Industrial Training Report

on

Frontend Web Development

Submitted in Partial Fulfillment for the Degree of

Bachelor of Technology

in

Computer Science and Engineering

Submitted By:

Sneha Gupta: (21EEMCS116)

Training

by

AI Chef Master

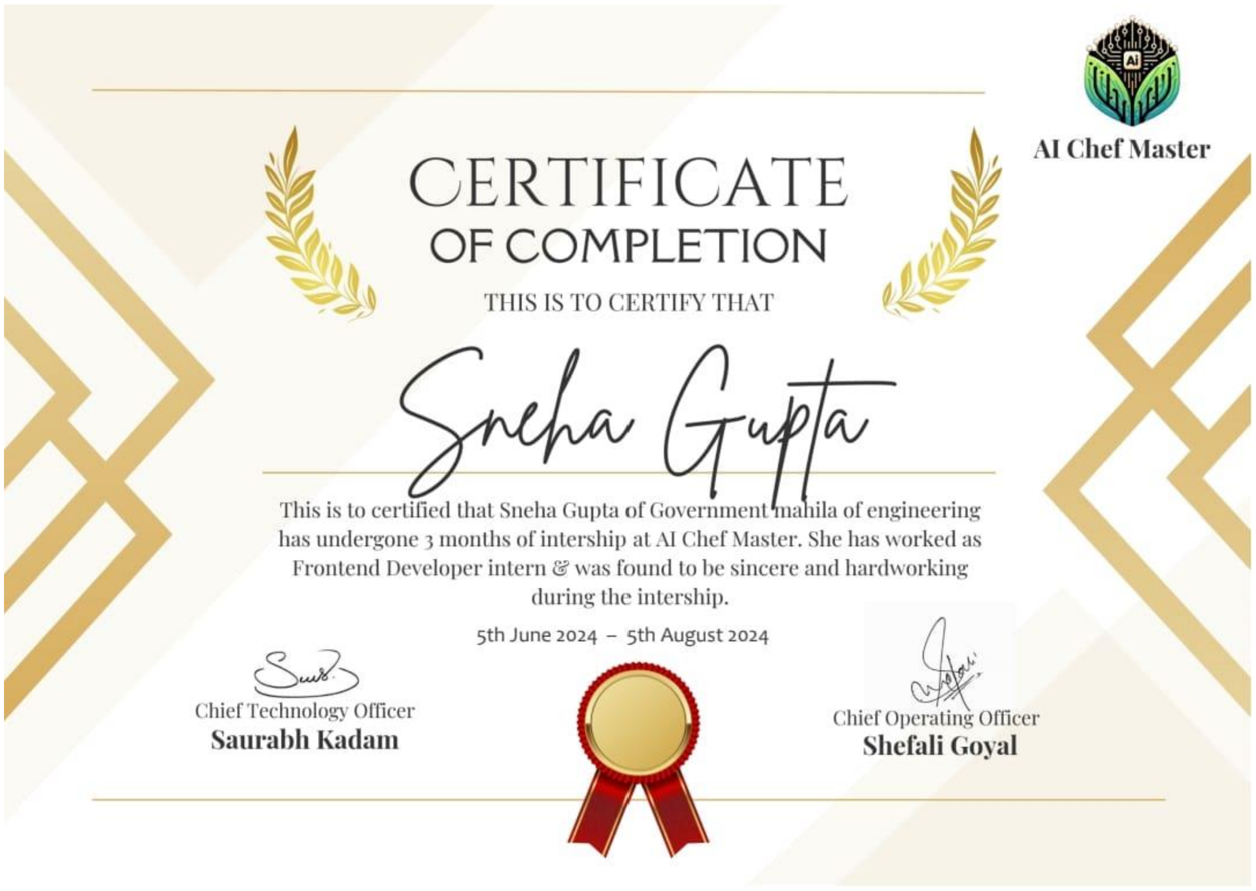
Submitted

to

Ms. Monalisa Meena	Dr. Meeta Sharma
Assistant Professor	Assistant Professor
Industrial Training Coordinator	HoD CE Department

Government Women Engineering College, Ajmer
Department of Computer Engineering
December, 2024-25

CERTIFICATE



AI Chef Master

CERTIFICATE
OF COMPLETION

THIS IS TO CERTIFY THAT

Sneha Gupta

This is to certified that Sneha Gupta of Government mahila of engineering has undergone 3 months of intership at AI Chef Master. She has worked as Frontend Developer intern & was found to be sincere and hardworking during the intership.

5th June 2024 – 5th August 2024

[illegible]
Chief Technology Officer
Saurabh Kadam

[illegible]
Chief Operating Officer
Shefali Goyal

ACKNOWLEDGEMENT

I **Sneha Gupta (21EEMCS116)** deeply grateful for the incredible opportunity to work as a frontend web developer at AI Chef Master. This journey has been an enriching and transformative experience, allowing me to grow both professionally and personally.

I extend my heartfelt gratitude to **Mr. Saurabh Kadam Sir**, **Mr. Prem Patil Sir**, and **Ms**. **Shefali Goyal Ma'am** for their unwavering guidance, mentorship, and support throughout this internship. Your expertise, encouragement, and invaluable insights have been instrumental in shaping my contributions to this innovative venture.

Thank you for believing in my abilities and fostering an environment of learning and collaboration. It has been a privilege to work under your guidance, and I am truly honored to have been a part of this remarkable team.

Also I extend my thanks and gratitude to MS. Monalisa Meena (Asst.Professor - CE), Govt. Women Engineering College, Ajmer, my internship coordinator whose guidance, teaching and certain suggestions provide me with timely valuable input which enhances my knowledge and thus helped in the development of this internship report.

Sneha Gupta: (21EEMCS116)

ABSTRACT

This report provides an overview of the industrial training undertaken at **AI Chef Master** from **5th June** to **5th August**, as part of the academic curriculum for **Frontend Web Developer**. The primary goal of the training was to bridge the gap between theoretical knowledge and practical application by engaging in real-world industrial processes.

During my internship, I contributed to creating an innovative platform that integrates artificial intelligence into culinary experiences. My key responsibilities included designing and implementing responsive, user-friendly interfaces using modern frontend technologies like HTML, CSS, JavaScript, and React, as well as integrating AI-driven APIs for features like real-time recipe recommendations and personalized interactions. Collaborating closely with backend developers, UI/UX designers, and project leads in an agile environment, I worked on ensuring seamless user experiences, optimizing performance through rigorous testing and debugging, and enhancing engagement with interactive elements. This enriching experience allowed me to deepen my technical skills and understanding of professional collaboration.

TABLE OF CONTENTS

Chapter - 1

INTRODUCTION

AI Chef Master is a growing company in the **Technology, Information, and Internet** industry, headquartered in **Mumbai, Maharashtra**. AI Chef Master sits at the intersection of technology and culinary arts, leveraging cutting-edge AI to create personalized cooking experiences for users.

The company is dedicated to developing advanced AI-powered tools that enhance the cooking process, enabling users to create their own custom AI chefs. Through the platform, users can access personalized recipe suggestions, optimize ingredient usage, and receive real-time cooking guidance. By merging technology with culinary expertise, AI Chef Master aims to make cooking more efficient, accessible, and enjoyable for individuals globally.

The **Technology, Information, and Internet** industry has witnessed exponential growth in recent years, shaping the way we live, work, and interact. Companies like AI Chef Master are tapping into this potential by blending AI with everyday tasks, offering innovative solutions that make life easier and more connected. As part of this fast-growing industry, AI Chef Master is poised to revolutionize the culinary experience, providing users with a platform that combines the latest technological advancements with the art of cooking.

Chapter - 2

OBJECTIVES

1. **Simplify Meal Preparation**: AI-driven recipe recommendations based on ingredients, preferences, and dietary needs to ease meal planning.

2. **Encourage Culinary Exploration**: Inspire users to try new recipes, techniques, and global cuisines to broaden culinary horizons.

3. **Promote Healthy Eating**: Provide nutritious recipe options and personalized suggestions aligned with users' dietary goals and needs.

4. **Save Time and Resources**: Offer quick, efficient recipes and smart ingredient substitutions to minimize waste and cooking time.

5. **Enhance Cooking Skills**: Provide step-by-step guidance and real-time support to improve users' cooking abilities and confidence.

6. **Personalize the Cooking Experience**: Tailor recipe suggestions based on users' preferences, dietary needs, and previous cooking history.

7. **Build a Cooking Community**: Foster a platform for users to share recipes, tips, and connect with other food enthusiasts.

8. **Improve User Engagement**: Encourage interaction through features like saving, rating, reviewing, and personalized recipe recommendations.

Chapter - 3

KEY RESPONSIBILITIES

As a Frontend Web Developer at AI Chef Master, I was responsible for designing and developing user interfaces that were visually appealing and user-friendly. My focus was on creating seamless and intuitive experiences for users, ensuring easy navigation and interaction across the platform.

I worked on developing responsive web pages to ensure the platform provided a smooth experience across various devices, including desktops, tablets, and smartphones. By leveraging modern web technologies, I aimed to create adaptable designs that maintained consistency and functionality, regardless of screen size.

Collaboration with backend developers was essential in my role, as I integrated AI-powered features into the platform. This included working on personalized recipe recommendations and ingredient substitution features, ensuring these AI elements were seamlessly incorporated into the frontend for optimal user experience.

Additionally, I worked closely with UI/UX designers and project managers to implement new features, such as allowing chefs to add their recipes to the platform. This collaboration ensured that new features aligned with the platform's overall user experience goals and design principles.

.

Chapter - 4

TECHNOLOGIES USED

1. React:

React is a JavaScript library for building user interfaces, allowing you to create dynamic, component-based web applications. It helps in creating fast, interactive UIs with minimal effort. It is maintained by Facebook and a community of developers. React allows developers to create reusable UI components that handle their state and efficiently update the DOM (Document Object Model).

Key Features of React:

- Component-Based Architecture: UI is divided into reusable components.
- Virtual DOM: React uses a virtual representation of the DOM to improve performance by minimizing direct DOM manipulations.
- JSX (JavaScript XML): Syntax extension for JavaScript, enabling HTML-like code within JavaScript.
- One-Way Data Binding: Ensures predictable data flow, making debugging easier.
- Declarative: React makes it simple to design interactive UIs by focusing on the "what" instead of the "how."

Installation Command:

```
npx create-react-app my-app
cd my-app
npm start
```

2. Tailwind CSS:

Tailwind CSS is a utility-first CSS framework for rapidly building custom designs. Instead of providing pre-designed components like other CSS frameworks (e.g., Bootstrap), Tailwind offers a collection of utility classes that you can combine to style your elements directly in your markup.

Features of Tailwind CSS:

- Utility-First Approach: Provides low-level utility classes like text-centre, mt-4, flex, etc., to create custom designs without writing CSS.
- Customization: Highly customizable via a configuration file (tailwind.config.js).
- Responsive Design: Includes built-in responsive utilities like sm:, md:, lg:, and xl: for mobile-first development.
- Dark Mode: Provides support for dark mode using utility classes.
- No Design Constraints: You have complete control over the design without being restricted by pre-defined components.

Installation Command:

```
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init
```

Then, add the following to your `tailwind.config.js` file:

```
module.exports = {
 content: ["./src/**/*.{html,js,jsx,ts,tsx}"],
 theme: {
   extend: {},
 },
 plugins: [],
}
```

3. GitHub:

GitHub is a web-based platform for version control using Git, enabling teams to collaborate on code and manage project versions. It helps in tracking changes and working with repositories.

Key Features of GitHub:

- Version Control: Tracks changes to code and allows reverting to previous versions.
- Collaboration: Facilitates teamwork with features like pull requests, reviews, and comments.
- Branching: Enables developers to work on separate features or fixes without disrupting the main codebase.
- Hosting: Provides a remote storage location for Git repositories.
- Integration: Supports continuous integration/deployment (CI/CD) workflows with services like GitHub Actions.
- Community: Hosts open-source projects, where developers contribute to shared repositories.

Git Bash:

Git Bash is a terminal application that provides a Unix-like command-line interface for Windows users to interact with Git. It includes basic Unix commands, enabling seamless use of Git's functionality in Windows environments.

Key Features of Git Bash:

- Git Commands: Offers full access to Git commands like git init, git commit, git push, etc.
- Unix Commands: Provides Unix utilities such as ls, cat, and ssh.
- Cross-Platform Development: Bridges the gap between Windows and Unix-like systems for Git users.

Installation Command:

```
sudo apt install git   # Linux
brew install git       # macOS

git config --global user.name "Your Name"
git config --global user.email "your-email@example.com"
```

Git pull, commit, push commands to make and save changes in repository:

```
git pull origin main

git add .
git commit -m "Your commit message here"

git push origin main
```

4. Visual Studio Code (VS Code):

Visual Studio Code (VS Code) is a lightweight yet powerful source-code editor developed by Microsoft. It supports multiple programming languages and frameworks, offering features that enhance productivity for developers.

Key Features of VS Code:

- Cross-Platform: Available on Windows, macOS, and Linux.
- Integrated Terminal: Provides a built-in terminal for running commands without leaving the editor.
- Extensions Marketplace: Thousands of extensions to add functionalities, such as language support, linters, formatters, themes, and debuggers.
- IntelliSense: Offers smart code suggestions, autocompletion, and error highlighting.
- Git Integration: Includes built-in Git support for version control.
- Debugger: Allows debugging of code directly from the editor.
- Customizable: Fully customizable interface and keyboard shortcuts.
- Live Share: Collaborate with others in real-time for pair programming.

5. Material Tailwind:

Material Tailwind combines the aesthetics of Material Design with the utility-first approach of Tailwind CSS. It is a library of pre-designed, responsive components and templates styled using Tailwind CSS with a Material Design touch. This makes it an excellent choice for developers looking for a fast, elegant, and customizable design system.

Key Features of Material Tailwind:

- ➢ Pre-designed Components: Includes buttons, cards, modals, navigation bars, and more.
- ➢ Material Design Principles: Incorporates Google's Material Design philosophy.
- ➢ Utility-First Approach: Leverages Tailwind's utility classes for custom styling.
- ➢ Customizability: Tailwind's configuration allows for complete control over styling.
- ➢ Responsive and Mobile-Friendly: Ensures components work seamlessly across devices.

Installation and Setup:

Prerequisites:

- A React project set up using Create React App, Next.js, or similar.
- Tailwind CSS installed and configured in your project.

Steps to Install Material Tailwind:

- Install Material Tailwind via NPM:

```
npm install @material-tailwind/react
```

- Add Tailwind Directives: In your Tailwind CSS file (e.g., index.css), ensure you have the Tailwind base, components, and utilities directives:

```
@tailwind base;

@tailwind components;

@tailwind utilities;
```

- Import Material Tailwind Components: You can now use Material Tailwind components in your React application:

```
import { Button } from "@material-tailwind/react";
function App() {
 return (
    <div className="flex items-center justify-center min-h-screen">
              <Button color="blue">Click Me</Button>
    </div>
   );
}
export default App;
```

6. Vite:

Vite (pronounced "veet," meaning "fast" in French) is a next-generation front-end tool that provides an extremely fast development environment and optimized production builds for modern web projects. Created by Evan You, the creator of Vue.js, Vite is framework-agnostic and works with React, Vue, Svelte, and others.

Key Features of Vite:

- ➢ Blazing Fast Development: Uses native ES modules for instant server startup and fast updates.
- ➢ Optimized Builds: Bundles code with Rollup, optimizing it for production.
- ➢ Hot Module Replacement (HMR): Updates specific modules without a full page reload, preserving application state during development.
- ➢ Framework-Agnostic: Supports popular frameworks like React, Vue, Svelte, and Vanilla JavaScript.
- ➢ Rich Plugin Ecosystem: Compatible with Rollup plugins and has its own ecosystem of plugins.

Installation and Setup:

Prerequisites:

- Node.js installed on your system (version 12.0.0 or higher is recommended).

Steps to Create a Vite Project:

- Create a New Project: Use the following command to scaffold a new project:

  ```
  npm create vite@latest my-project
  ```

  During the process, you'll be asked to select a framework (e.g., React, Vue, Svelte) and variant (e.g., TypeScript).

- Navigate to Your Project Directory:

  ```
  cd my-project
  ```

- Install Dependencies:

  ```
  npm install
  ```

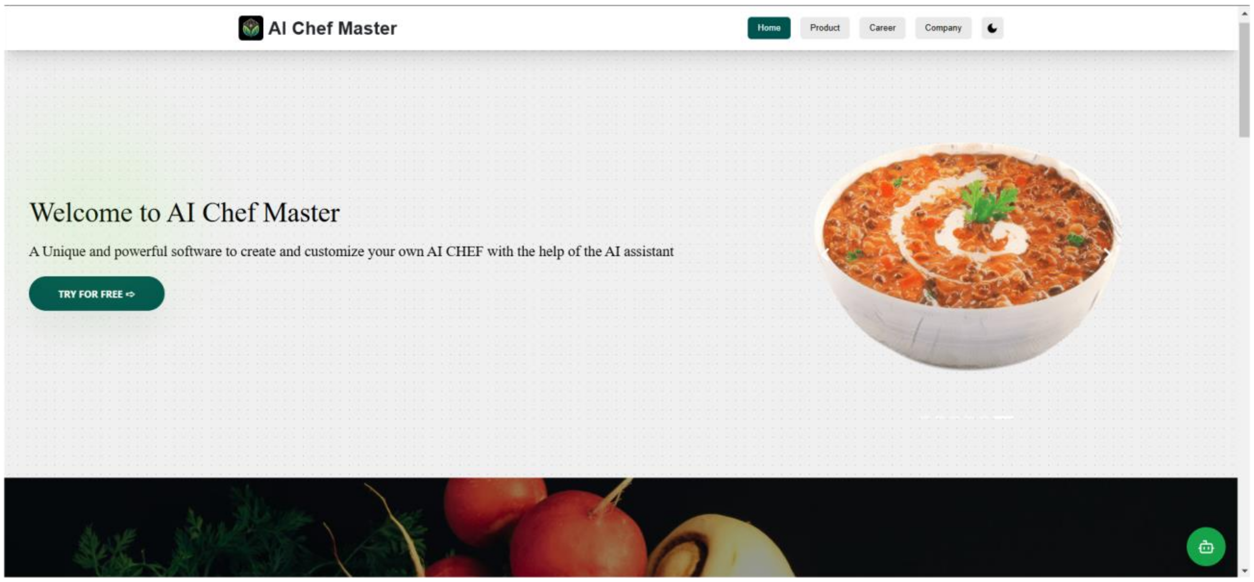
- Start the Development Server:

  ```
  npm run dev
  ```

- The server will start, and you can access your application at http://localhost:5173 (or another specified port).

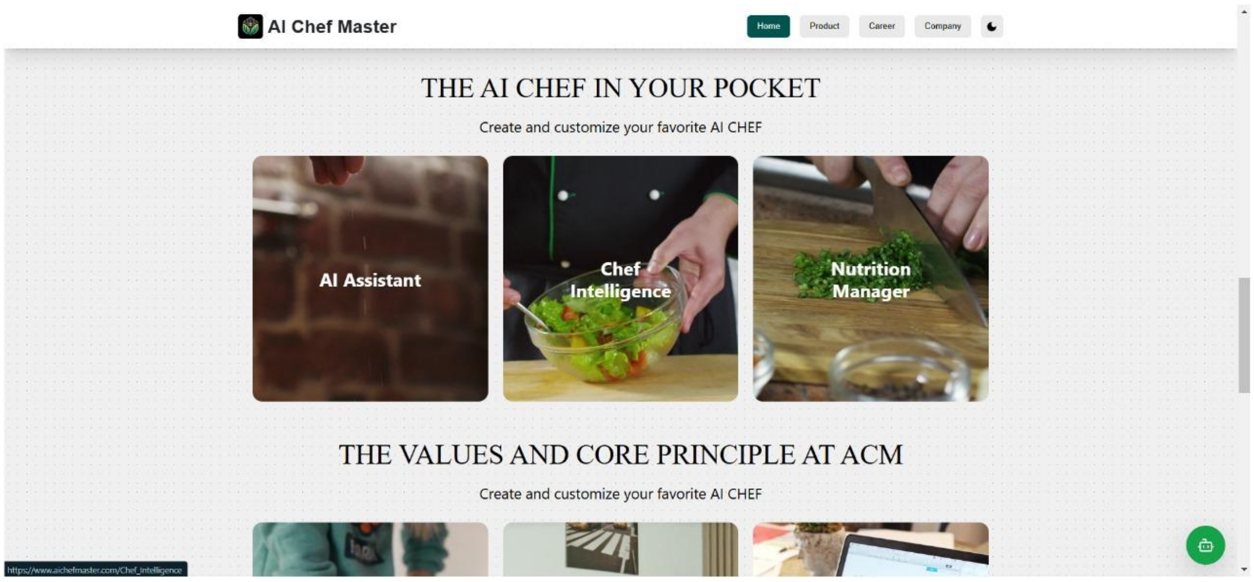
Chapter - 5

WEBSITE



Welcome to AI Chef Master

A Unique and powerful software to create and customize your own AI CHEF with the help of the AI assistant



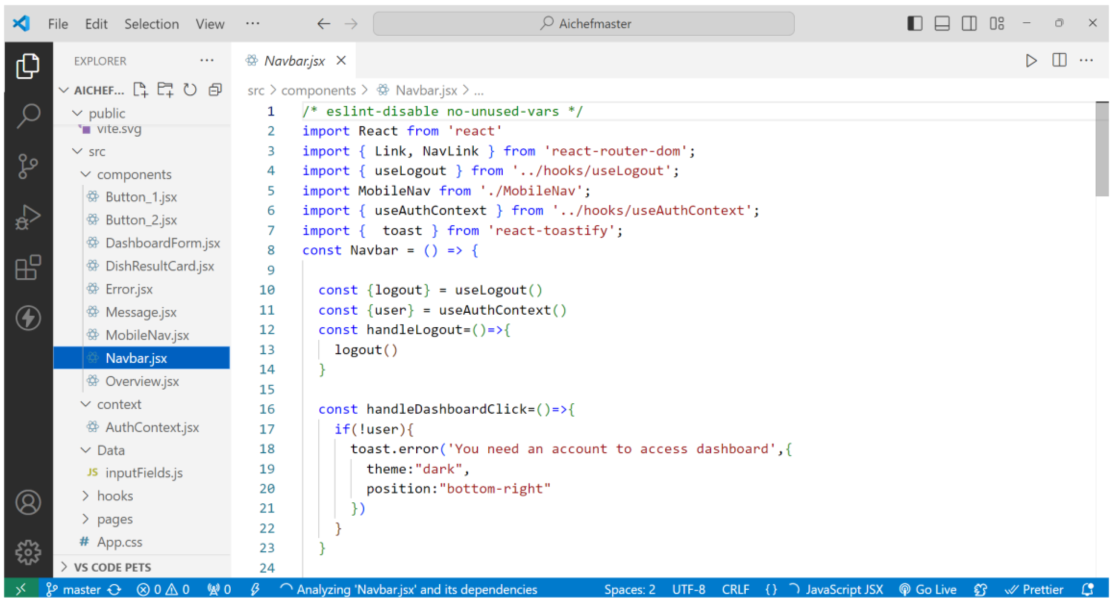
THE AI CHEF IN YOUR POCKET

Create and customize your favorite AI CHEF

THE VALUES AND CORE PRINCIPLE AT ACM

Create and customize your favorite AI CHEF

CODE BASE

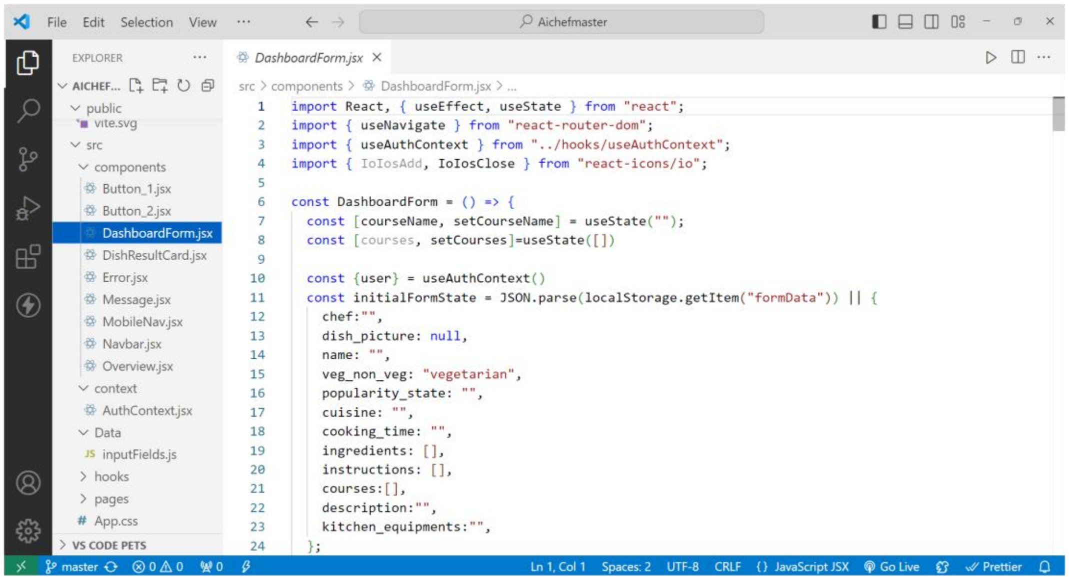

Navbar.jsx
src > components > Navbar.jsx > ...
Analyzing 'Navbar.jsx' and its dependencies

```
/* eslint-disable no-unused-vars */
import React from 'react'
import { Link, NavLink } from 'react-router-dom';
import { useLogout } from '../hooks/useLogout';
import MobileNav from './MobileNav';
import { useAuthContext } from '../hooks/useAuthContext';
import {  toast } from 'react-toastify';
const Navbar = () => {

  const {logout} = useLogout()
  const {user} = useAuthContext()
  const handleLogout=()=>{
    logout()
  }

  const handleDashboardClick=()=>{
    if(!user){
      toast.error('You need an account to access dashboard',{
        theme:"dark",
        position:"bottom-right"
      })
    }
  }

```



```
import React, { useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";
import { useAuthContext } from "../hooks/useAuthContext";
import { IoIosAdd, IoIosClose } from "react-icons/io";

const DashboardForm = () => {
  const [courseName, setCourseName] = useState("");
  const [courses, setCourses]=useState([])

  const {user} = useAuthContext()
  const initialFormState = JSON.parse(localStorage.getItem("formData")) || {
    chef:"",
    dish_picture: null,
    name: "",
    veg_non_veg: "vegetarian",
    popularity_state: "",
    cuisine: "",
    cooking_time: "",
    ingredients: [],
    instructions: [],
    courses:[],
    description:"",
    kitchen_equipments:"",
  };
```

Chapter - 7

CHALLENGES FACED AND SOLUTIONS

S.N.	Challenges	Solution
1.	Responsive Design Issues	Used **Tailwind CSS** to create flexible layouts and tested on multiple devices for consistency.
2.	Integrating AI Features with Frontend	Utilized **React's** state management and lifecycle methods (`useEffect`) for efficient data fetching and rendering.
3.	Cross-Browser Compatibility	Employed **CSS resets** and used tools like **BrowserStack** to test across different browsers and fix inconsistencies.
4.	Code Collaboration and Merge Conflicts	Regularly committed code, pulled the latest changes from the main branch, and resolved merge conflicts through **GitHub**.
5.	Debugging UI Issues	Leveraged **Chrome Developer Tools** and **Visual Studio Code's** debugging tools to identify and fix UI bugs.
6.	Performance Optimization	Compressed images, implemented **lazy loading**, and optimized JavaScript to improve page load times.
7.	Managing User Authentication	Integrated **Firebase** for secure user authentication and ensured proper token management for data security.
8.	Adapting to Agile Workflows	Actively participated in **sprint planning**, daily stand-ups, and retrospectives, managing tasks through **Jira**.
9.	UI/UX Design and User Experience	Worked closely with **UI/UX designers** to ensure the platform was visually appealing and user-friendly.

Chapter - 8

LEARNING AND GROWTH

1. During my internship at AI Chef Master, I gained valuable hands-on experience in **React** and **Tailwind CSS**. I learned to build responsive, dynamic web applications by creating reusable components and managing state effectively.
2. I also had the opportunity to integrate **AI features**, such as personalized recipe suggestions and ingredient substitutions. This taught me how to connect machine learning models to frontend applications.
3. A key takeaway was **responsive web design**. Using Tailwind CSS, I created mobile-first layouts that worked well across various devices and screen sizes.
4. I focused on **performance optimization**, improving website speed by compressing images, using lazy loading, and optimizing JavaScript for faster load times.
5. The internship allowed me to gain practical experience with **Git** and **GitHub**, where I managed code, collaborated with the team, and resolved merge conflicts.
6. I collaborated with **UI/UX designers** to ensure the platform was intuitive and visually appealing, applying user-centered design principles.
7. I enhanced my **problem-solving and debugging** skills by using **Chrome Developer Tools** and **Visual Studio Code** to troubleshoot and fix UI issues.
8. Finally, the internship strengthened my **team collaboration and communication** skills. I worked closely with developers, designers, and project managers, ensuring effective project delivery.

Chapter - 9

CONCLUSION

In conclusion, my three-month internship at AI Chef Master has been an enriching learning experience. I had the chance to work on cutting-edge technologies and contributed to building a dynamic AI-driven platform. This internship helped me sharpen my technical skills, such as React, Tailwind CSS, and version control with Git, while also enhancing my ability to work in a collaborative team environment. The exposure to real-world projects, Agile methodologies, and cross-functional teamwork has prepared me to take on more complex challenges in web development, making me confident in my future career path.

In the end I would like to thank all my mentors for this successful and staggering summer training . I am glad that I got to learn from you.

Once again thank you so much to everyone who was involved in this training period. I will remember each and everything taught in this session

Chapter - 10

FUTURE SCOPE AND OUTCOME

The future of frontend development is bright and constantly evolving, with several key trends shaping its trajectory. As websites and applications become more interactive and user-centric, frontend development will continue to focus on creating seamless, fast, and responsive user experiences. JavaScript frameworks like React, Vue, and Angular will remain at the forefront, while WebAssembly may allow for even faster web performance.

With the rise of AI and machine learning, frontend developers will increasingly integrate these technologies to create more personalized and dynamic user experiences. Additionally, progressive web apps (PWAs) will continue to bridge the gap between mobile apps and websites, offering offline capabilities and better performance.

The future also holds opportunities for front-end developers to work closely with UX/UI designers, creating user interfaces that are not only functional but also visually appealing. As the demand for rich, interactive web applications grows, frontend development will continue to be a dynamic and essential field in the tech industry.

REFERENCES

1. React - https://legacy.reactjs.org/docs/getting-started.html
2. Tailwind CSS - https://tailwindcss.com/
3. Github - https://docs.github.com/en
4. W3schools - https://www.w3schools.com/
5. Material Tailwind Documentation: https://www.material-tailwind.com/
6. Vite: https://vite.dev/
7. VS Code: https://code.visualstudio.com/