

Portfolio Project

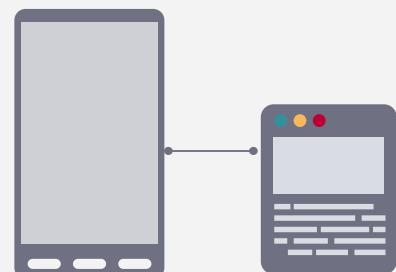
Shorooq Abu Zaid
202257840
F04





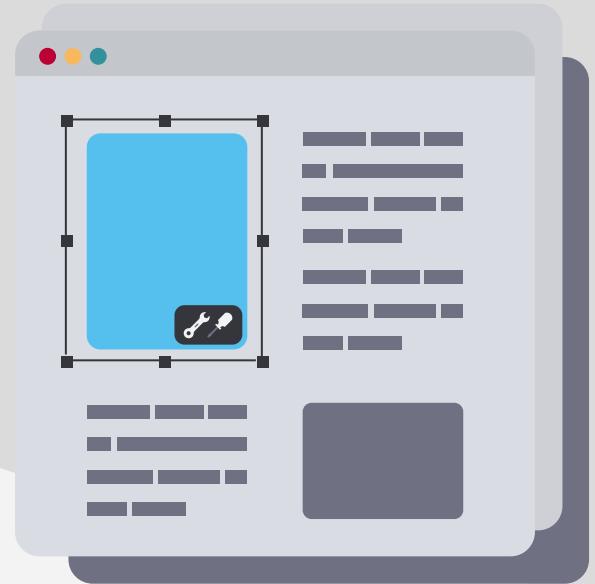
Introduction

This presentation details the development of a polished, responsive *personal portfolio web application* built with React and custom CSS. It highlights the main objectives, technical components, and key features designed to showcase real-time GitHub data, interactive project displays, and a dynamic contact form. The goal is to demonstrate professional web development skills while creating a lasting personal site.



01

Project Overview and Motivation



Objective and Core Features

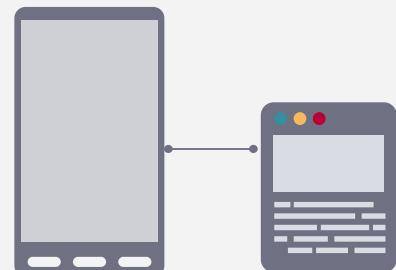
The project aims to deliver a **responsive, modern web portfolio** with seamless navigation and appealing design. Core features include a homepage with live GitHub integration, searchable projects with filtering by level and keywords, an animated contact form, and a light/dark theme toggle. The app uses React, Vite, and custom CSS for a robust user experience on all devices.





Technical Goals and Learning Outcomes

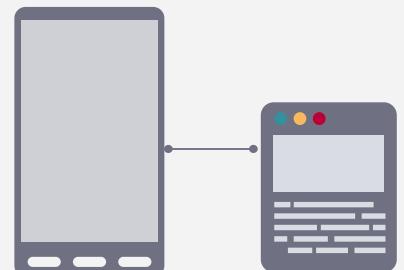
Key technical goals include mastering React routing, state management via Context API, and API integration using GitHub's REST API. The project reinforces skills in responsive design, animation with Framer Motion, and deployment with Netlify. It also emphasizes *responsible AI usage* for debugging and documentation, preparing for professional frontend development challenges.





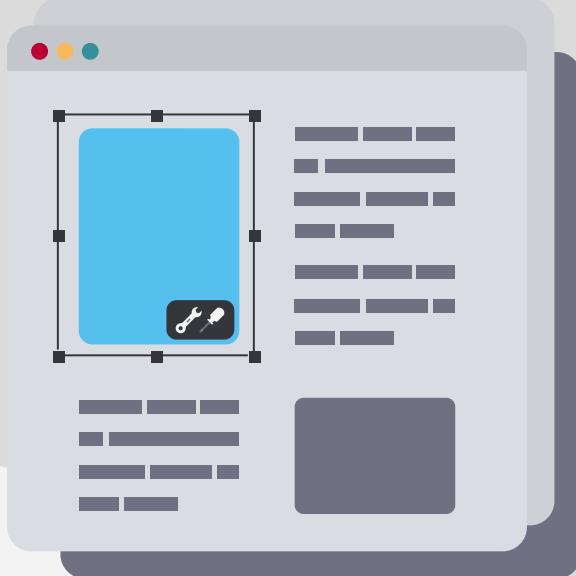
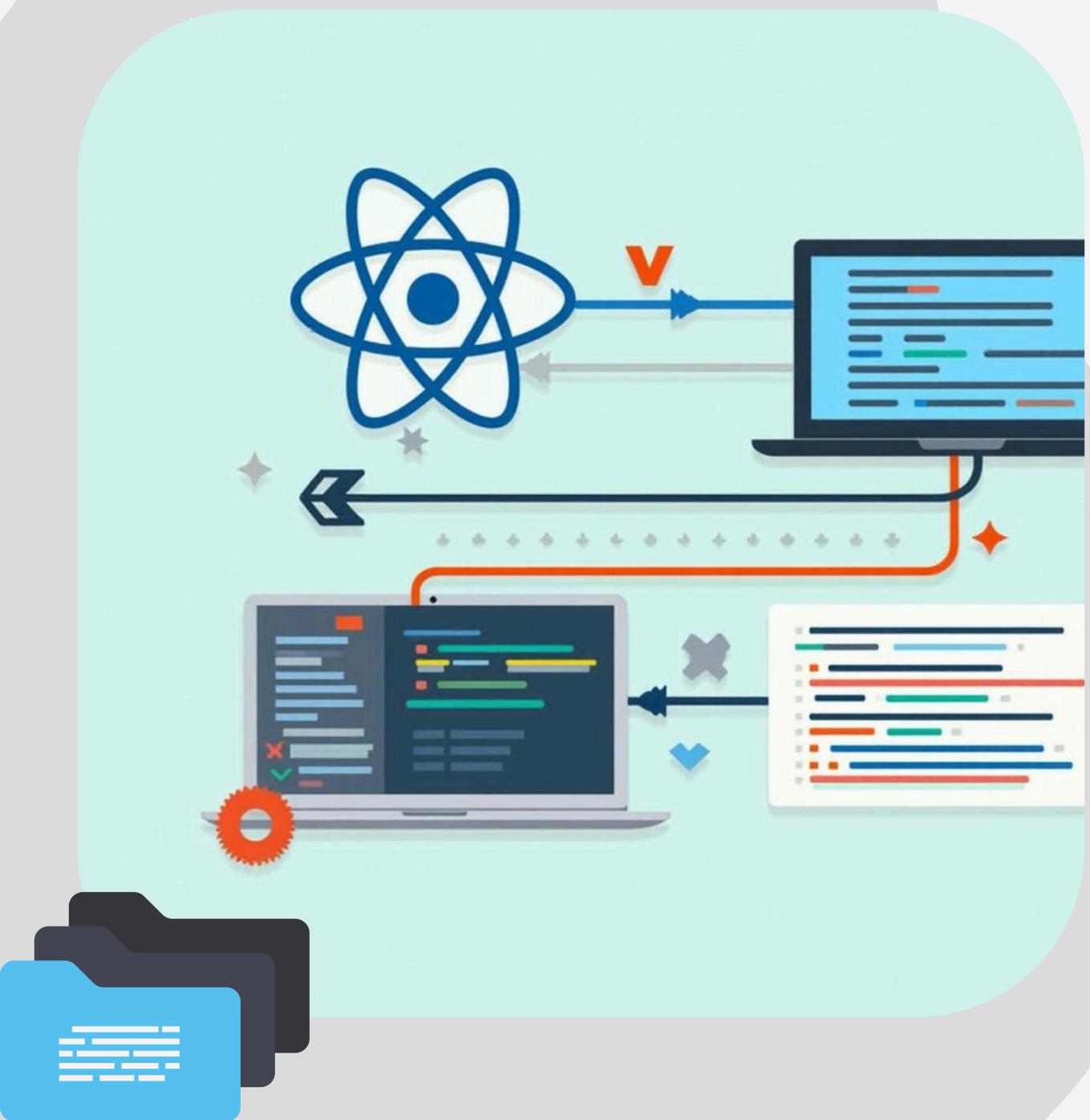
Personal and Professional Aspirations

