A

Project Report on

INTENRSHIP PRESENTATION

submitted in partial fulfillment for the award of the degree of Bachelor of Technology

in

Computer Science & Engineering

Submitted By:

RAVI RANJAN(Q_id:22030209)

Project Supervisor:

Ms. Neetu Mourya

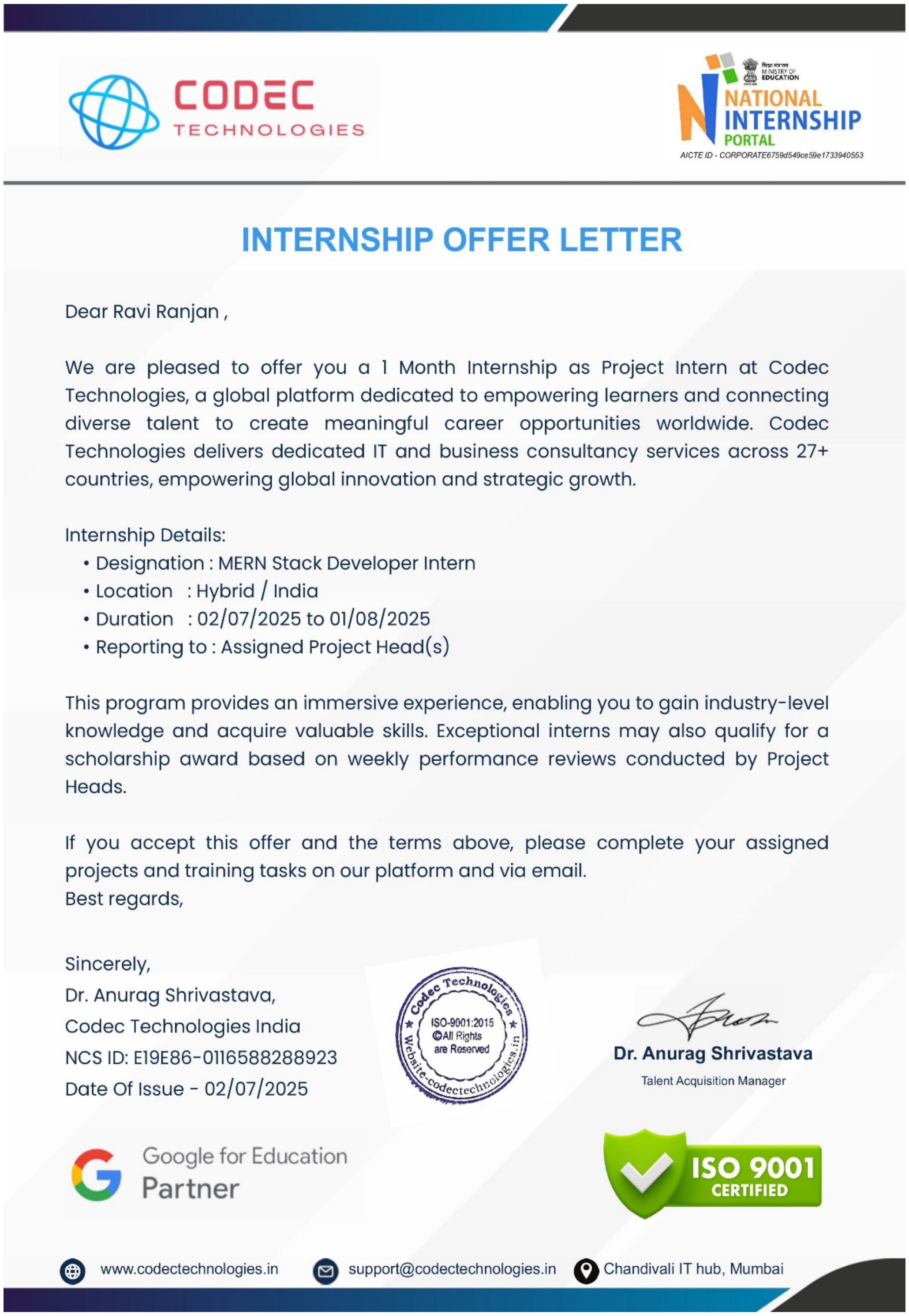
Department of Computer Science and Engineering

School of Technology

Quantum University, Roorkee

Dec., 2025

Internship Offer Letter



CODEC TECHNOLOGIES

शिक्षा मंत्रालय MINISTRY OF EDUCATION
NATIONAL INTERNSHIP PORTAL
AICTE ID - CORPORATE6759d549ce59e1733940553

INTERNSHIP OFFER LETTER

Dear Ravi Ranjan ,

We are pleased to offer you a 1 Month Internship as Project Intern at Codec Technologies, a global platform dedicated to empowering learners and connecting diverse talent to create meaningful career opportunities worldwide. Codec Technologies delivers dedicated IT and business consultancy services across 27+ countries, empowering global innovation and strategic growth.

