

CAPSTONE PROJECT REPORT

Project Title: React-Based Web Application using Node.js Environment

1. Abstract

This project is a modern web application built using React.js and deployed through a Node.js environment. The application demonstrates core concepts of frontend development such as component-based architecture, state management, routing, responsive UI, and API integration. The project also includes version control using Git & GitHub and deployment using Vercel.

2. Introduction

The objective of this capstone project was to create a fully functional React application, implement essential features, manage the codebase with Git/GitHub, and deploy the final output to a live hosting platform.

3. Project Objectives

- Develop a clean and structured front-end application using React.
- Implement interactive features using props, state, and hooks.
- Practice modular coding and reusable components.
- Use Git and GitHub for version control.
- Deploy using Vercel.

4. System Requirements

Software: Node.js, npm, VS Code, Git, Vercel.

Hardware: Minimum 4GB RAM.

5. Technologies Used

React.js, JavaScript, HTML, CSS, Git/GitHub, Node.js environment, Vercel.

6. Project Features

- Interactive UI with reusable components
- State management using hooks
- Routing with react-router-dom

- Form handling and validation
- Responsive design
- LocalStorage support (optional)
- API integration (optional)

7. System Architecture

Frontend structure: src/, components/, pages/, public/. Data flow uses state + props and component re-rendering.

8. Implementation Details

Project setup, development phases, version control workflow, and deployment steps were followed as per standard industry practices.

9. Screenshots

(Placeholder section)

10. Testing

Performed mobile responsiveness tests, browser compatibility checks, validation tests, and routing verification.

11. Deployment Steps (Vercel)

1. Login to Vercel.
2. Import GitHub repository.
3. Configure build command: npm run build.
4. Set output directory (build or dist).
5. Deploy.

12. Conclusion

This project demonstrates complete lifecycle development using React, GitHub, and Vercel. It strengthened understanding of UI components, state management, routing, deployment, and documentation.

13. Future Enhancements

- Add authentication
- Add backend (Node/Express)
- Add database
- Improve UI themes
- Expand functionality

14. References

React Docs, MDN, Vercel Docs, GitHub Docs.