This project serves as a **professional portfolio** to secure future internships by demonstrating authentic web development skills. It reflects a commitment to ongoing improvement and personal branding. The focus is on building a polished, scalable site that evolves with new projects and technologies, showcasing expertise in React and modern frontend frameworks while preparing for industry challenges.



02

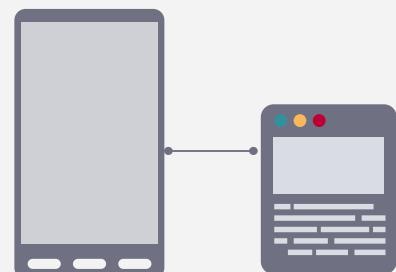
Implementation and Enhancements





Architecture and Component Design

The application's architecture employs React with a **component-based structure** for maintainability. Key components include header navigation, project cards with glassmorphism styling, a dynamic contact form, and theme toggle using Context API. Each component is designed for reusability and responsive behavior, ensuring consistent user experience across devices and screen sizes.



API Integration and User Interaction

Real-time data from GitHub's REST API enhances portfolio relevance by displaying live repository stats. The Projects page features **search and filtering functionality** to improve navigation. User interaction is enriched by animated forms with input validation and light/dark theme persistence, creating a professional and engaging interface tailored for diverse users.





Technology Stack

Frontend Technologies:

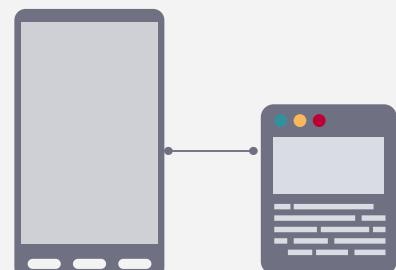
- **React** – Component architecture and UI rendering
- **React Router** – Navigation between pages
- **Context API** – Global theme management
- **CSS (Variables + custom layouts)** – Styling and responsiveness

API & Data:

- **GitHub REST API** – Fetching live repository data
- **JavaScript Fetch** – Handling requests and responses

