

# CREATE-PROJECT-CLI

## Command Line Project Generator

Assignment Submission containing all the screenshots

Chandan Mishra

Date: 20/01/ 2026

Command Execution creation of Project and Backend Initialization

```
PS C:\Users\mishr\Desktop> create-proj project2 -t fullstack
Creating fullstack project: project2...
✓ Project structure created
✓ Project "project2" created successfully!

Next steps:
  cd project2
  Start coding!
PS C:\Users\mishr\Desktop> cd project2
PS C:\Users\mishr\Desktop\project2> cd backend
PS C:\Users\mishr\Desktop\project2\backend> npm run dev
> project2-backend@1.0.0 dev
> node server.js

🚀 project2 Backend Server Started!
📡 Server running on http://localhost:5000
📝 API Endpoints:
  GET  http://localhost:5000/api/users
  POST http://localhost:5000/api/users
  GET  http://localhost:5000/api/status

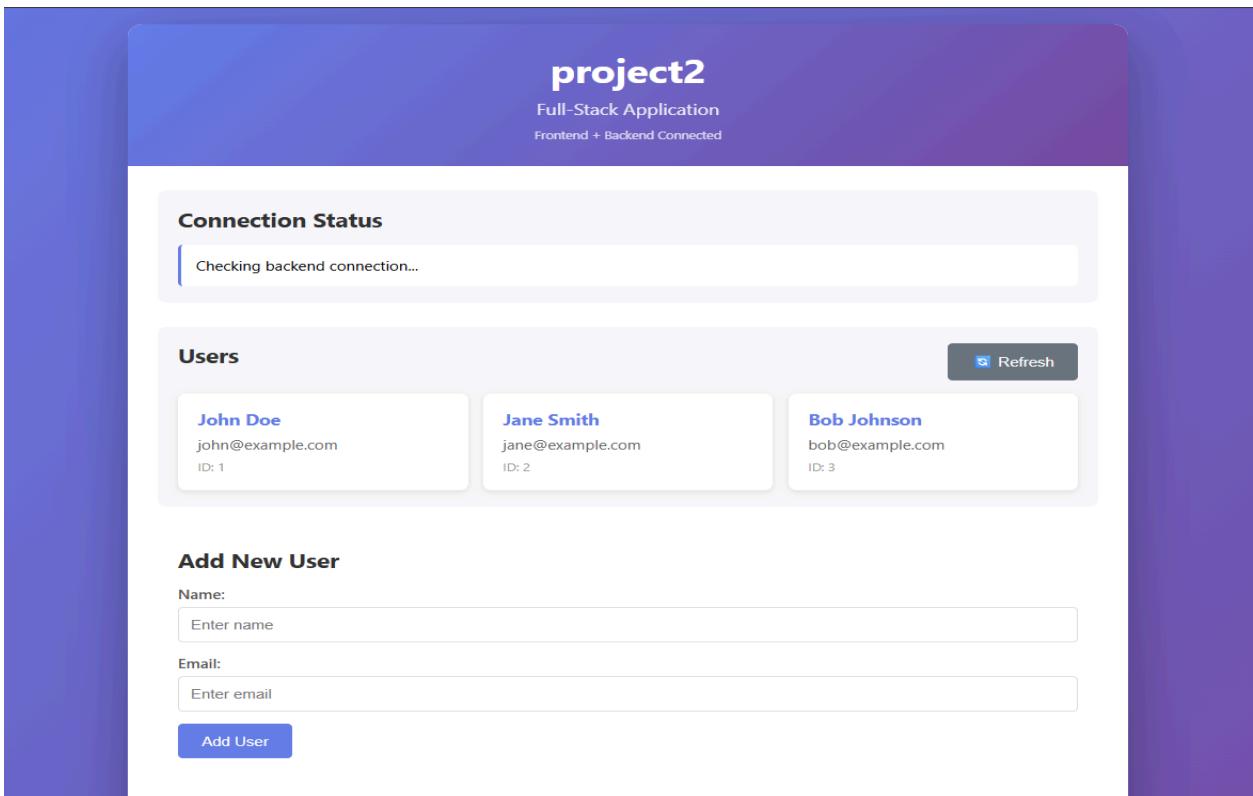
✓ Ready to receive requests from frontend!
```

