



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування та спеціалізованих комп'ютерних систем

Лабораторна робота №3

з дисципліни

«Web-дизайн»

тема: **“Розробка Web-додатка засобами Javascript/VueJS”**

Виконав: студент III курсу

ФПМ групи КВ-02

Телеганенко Богдан Валерійович

Перевірив: Петрашенко А.

Київ-2023

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

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

Інструменти розробки: Javascript, VueJS, NodeJS, Express/

Зауваження: студент може обрати інший серверний фреймворк (наприклад, Django).

Лінки:

-GitHub rep: <https://github.com/nothing549/web-app/tree/web-3>

-Telegram: <https://t.me/nothing549>

Опис додатка

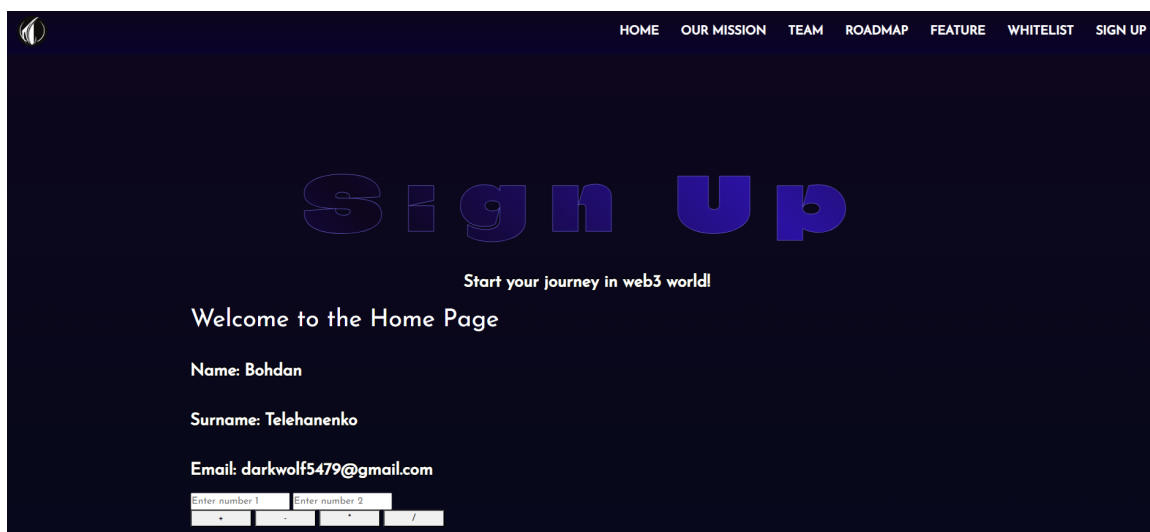
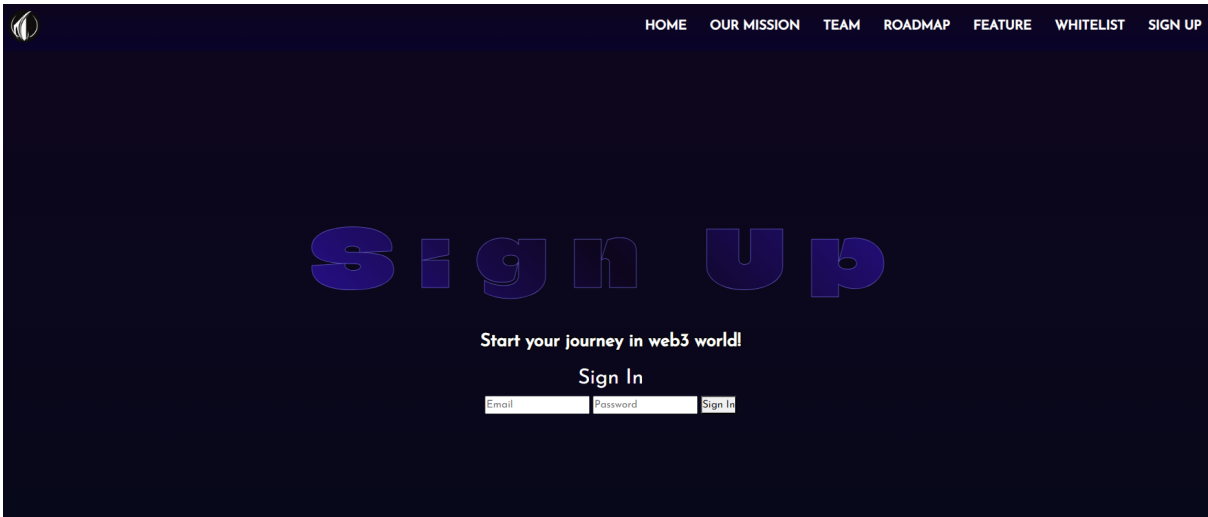
Назва: Doomies.