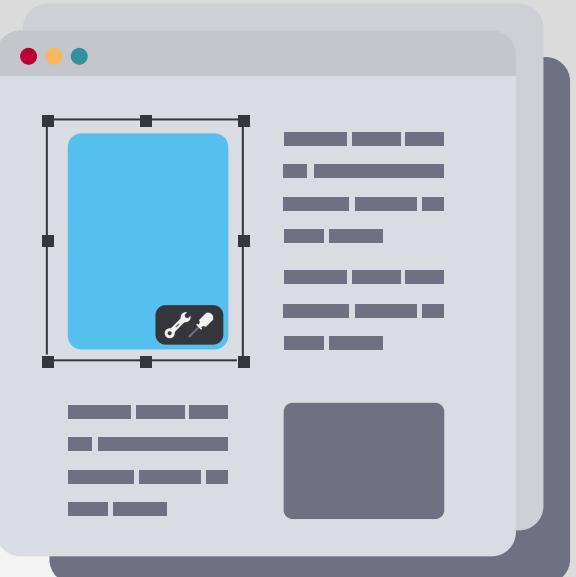
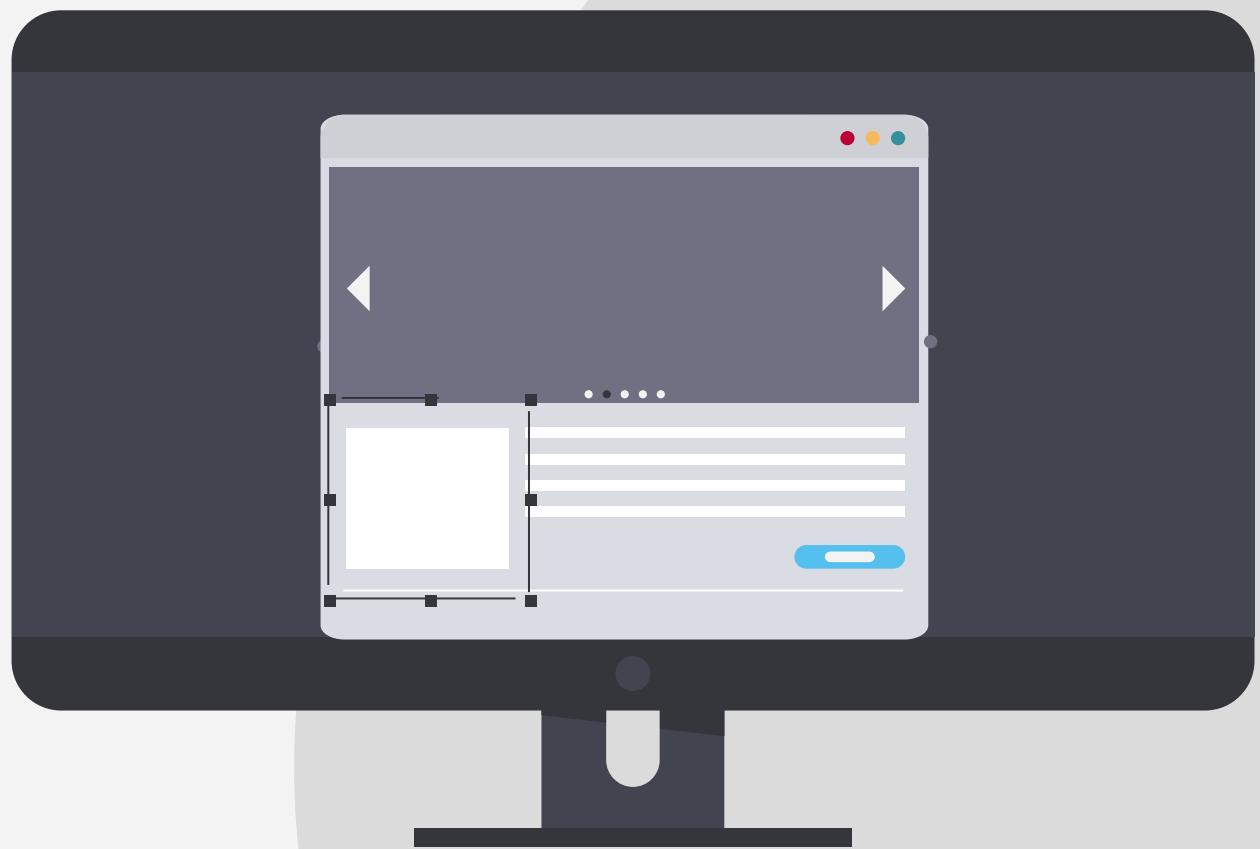
Deployment & Tools:

- **Vite** – Development and build setup
- **Netlify** – Hosting and automated deployment

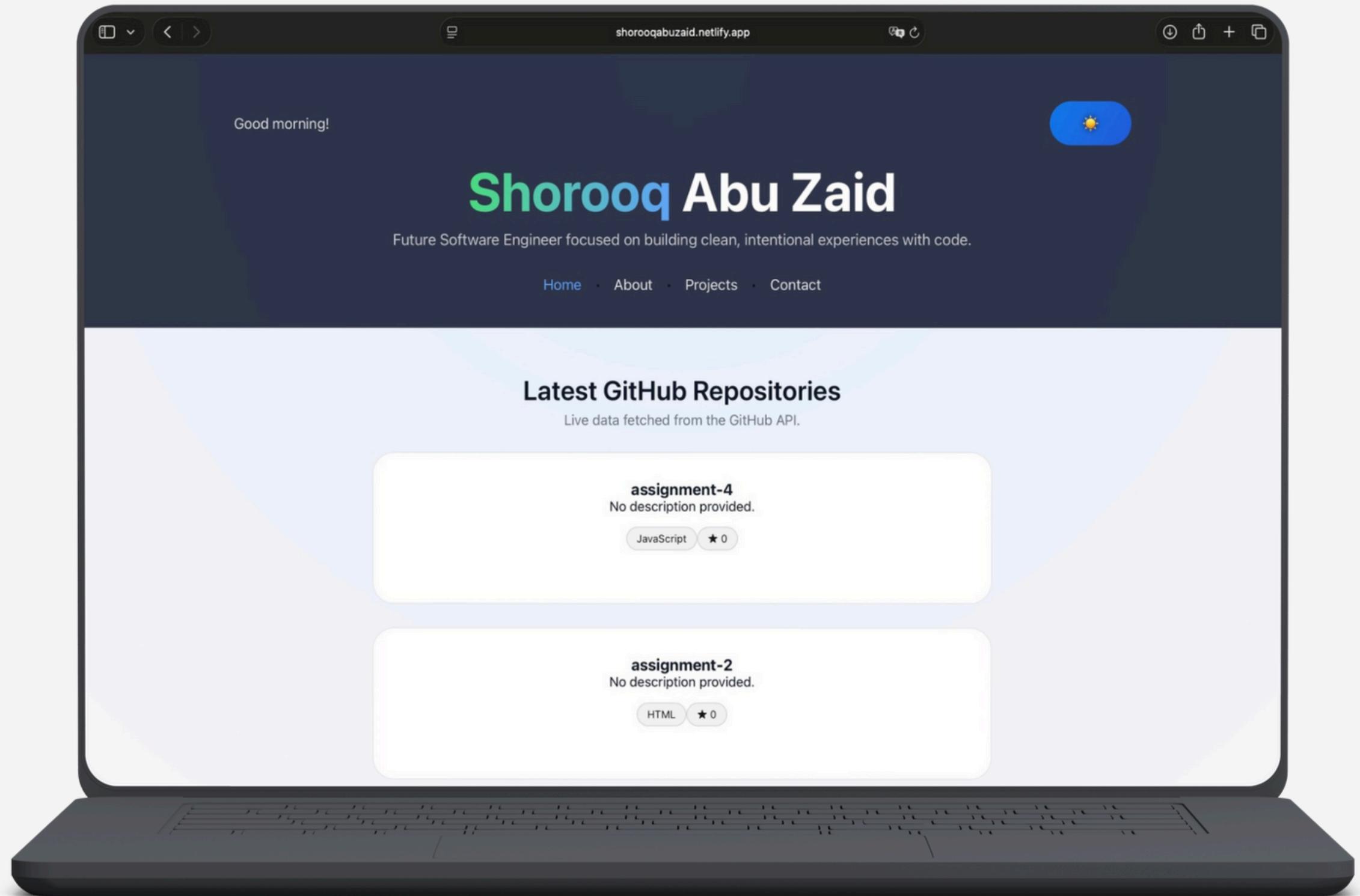


03

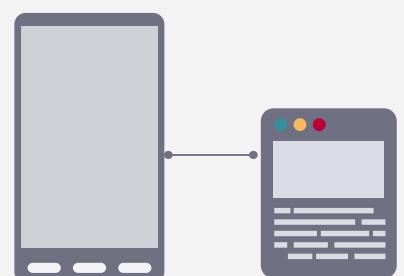
Live Demo



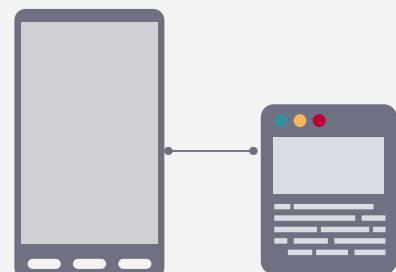
Home Page



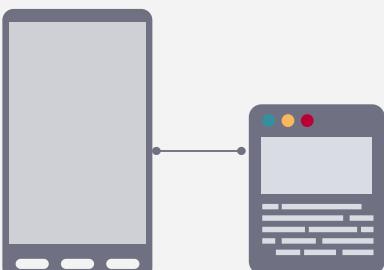
Live GitHub repository display using GitHub REST API