Frontend Initialization

```
PS C:\Users\mishr\Desktop> cd project2
PS C:\Users\mishr\Desktop\project2> cd frontend
PS C:\Users\mishr\Desktop\project2\frontend> npm run dev
> project2-frontend@1.0.0 dev
> node server.js

🌐 project2 Frontend Server Started!
💻 Frontend running on http://localhost:3000
✓ Open http://localhost:3000 in your browser!
```

## Website screenshot where frontend is completely connected with backend



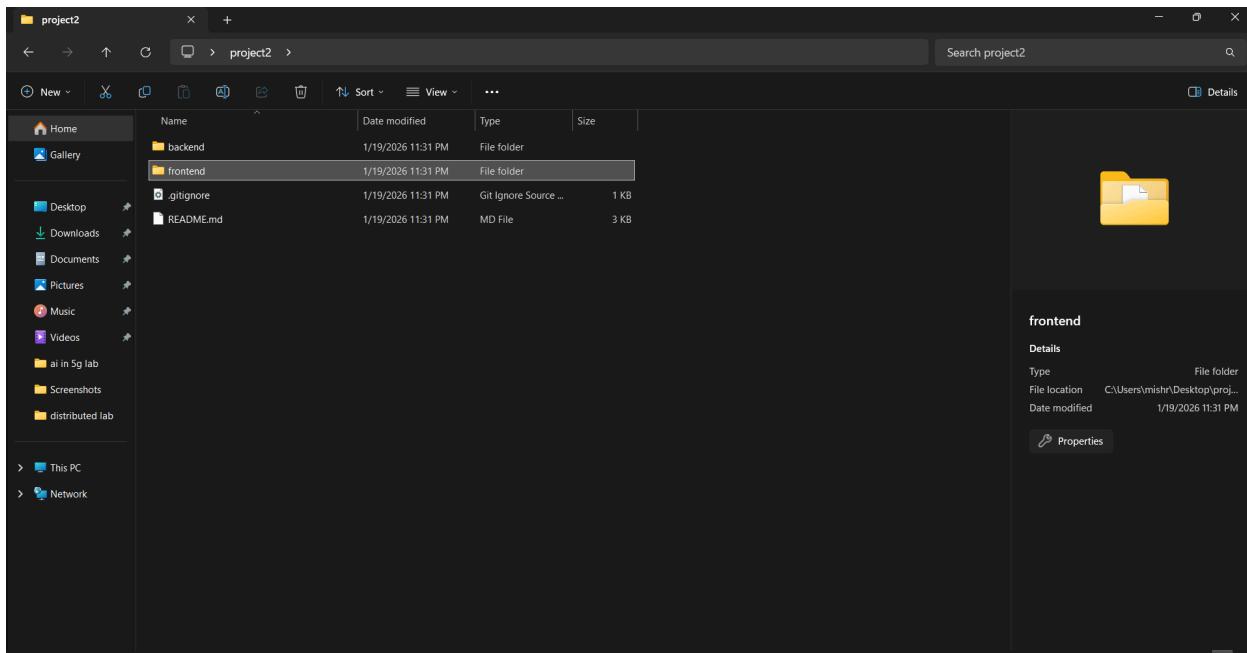
## Code folder and file structure

A screenshot of a code editor (Visual Studio Code) showing the file structure of a project named "CREATE-PROJECT-CLI". The "EXPLORER" sidebar on the left lists files and folders: README.md, LICENSE, test.js, argumentParser.js, logger.js, templateGenerator.js, .gitignore, index.js, package.json, README.md, and test.js. The main editor area displays the content of "index.js". The code defines a class "ProjectCLI" with a constructor that initializes a logger, parser, and generator. It includes a run method that handles command-line arguments, parses them, validates configuration, and creates a project. Success and info logs are used to output messages.