Опис: проектування адаптивного лендінг пейдж для бізнес проекту.

В свою чергу сторінка є візитною карткою компанії, на якій можна знайти інформацію про рід діяльності проекту.

Перероблена механіка входу на сторінку. Зараз потребуються серверні виклики. Після реєстрації дані відправляються на сервер, де зберігаються у базі даних MongoDB. Якщо юзер захочу увійти в систему - дані будуть передаватися до серверу - буде здійснюватися перевірка правильності даних і у відповідь на сабміт юзер буде переадресований на домашню сторінку.

Скріни



Sign Up

Start your journey in web3 world!

Welcome to the Home Page

Name: Bohdan

Surname: Telehanenko

Email: darkwolf5479@gmail.com

10		20	
+	-	*	/

Result: -10

Sign Up

Start your journey in web3 world!

Welcome to the Home Page

Name: Bohdan

Surname: Telehanenko

Email: darkwolf5479@gmail.com

10		20	
+	-	*	/

Result: 200

Sign Up

Start your journey in web3 world!

Welcome to the Home Page

Name: Bohdan

Surname: Telehanenko

Email: darkwolf5479@gmail.com

10	20
+	-
*	/

Result: 0.5

Код

```
// sign.js
```

```
new Vue({
  el: '#app',
  data: {
    showSignIn: true,
    signInEmail: "",
    signInPassword: "",
    signUpName: "",
    signUpSurname: "",
    signUpEmail: "",
    signUpPassword: "",
    signInError: "",
    signUpError: ""
  },
  methods: {
    signIn: function() {
      // Perform sign-in logic
      const formData = {
        email: this.signInEmail,
```

```
password: this.signInPassword
};

fetch('/signin', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(formData)
})
.then(response => response.json())
.then(data => {
  if (data.error) {
    this.signInError = data.error;
  } else {
    // Redirect to the user account page or perform any other actions
    console.log('Sign in successful');
    console.log('User:', data.user);
  }
})
.catch(error => {
  console.error('Error during sign in:', error);
});
},
signUp: function() {
  // Perform sign-up logic
  const formData = {
    name: this.signUpName,
    surname: this.signUpSurname,
    email: this.signUpEmail,
    password: this.signUpPassword
  };
}
```

```

fetch('/signup', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(formData)
})
.then(response => response.json())
.then(data => {
  if (data.error) {
    this.signUpError = data.error;
  } else {
    // Redirect to the user account page or perform any other actions
    console.log('Sign up successful');
    console.log('User:', data.user);
  }
})
.catch(error => {
  console.error('Error during sign up:', error);
});
}
}
});

// node.js
const express = require('express');
const mongoose = require('mongoose');
const app = express();
const port = 3000;

// Connect to MongoDB

```

```
mongoose.connect('mongodb://localhost/mydatabase', { useNewUrlParser: true,  
useUnifiedTopology: true })
```

```
.then(() => {  
  console.log('Connected to MongoDB');  
})  
.catch((error) => {  
  console.error('Error connecting to MongoDB:', error);  
});
```

```
// Define User schema
```

```
const userSchema = new mongoose.Schema({  
  name: String,  
  surname: String,  
  email: String,  
  password: String  
});
```

```
const User = mongoose.model('User', userSchema);
```

```
app.use(express.urlencoded({ extended: true }));  
app.use(express.json());
```

```
// Sign In API
```

```
app.post('/signin', (req, res) => {  
  const email = req.body.email;  
  const password = req.body.password;
```

```
// Perform sign-in logic
```

```
User.findOne({ email: email, password: password }, (error, user) => {  
  if (error) {  
    console.error('Error during sign in:', error);  
    res.status(500).json({ error: 'Internal Server Error' });
```



```
    } else if (!user) {  
      res.status(401).json({ error: 'Invalid credentials' });  
    } else {  
      res.json({ message: 'Sign in successful', user: user });  
    }  
  });  
});
```

// Sign Up API

```
app.post('/signup', (req, res) => {  
  const name = req.body.name;  
  const surname = req.body.surname;  
  const email = req.body.email;  
  const password = req.body.password;  
  
  // Perform sign-up logic  
  const newUser = new User({ name: name, surname: surname, email: email, password:  
password });  
  newUser.save((error, savedUser) => {  
    if (error) {  
      console.error('Error during sign up:', error);  
      res.status(500).json({ error: 'Internal Server Error' });  
    } else {  
      res.json({ message: 'Sign up successful', user: savedUser });  
    }  
  });  
});  
  
app.listen(port, 'localhost', () => {  
  console.log(`Server running at http://localhost:${port}`);  
});
```