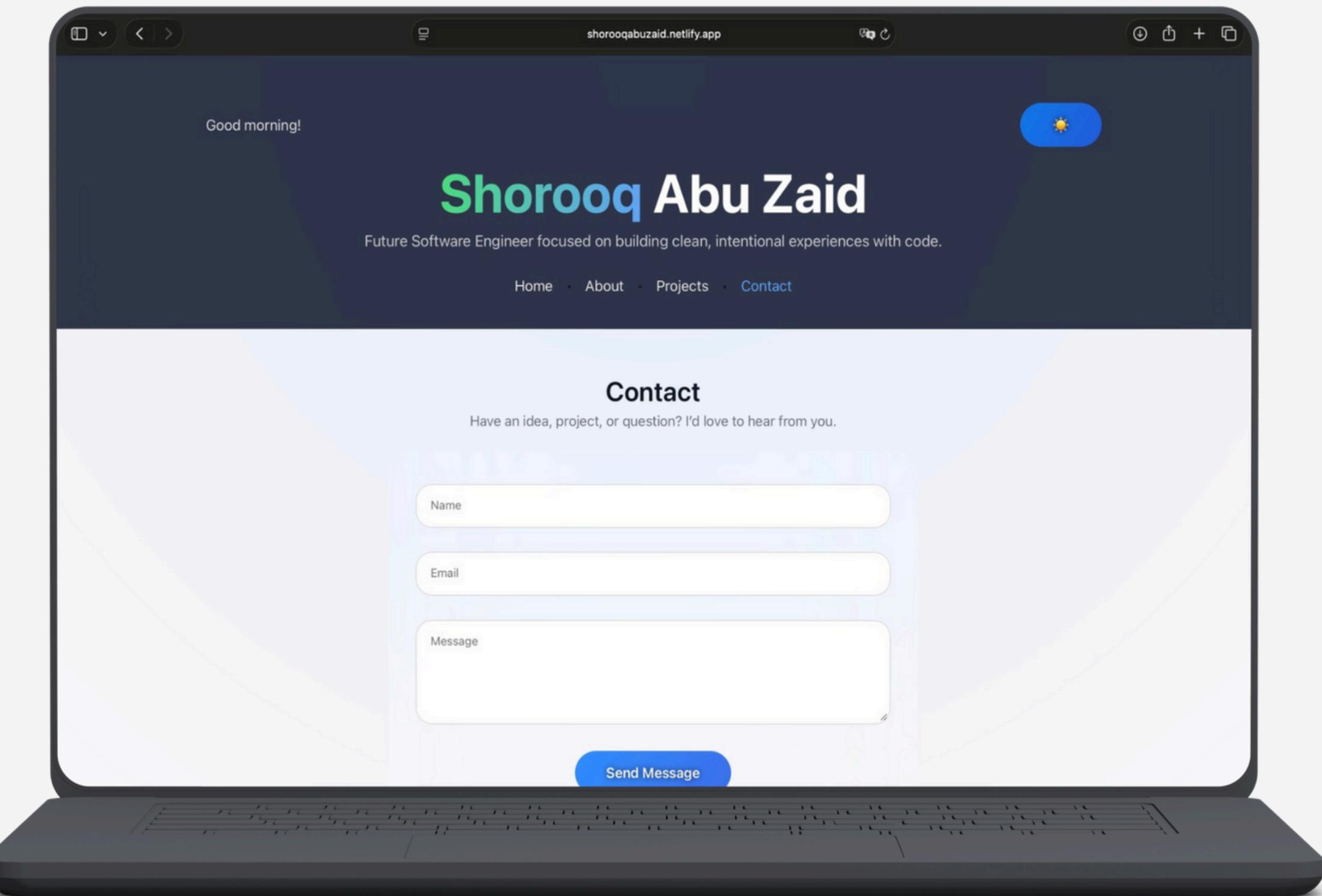
About Page



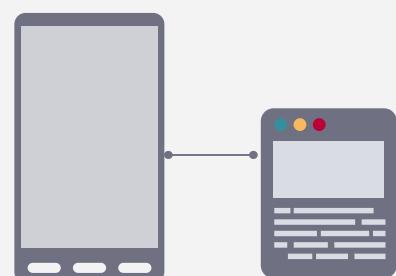
Project Page



Contact Page



Validation, error handling, and animated feedback



04

AI Integration





AI Integration

How AI supported development:

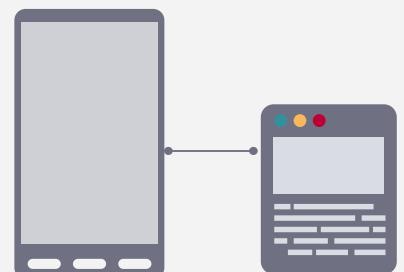
- Clarified React concepts (routing, state, context)
- Helped debug API and styling issues
- Provided suggestions for layout and responsive UI
- Assisted in writing documentation and presentation structure

Responsible usage:

- All AI outputs were reviewed and edited
- Code was never copied blindly
- AI used as a learning and debugging companion

Outcomes:

- Faster development process
- Cleaner architecture
- Clearer documentation
- Stronger understanding of front-end patterns





Challenges, Solutions, and Future Plans

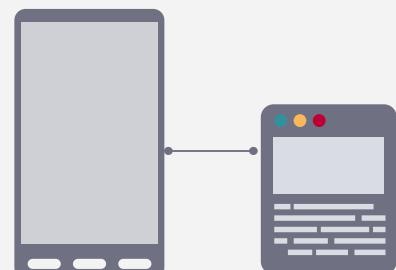
Key challenges included mastering Context API, handling GitHub API errors, and ensuring responsive design.

These were addressed with AI-assisted debugging, improved error handling, and rigorous cross-device testing.

Lessons Learned:

- Strengthened understanding of React's component architecture
- Improved skills in API handling, loading states, and error boundaries
- Learned to design scalable UI with reusable components
- Improved responsive CSS techniques across different screens
- Gained experience using Context API for global state
- Learned to use AI responsibly for debugging and documentation

Future plans include backend email integration, accessibility enhancements, analytics implementation, and adding multilingual support to broaden reach and improve functionality.



Conclusions

This personal portfolio project combines **modern frontend technologies** with practical design and deployment strategies to create a professional web presence. It reflects a strong foundation in React development, API integration, and UI/UX best practices, supporting continuous growth and adaptability in the evolving digital landscape. The project sets a solid base for future enhancements and career opportunities.



Thanks!

Do you have any questions?

s202257840@kfupm.edu.sa

<https://shorooqabuzaid.netlify.app>

<https://github.com/ShorooqAbuZaid711/assignment-4.git>