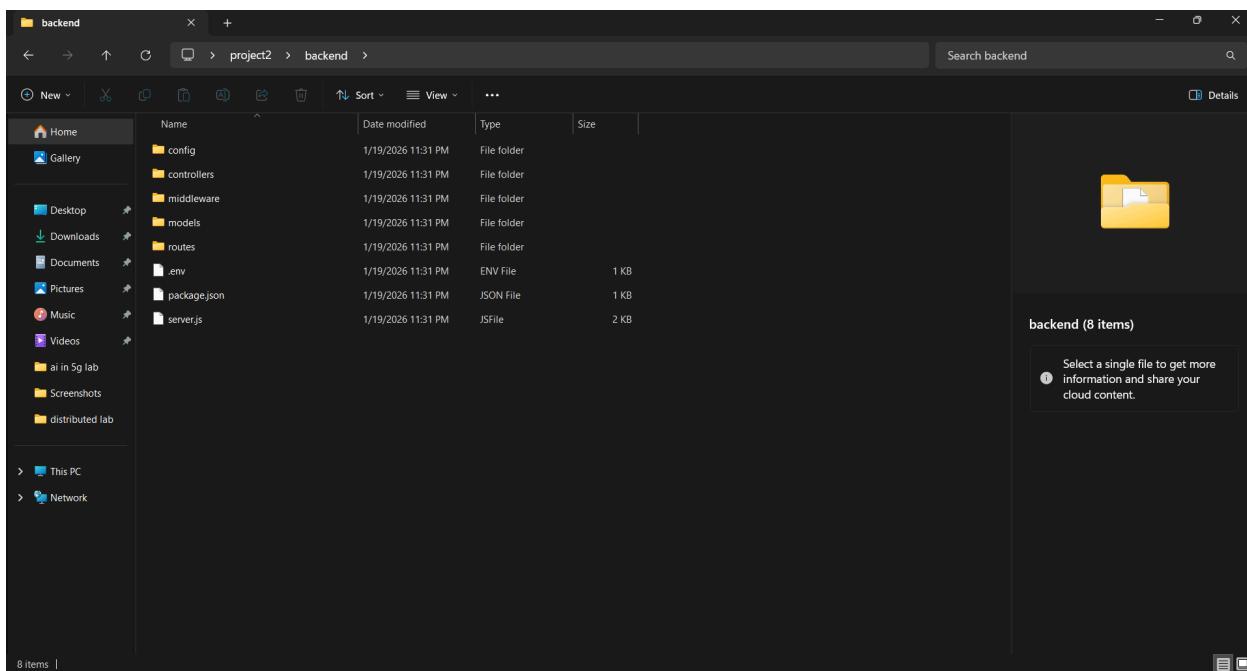
```
1  const fs = require('fs');
2  const path = require('path');
3  const { execSync } = require('child_process');
4  const ArgumentParser = require('../utils/argumentParser');
5  const TemplateGenerator = require('../utils/templateGenerator');
6  const logger = require('../utils/logger');

7  class ProjectCLI {
8    constructor() {
9      this.logger = new Logger();
10     this.parser = new ArgumentParser();
11     this.generator = new TemplateGenerator();
12   }
13
14   run() {
15     try {
16       const args = process.argv.slice(2);
17
18       // Show help if no arguments provided
19       if (args.length === 0 || args.includes('--help') || args.includes('-h')) {
20         this.showHelp();
21         return;
22       }
23
24       // Parse command-line arguments
25       const config = this.parser.parse(args);
26
27       // Validate configuration
28       this.validateConfig(config);
29
30       // Create the project
31       this.createProject(config);
32
33       this.logger.success(`\nProject "${config.projectName}" created successfully!`);
34       this.logger.info(`\nNext steps:`);
35     }
36   }
37 }
```

## Folder and File structure of Project which we created with the help of CLI tool



## Backend folder structure



## Frontend folder structure

