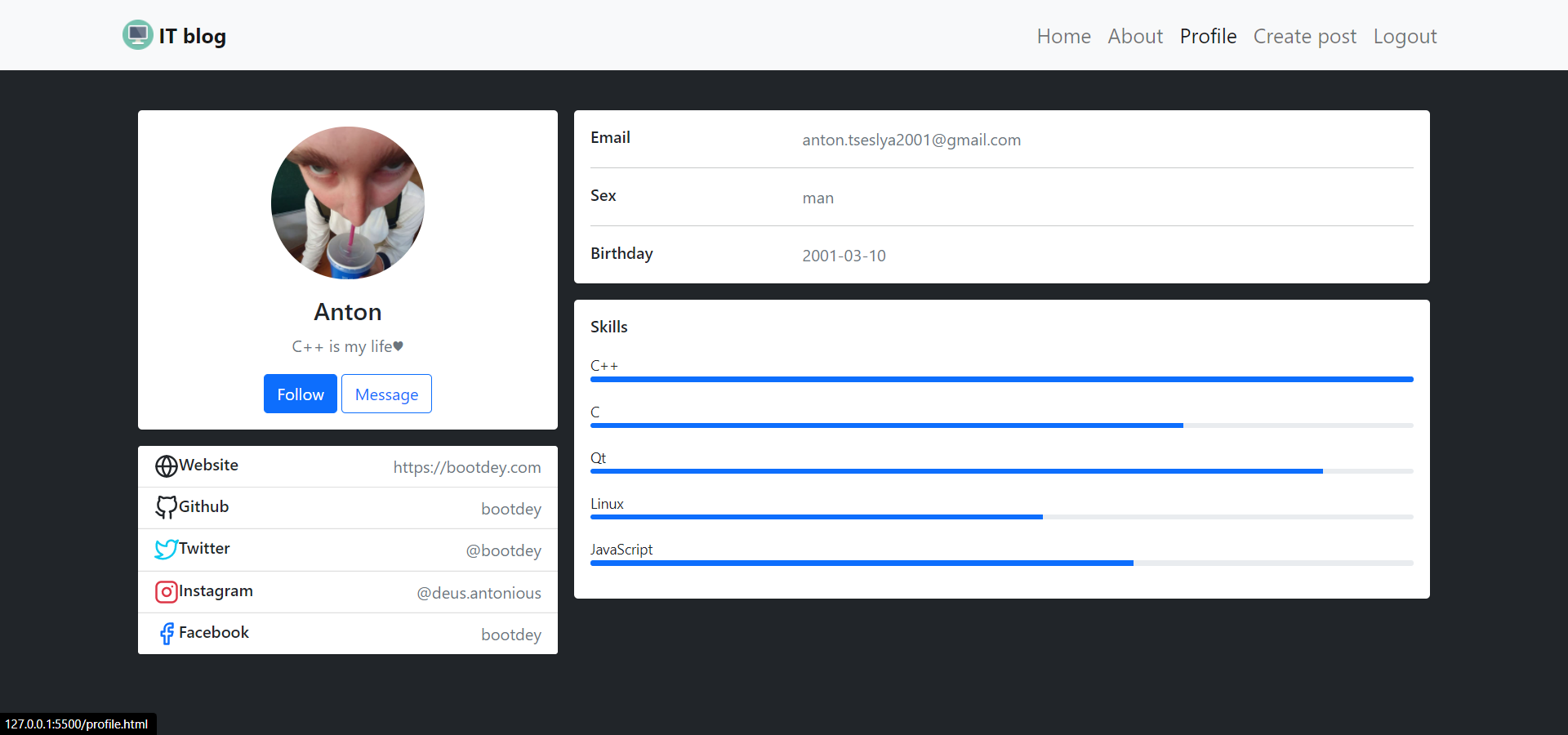


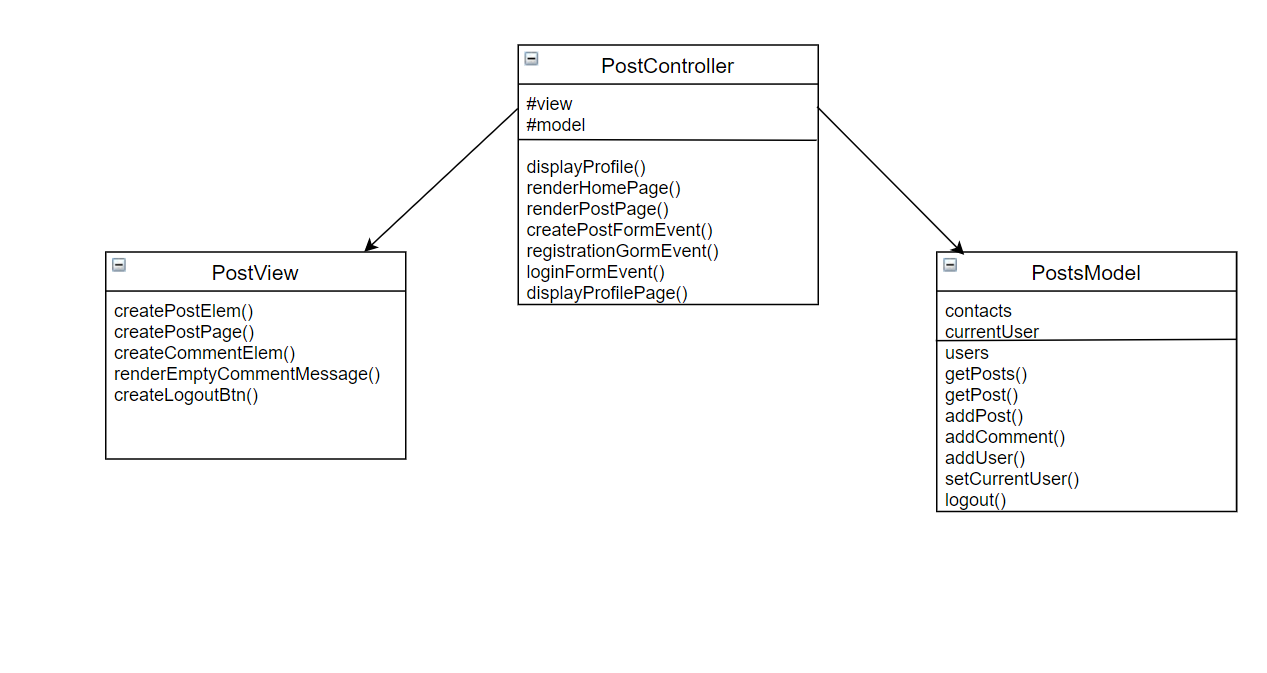
* **login.html**Сторінка для авторизації користувача. Якщо у користувача ще немає профілю, потрібно скористатися посиланням “Register now”.





* **registration.html**Сторінка для створення профілю користувача. Якщо користувач вже має профіль, потрібно скористатися посиланням “Already has account?”
* **UPDATE**Реалізована авторизація користувача. Дані про існуючих користувачів знаходиться в localStorage. Якщо користувач не авторизований, йому не відображаються сторінки “Profile” та “Create-post”.  
    
  Якщо користувач авторизувався, йому доступні всі сторінки, а посилання “Login” змінюється на “Logout”, при натисканні на яку, користувач виходить зі свого профілю.  
    
  Якщо користувач ввів невірні пароль та логін, йому висвітлюється повідомлення про невірні дані.  
  

Cторінка “Profile” відображає дані про користувача  


**Діаграма класів**

* **PostsView -** зберігає посилання на необхідні DOM елементи та методи для створення необхідних DOM компонентів.
* **PostsModel** - відповідає за роботу з даними та localStorage.
* **PostsController** - займається обробкою подій та слугує посередником між моделлю та контролером.

**data.js**

const data = [

{

title: 'Post title',

description:

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

img: '../assets/images/1.jpg',

comments: [

{

username: 'Username',

usernameImg: '../assets/images/avatar.png',

text:

'Lorem ipsum dolor sit amet, consectetur adipisicing elit sit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit adipisicing elitsit amet, consectetur adipisicing elitsit'

}

]

},

{

title: 'Post title',

description:

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

img: '../assets/images/2.jpg',

comments: [

{

username: 'Username',

usernameImg: '../assets/images/avatar.png',

text:

'Lorem ipsum dolor sit amet, consectetur adipisicing elit sit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit adipisicing elitsit amet, consectetur adipisicing elitsit'

}

]

},

{

title: 'Post title',

description:

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

img: '../assets/images/3.jpg',

comments: [

{

username: 'Username',

usernameImg: '../assets/images/avatar.png',

text:

'Lorem ipsum dolor sit amet, consectetur adipisicing elit sit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit adipisicing elitsit amet, consectetur adipisicing elitsit'

}

]

},

{

title: 'Post title',

description:

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

img: '../assets/images/4.jpg',

comments: [

{

username: 'Username',

usernameImg: '../assets/images/avatar.png',

text:

'Lorem ipsum dolor sit amet, consectetur adipisicing elit sit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit adipisicing elitsit amet, consectetur adipisicing elitsit'

}

]

},

{

title: 'Post title',

description:

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

img: '../assets/images/5.jpg',

comments: [

{

username: 'Username',

usernameImg: '../assets/images/avatar.png',

text:

'Lorem ipsum dolor sit amet, consectetur adipisicing elit sit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit adipisicing elitsit amet, consectetur adipisicing elitsit'

}

]

},

{

title: 'Post title',

description:

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

img: '../assets/images/6.jpg',

comments: [

{

username: 'Username',

usernameImg: '../assets/images/avatar.png',

text:

'Lorem ipsum dolor sit amet, consectetur adipisicing elit sit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit amet, consectetur adipisicing elitsit adipisicing elitsit amet, consectetur adipisicing elitsit'

}

]

}

];

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

localStorage.setItem('posts', JSON.stringify(data));

}

if (!localStorage.getItem('auth')) {

localStorage.setItem(

'auth',

JSON.stringify({

currentUser: null,

users: [{ email: 'test@test.com', password: '123', sex: 'man', birthday: '12:00' }]

})

);

}

**PostsController.js**

class PostsController {

constructor() {

this.displayProfile();

const url = window.location.href;

if (url.includes('/create-post')) {

this.createPostFormEvent();

} else if (url.includes('/post')) {

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

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

this.renderPostPage(id);

} else if (url.includes('/login')) {

this.loginFormEvent();

} else if (url.includes('/registration')) {

this.registrationFormEvent();

} else if (url.includes('/profile')) {

this.displayProfilePage();

} else if (url.includes('/index.html')) {

this.renderHomePage();

}

}

#model = new PostsModel();

#view = new PostsView();

displayProfile = () => {

if (!this.#model.currentUser) {

[...this.#view.profilePage].map(link => (link.style.display = 'none'));

} else {

const logoutBtn = this.#view.createLogoutBtn();

logoutBtn.onclick = () => {

this.#model.logout();

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

};

this.#view.loginLogout.innerHTML = '';

this.#view.loginLogout.appendChild(logoutBtn);

}

};

renderHomePage = () => {

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

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

const postElem = this.#view.createPostElem(id, post.title, post.img, post.description);

this.#view.postsContainer.appendChild(postElem);

});

};

renderPostPage = id => {

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

const postElem = this.#view.createPostPage(post.title, post.img, post.description);

this.#view.postContainer.appendChild(postElem);

if (!post.comments.length) {

this.#view.renderEmptyCommentsMessage();

} else {

post.comments.forEach(comment => {

const commentElem = this.#view.createCommentElem(comment.username, comment.usernameImg, comment.text);

this.#view.commentsContainer.appendChild(commentElem);

});

}

this.#view.commentForm.onsubmit = event => {

event.preventDefault();

const comment = new Comment('Anton', '../assets/images/avatar.jpg', this.#view.commentTextInput.value);

this.#model.addComment(id, comment);

location.reload();

};

};

createPostFormEvent = () => {

this.#view.postForm.onsubmit = () => {

const post = new Post(

this.#view.titleInput.value,

this.#view.imgUrlInput.value,

this.#view.descriptionInput.value,

[]

);

this.#model.addPost(post);

};

};

registrationFormEvent = () => {

this.#view.registrationForm.onsubmit = () => {

const sex = [...this.#view.sexRadioBtns].find(btn => btn.checked).value;

const newUser = {

email: this.#view.emailInput.value,

password: this.#view.passwordInput.value,

sex,

birthday: this.#view.dateInputs.value

};

this.#model.addUser(newUser);

};

};

loginFormEvent = () => {

this.#view.loginForm.onsubmit = () => {

const user = {

email: this.#view.emailInput.value,

password: this.#view.passwordInput.value

};

const isLogin = this.#model.setCurrentUser(user);

!isLogin ? alert('Email or password is incorrect!') : null;

};

};

displayProfilePage = () => {

const { email, sex, birthday } = this.#model.currentUser;

this.#view.profileEmail.innerHTML = email;

this.#view.profileSex.innerHTML = sex;

this.#view.profileBirthday.innerHTML = birthday;

};

}

const app = new PostsController();

**PostsModel.js**

class Post {

constructor(title, img, description, comments) {

this.title = title;

this.img = img;

this.description = description;

this.comments = comments;

}

}

class Comment {

constructor(username, usernameImg, text) {

this.username = username;

this.usernameImg = usernameImg;

this.text = text;

