Implementation Document

for

ProfInfo Central

Version 1.0

Prepared by

Group : 2 **Group Name:** Error 404: team not found

Pavani priya	220415	gppriya22@iitk.ac.in
Sontam Deekshitha	221075	sontamd22@iitk.ac.in
Lakshyta Mahajan	220581	lakshyta22@iitk.ac.in
Kartik	220503	kartik22@iitk.ac.in
Atharv Moghe	220250	atharvm22@iitk.ac.in
Prabhat Kumar Yadav	220774	prabhatky22@iitk.ac.in
Nilesh Maneshwar	220715	mnilesh22@iitk.ac.in
Kuldeep Sandip Thakare	220557	kuldeeps22@iitk.ac.in
Mohd Nasar Siddiqui	220661	snasar22@iitk.ac.in
Sanapala Jaswanth	220955	sjaswanth22@iitk.ac.in

Course:	CS253
Mentor TA:	Abhilash Chandra
Date:	14-03-2024

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Version 1	Error 404: team not found	Initiated the document and added all necessary details. Completed all sections.	14/03/2024

1 Implementation Details

Details of Programming Languages, Frameworks and Libraries used :-

1) **Programming Languages** – JavaScript, HTML and CSS.

For frontend development we have used JavaScript as our main programming language along with HTML (Hyper Text Markup Language) as a standard markup language for structuring ,creating pages and displaying them on the website.
JavaScript was used to make the pages dynamic enabling user interaction and providing a more engaging user experience which is very limited in HTML. For styling we have used CSS(Cascading Style Sheets).

2) **Libraries Used** – React.js ,React-router-dom, Axios.

Here we have preferred the React library of JavaScript because it offers a component-based architecture, virtual DOM for efficient updates, and a rich ecosystem. It facilitates reusable components, simplifies complex UIs, and enhances performance. Overall, React.js provides productivity, scalability, and a vibrant community, distinguishing it from alternatives.
React-Router-Dom is used for routing of all the pages we have created in different .jsx files. It helped for navigation and URL handling by providing components like BrowserRouter, Routes, and Route.
Axios offers a cleaner, promise-based approach for handling HTTP requests, simplifying asynchronous code. Its intuitive API, compatibility with both browsers and Node.js, and features like interceptors and automatic JSON parsing make it a popular choice for developers seeking simplicity, flexibility, and robustness in their HTTP interactions.

3) **Frameworks Used** – Express.js

We have used the Express framework because it offers simplicity, flexibility, and robust middleware support, allowing efficient creation of web servers. It provides easy routing and streamlines development. Express remains highly popular due to its scalability, versatility, and vibrant community support, making it a top choice for Node.js developers.

4) **Database Systems Used** – MongoDB

We have used MongoDB over other Relational Database Management systems because of its immense benefits over other Database management systems. MongoDB provides advantages over traditional SQL databases with its schema flexibility, enabling storage of dynamic data models. Its horizontal scalability ensures efficient handling of large datasets and high traffic. Performance benefits arise from its document-oriented model, reducing the need for complex joins, while developer productivity improves due to native support for modern application development practices.

5) **Runtime JS environment** – NodeJS

We have used NodeJS for building the runtime environment. NodeJS is used because of its faster development time and it offers an asynchronous, event-driven runtime environment, making it ideal for scalable, real-time applications. Its single-language paradigm with JavaScript simplifies development, and npm provides a vast library ecosystem. With strong community support and enterprise adoption, Node.js facilitates rapid development of microservices architectures and real-time web applications. Overall, Node.js delivers agility, scalability, and performance, making it a compelling choice for modern web development needs.

2 Codebase

GitHub Repository - https://github.com/kartik3777/ProfInfo-Central

Structure of GitHub Repository

- /Documentation -
    - Software Design Document.pdf
    - Software Requirement Specifications.pdf
    - Software Implementation Document.pdf

- /Frontend
    - public
        - images
            - abhilash.jpeg
            - atharv.jpg
            - back.png
            - deekshitha.jpg
            - home_image.png
            - home_image_2.png
            - iitk.gif
            - indmil.jpg
            - jaswanth.jpg
            - kartik.jpg
            - kuldeep.jpg
            - lakshyta.jpg
            - logo.gif
            - nasar.jpg
            - nilesh.jpg
            - pavani.jpg
            - prabhat.jpeg
            - prof-profile-image.png
        - index.html
        - style.css
    - src
        - components
            - Faculty
                - department
                    - Ce.jsx
                    - Cse.jsx
                    - Ee.jsx
                - Faculties.js
                - Loader.css
                - Loader.js
                - Menu.js
                - Prof.js
                - main.js

        - prof.css
      - Student
        - Project Categories.jsx
        - ProjectDesc.jsx
        - StudentCard.jsx
        - User-Interface-Design-Inspiration-25-29.png
        - domain.js
        - projectcard.css
        - Inspira...
        - projectdesc.css
        - projectdetails.js
        - studentcard.css
        - studentpage.jpg
      - Authentication
        - Forgot Password.js
        - SignIn.js
        - SignUp.js
        - styl.css
    - pages
    - App.jsx
    - Requests.css
    - index.js
  - .gitignore
  - README.md
  - package-lock.json
  - package.json
- /Backend
  1. Controllers
     - errorController.js
     - facultyController.js
     - projectController.js
     - userController.js
     - userauthController.js
  2. Middleware
     - professor.js
     - uerinfo.js
  3. Models
     - professors.js
     - project.js
     - usermodel.js
  4. Routes
     - facultyRoutes.js
     - professorRoutes.js
     - userRoutes.js
  5. utils
     - APifeatures.js
     - appError,js
     - catchAsync.js
  6. Procfile

7. README.md
8. app.js’
9. config.env
10. package-lock.json
11. professorsdb.js
12. vercel.json

About the Web App

As of now, the front-end and the backend are separate units.

For Back-end Part,

GitHub repositories can be accessed anywhere worldwide.

For the Front-end Part,

The web app source code can be run using npm and hosts the main page, login page, profile page, project page, project details page, and page.

Codebase Navigation

The Front-end code is present in “/Frontend/.” The “src” folder contains all the source code files.

The Back-end code is present in “/Backend Units.” The individual folders contain the source codes of the individual units. The “src” folder in every folder includes the source code files individually. However, the “sample data” folder contains some sample data fetched from the MongoDB database for demonstration purposes.

Installation Requirements

1. Node.js
2.React.js
3.Express.js
4.Axios
5.Slugify
6.Validators
7. Web vitals

Installation

For Frontend, i.e., website,

1. Navigate to “/Frontend/.”
2. Open Command Line Terminal in this folder.
3. Run the following commands one by one -
4. npm install
5. npm start

The webpage will automatically open in localhost (http://localhost:3000/).

If it doesn't automatically open, the user can open it manually.

For the Back-end part,

1. Navigate to “ ”
2. Open Command Line Terminal in this folder.
3. Run the following command one by one.
4. npm install
5.Nodemon app.js

Now the user can test the APIs on localhost (http://localhost:5000/).

Test Run

The webpage can be tested using npm as stated in the "Installation" part.

The Backend Units can be tested using POSTMAN .

3 Completeness

SRS Feature	Functionality	User	Status	Future Development Plan
Profile Login/ Registration	Registration and Logging in by User	Students	Completed	Give users the option to reset their password by authenticating the user with the help of OTP.
Profile Login/ Registration	Registration and Logging in by User	Professors	Completed	Give Professors the option to set their own passwords for their account.
Profile	Profile info and edit option	Both Professors and Students	Completed	Enable users to link and share content from their profiles on external platforms.
Projects Info	Info about the projects offered by all the professors	Students	Completed	Filtering of projects based on tags.
Project Edits	Info, Edit and Add options for all the projects offered by a professor	Professors	Completed	Send a pop-up to all students whenever a professor adds a new project.
Requests Info	Request and Withdraw request option for Projects	Students	In Progress	Enabling the request option only if basic criteria like department or cpi(if applicable) are satisfied.
Request Edits	Request Accept and Decline options for a specific project	Professors	In Progress	Giving an option of notifying a student for a meet before

	offered by the professor			accepting/declining.

Appendix A - Group Log

S. No.	Date	Mode	Venue	Activity
1	12/02/24	Offline	RM	Delegated tasks across the team and strategized the implementation roadmap.
2	15/02/24	Offline	RM	Deliberated on the development of the backend infrastructure and provided a succinct overview of the application's design framework.
3	28/02/24	Offline	RM	Team members provided updates on the status of their tasks, following which additional assignments were allocated as needed.
4	03/03/24	Offline	RM	Integrated fundamental CSS styling to webpages and completed the implementation of React.js for the remaining pages.
5	05/03/24	Offline	RM	We deliberated on the database implementation and initiated the implementation process.
6	10/03/24	Offline	RM	Addressed and resolved issues encountered during the integration of the backend with the frontend, worked towards their resolution, and made minor adjustments to the CSS as needed.
7	13/03/24	Offline	RM	Completed the Software Implementation Document and addressed inquiries within the application.