Internship Details:

- Designation : MERN Stack Developer Intern
- Location : Hybrid / India
- Duration : 02/07/2025 to 01/08/2025
- Reporting to : Assigned Project Head(s)

This program provides an immersive experience, enabling you to gain industry-level knowledge and acquire valuable skills. Exceptional interns may also qualify for a scholarship award based on weekly performance reviews conducted by Project Heads.

If you accept this offer and the terms above, please complete your assigned projects and training tasks on our platform and via email.
Best regards,

Sincerely,
Dr. Anurag Shrivastava,
Codec Technologies India
NCS ID: E19E86-0116588288923
Date Of Issue - 02/07/2025

Codec Technologies ★ ISO-9001:2015 ©All Rights are Reserved ★ Website-codectechnologies.in

[illegible]
Dr. Anurag Shrivastava
Talent Acquisition Manager

Google for Education Partner

ISO 9001 CERTIFIED

www.codectechnologies.in support@codectechnologies.in Chandivali IT hub, Mumbai

DECELERATION

I hereby declare that the internship report entitled **" MERN STACK DEVELOPER"** submitted for the B. Tech Degree in Computer Science and

Engineering is my original work and the project has not formed the basis or submitted for the award of any degree, diploma, or any other similar titles in any other college / institute / university.

Name: Ravi Ranjan

Signature: ________________

Place: ________________

Date: ________________

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to CODEC TECHNOLOGY & Engineering Services for providing me with the opportunity to undergo my Summer Internship in Frontend Development. The exposure to professional software development practices, specifically in React and the MERN stack, has been invaluable for my career growth and has bridged the gap between academic theory and industry application.

I am deeply grateful to Dr. A. Srivastav (Sr. HR-Manager) for his support and for facilitating a smooth onboarding process. I also extend my thanks to the technical team at Bridge4Engineers.com, the official Technology Partner for this internship, for their technical mentorship and guidance throughout the diverse projects I undertook. Their code reviews and architectural insights were instrumental in the successful completion of my projects.

I would also like to thank my Head of Department and my internal guide at QUANTUM UNIVERSSITY for their constant encouragement and for providing the necessary academic support to complete this report.

Finally, I thank my parents and friends for their unwavering support throughout this journey.

RAVI RANJAN Roll No: [22030209]

ABSTRACT

This report documents the work undertaken during the 30-day Online Summer Internship in Frontend Development at **CODEC TECHNOLOGY**, Noida, from **2 JULY 2025 to 1 AUGUST 2025**.

The primary objective of this internship was to gain hands-on experience in modern web development technologies and to understand the complete software development lifecycle (SDLC) in a corporate environment. Over the course of four weeks, I developed three distinct applications that progressively increased in complexity, allowing for a structured learning path:

WORD MASTER: A React-based utility tool using the Dictionary API. This project focused on

DOM manipulation, audio handling for pronunciation, and implementing advanced UI features like dynamic theming (Dark/Light mode) and font customization.

MOVIE MATRIX: A dynamic movie search application utilizing React, Tailwind CSS, and the

OMDB API. This project emphasized responsive design, asynchronous data fetching using Axios, and handling complex JSON data structures to display movie details, ratings, and visual assets.

AUTH FLOW SYSTEM : A robust full-stack application built using the MERN stack (React,

Node.js). This capstone project featured real-time secure user authentication using JSON Web Tokens (JWT), and global state management via Zustand.

This report details the technologies used, including **React.js, Tailwind CSS, Axios, Zustand, and JWT**, and outlines the agile methodologies and testing strategies followed during the creation of these projects.

CHAPTER 1: INTRODUCTION

1.1 Company Profile

CODEC TECHNOLOGY is a prominent technology solutions provider located at **D-107, 91Springboard, Vyapar Marg, