

ЛАБОРАТОРНА РОБОТА

«Використання Git та GitHub у React / Node.js проєктах»



Мета роботи

Навчитися:

- створювати React / Node.js проєкти;
- використовувати Git для контролю версій;
- працювати з гілками;
- публікувати проєкти на GitHub;
- правильно використовувати .gitignore у JS-проєктах.



Обладнання та ПЗ

- ОС: Windows
- Git
- GitHub (акаунт)
- Node.js (LTS)
- npm
- VS Code
- Браузер



Завдання

1. Створити React або Node.js проєкт
2. Ініціалізувати Git-репозиторій
3. Налаштувати .gitignore
4. Створити репозиторій на GitHub
5. Створити гілку для нової функції
6. Додати новий компонент / маршрут
7. Злити гілку з main
8. Опублікувати код на GitHub



Хід роботи

◆ ВАРІАНТ А: React (frontend)

Крок 1. Створення React-проєкту

```
npx create-react-app react-git-lab  
cd react-git-lab
```

Запуск:

```
npm start
```



Скріншот 1: стартова сторінка React

Крок 2. Ініціалізація Git

```
git init
```

```
git add .
```

```
git commit -m "Initial React commit"
```

Крок 3. .gitignore

У React він **створюється автоматично**, але важливо перевірити:
node_modules
.env
build

Крок 4. Створення репозиторію на GitHub

```
git branch -M main  
git remote add origin https://github.com/USERNAME/react-git-lab.git  
git push -u origin main
```

 **Скріншот 2:** репозиторій на GitHub

Крок 5. Створення гілки

```
git checkout -b feature-header
```

Крок 6. Робота в гілці (React)

Створити компонент src/Header.js:

```
function Header() {  
  return <h2>My Header Component</h2>;  
}
```

export default Header;

Підключити в App.js:

```
import Header from './Header';
```

```
function App() {  
  return (  
    <div>  
      <Header />  
    </div>  
  );  
}
```

```
export default App;
```

Commit:

```
git add .
```

```
git commit -m "Add Header component"
```

 **Скріншот 3:** компонент у браузері

Крок 7. Злиття гілки

```
git switch main
```

```
git merge feature-header
```

```
git push
```

 **Скріншот 4:** оновлений код на GitHub

◆ ВАРІАНТ В: Node.js (backend)

Крок 1. Створення Node.js проєкту

```
mkdir node-git-lab
```

```
cd node-git-lab
```

```
npm init -y
```

Створити index.js:

```
console.log("Hello Node.js + GitHub");
```

Запуск:

```
node index.js
```

 **Скріншот 1:** результат у терміналі

Крок 2. Ініціалізація Git

```
git init
```

```
git add .
```

```
git commit -m "Initial Node.js commit"
```

Крок 3. .gitignore

```
node_modules
```

```
.env
```

```
*.log
```

Крок 4. GitHub

```
git branch -M main
```

```
git remote add origin https://github.com/USERNAME/node-git-lab.git
```

```
git push -u origin main
```

Крок 5. Гілка для нової функції

```
git checkout -b feature-route
```

Крок 6. Робота в гілці (Node.js)

Встановити Express:

```
npm install express
```

Оновити index.js:

```
const express = require('express');
```

```
const app = express();
```

```
app.get('/', (req, res) => {  
  res.send('Hello from feature branch');  
});
```

```
app.listen(3000, () => {  
  console.log('Server running');
```

```
});  
Commit:  
git add .  
git commit -m "Add Express server"
```

Крок 7. Злиття гілки

```
git switch main  
git merge feature-route  
git push
```

✓ Результат роботи

- створено React / Node.js проєкт;
- використано Git для керування версіями;
- створено та злино гілки;
- проєкт опубліковано на GitHub.

🔗 Контрольні питання

1. Навіщо .gitignore у Node.js / React?
2. Чому не можна додавати node_modules у Git?
3. Для чого створюють feature-гілки?
4. Чим відрізняється npm install від git clone?
5. Що таке Pull Request?