COLLEGE CODE : 9609

COLLEGE NAME : JAYAMATHA ENGINEERING COLLEGE

DEPARTMENT : CSE

STUDENT NM ID :8434ED142DF1C3BA5AF4C62904BCE6D8

ROLL NO : 960923104004

DATE : 13/10/2025

Completed the project named as

Phase - 5 PROJECT DEMONSTRATION & DOCUMENTATION

Name : IBM-FE- SINGLE PAGE APPLICATION

SUBMITTED BY,

NAME : M. GAYATHRI

MOBILE NO : 9159082673

1. FINAL DEMO WALKTHROUGH

- **This is typically a live or recorded demonstration of the final, working project.**
- **Goal: To showcase the project's features and functionality to the audience (e.g., instructors, stakeholders).**
- **Content should cover:**

- **Setup: A brief overview of how to get the project running.**

- **Core Functionality: A step-by-step demonstration of the key features and use cases that the project was designed for.**

- **User Experience: Showing the typical user flow through the application or system.**

- **Conclusion: Briefly summarizing the project's success in meeting the initial objectives.**

2. PROJECT REPORT

- **The project report is a comprehensive written document summarizing the entire project lifecycle, from initial concept to final deployment.**
- **Goal: To formally document the project's scope, methodology, results, and evaluation.**
- **Key Sections often include:**

- **Introduction: Project aims, scope, and objectives.**

- **Literature Review/Background: Contextual information and technologies used.**

- **Design & Methodology: Explanation of the architecture, chosen**

tools/technologies, and implementation steps.

- **Results & Evaluation: Discussion of the achieved results, meeting the requirements, and performance analysis.**

- **Conclusion & Future Work: Summary of findings and potential improvements or next steps.**

3. SCREENSHOTS / API DOCUMENTATION

- **This section covers the technical details necessary for others to understand and potentially use or integrate with the project.**

- **Screenshots: Visual proof of the working application. Include images of the main interfaces, key features, and successful results/outputs.**

- **API Documentation (if applicable): If the project involves an Application Programming Interface (API), this is crucial. It should detail:**

- **Endpoints: The URLs for accessing different functions (e.g., /users, /products).**

- **Methods: The HTTP methods used (GET, POST, PUT, DELETE).**

- **Request/Response Formats: Examples of data structures sent to and received from the API (e.g., JSON payload examples).**

- **Authentication/Authorization: How users/services secure access to the API.**

4. CHALLENGES & SOLUTIONS

- **This is a reflective section where you document the difficulties encountered during the project and how you overcame them.**
- **Goal: To show critical thinking, problem-solving skills, and a transparent**

account of the development process.

- **For each major challenge, document:**

- **The Problem: Clearly describe the technical, design, or conceptual hurdle faced (e.g., "Difficulty integrating the third-party payment gateway," or "Performance degradation with large datasets").**

- **The Solution: Explain the specific steps, code changes, or design decisions implemented to resolve the problem.**

- **Learnings: Briefly state what was learned from the experience.**

5. GITHUB README & SETUP GUIDE

- **The README file is the first thing people see in your code repository and is essential for project visibility and usability.**
- **GitHub README: Should contain a concise, professional overview of the project, including:**

- **Project Title & Description: A brief, compelling summary.**

- **Features: A bulleted list of the main capabilities.**

- **Technologies Used: A list of the tech stack (e.g., React, Python/Flask, MongoDB).**

- **Links: A link to the live deployment (if available).**
- **Setup Guide: Detailed, step-by-step instructions for a new user or developer to clone, install dependencies, configure, and successfully run the project locally. This should be clear enough for someone unfamiliar with your project to get it running without assistance.**

6. FINAL SUBMISSION (REPO + DEPLOYED LINK)

- **This is the delivery mechanism for the project's final artifacts.**

- **Repo (Repository): The final, clean, and complete source code, typically submitted via a link to a GitHub/GitLab repository. The code should be well-organized and include any necessary configuration files.**

- **Deployed Link: A live URL where the final, working application or system can be accessed and tested by the evaluators. This confirms the project is operational in a production or staging environment.**