}

}

class User {

constructor({ email, password, sex, birthday }) {

this.email = email;

this.password = password;

this.sex = sex;

this.birthday = birthday;

}

}

class PostsModel {

constructor() {

this.posts = JSON.parse(localStorage.getItem('posts')).map(

post => new Post(post.title, post.img, post.description, post.comments)

);

const { currentUser, users } = JSON.parse(localStorage.getItem('auth'));

this.currentUser = currentUser ? new User(currentUser) : null;

this.users = users.map(user => new User(user));

}

getPosts = () => {

return this.posts;

};

getPost = id => {

return this.posts[id];

};

addPost = ({ title, img, description, comments }) => {

const post = new Post(title, img, description, comments);

this.posts.push(post);

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

};

addComment = (id, { username, usernameImg, text }) => {

const comment = new Comment(username, usernameImg, text);

this.posts[id].comments.push(comment);

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

};

addUser = user => {

const newUser = new User(user);

this.users.push(newUser);

this.currentUser = newUser;

localStorage.setItem('auth', JSON.stringify({ currentUser: this.currentUser, users: this.users }));

};

setCurrentUser = user => {

const existingUser = this.users.find(u => user.email === u.email && user.password === u.password);

if (existingUser) {

this.currentUser = new User(existingUser);

localStorage.setItem('auth', JSON.stringify({ currentUser: this.currentUser, users: this.users }));

return true;

} else return false;

};

logout = () => {

this.currentUser = null;

localStorage.setItem('auth', JSON.stringify({ currentUser: this.currentUser, users: this.users }));

};

}

**PostsView.js**

class PostsView {

constructor() {

this.postsContainer = document.getElementById('posts-container');

this.commentsContainer = document.getElementById('comments-container');

this.postContainer = document.getElementById('post-container');

this.commentForm = document.getElementById('comment-form');

this.postForm = document.getElementById('post-form');

this.titleInput = document.getElementById('title');

this.imgUrlInput = document.getElementById('image');

this.descriptionInput = document.getElementById('description');

this.commentTextInput = document.getElementById('text');

this.loginForm = document.getElementById('login-form');

this.registrationForm = document.getElementById('registration-form');

this.emailInput = document.getElementById('email');

this.passwordInput = document.getElementById('password');

this.sexRadioBtns = document.getElementsByName('sex');

this.dateInputs = document.getElementById('birthday');

this.profilePage = document.querySelectorAll('.profile');

this.loginLogout = document.getElementById('login-logout');

this.profileEmail = document.getElementById('profile-email');

this.profileSex = document.getElementById('profile-sex');

this.profileBirthday = document.getElementById('profile-birthday');

}

createPostElem = (id, title, img, description) => {

const post = document.createElement('div');

post.classList.add('col-12', 'col-md-6', 'col-lg-4', 'mb-5');

post.innerHTML = `

<img style="height: 296px; width: 296px;"class="card-img-top shadow" src="${img}" alt="Card image cap" />

<div class="card-body">

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

<p class="card-text">

${description}

</p>

<a href="./post.html?post=${id}" class="card-link text-light">

Learn more...

</a>

</div>`;

return post;

};

createPostPage = (title, img, description) => {

const post = document.createElement('div');

post.classList.add('row', 'flex-column', 'flex-lg-row', 'bg-dark');

post.innerHTML = `

<div class="col-12 col-lg-7 jumbotron order-2 text-light">

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

<p class="lead">

${description}

</p>

</div>

<img src="${img}" class="col-12 col-lg-5 img-fluid order-1" alt="Responsive image" />`;

return post;

};

createCommentElem = (username, usernameImg, text) => {

const comment = document.createElement('div');

comment.classList.add('rounded', 'row', 'justify-content-md-center', 'bg-light', 'text-dark', 'mb-3');

comment.innerHTML = `

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

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

</div>

<div class="col-md-11 col-9 text-start">

<div class="row">

<div class="col">

<h4>${username}</h4>

</div>

</div>

<div class="row">

<div class="col">

<p class="col-11 text-wrap">

${text}

</p>

</div>

</div>

</div>`;

return comment;

};

renderEmptyCommentsMessage = () => {

const empty = document.createElement('h3');

empty.innerHTML = 'No comments:(';

this.commentsContainer.appendChild(empty);

};

createLogoutBtn = () => {

const elem = document.createElement('a');

elem.classList.add('nav-link');

elem.innerHTML = 'Logout';

return elem;

};

}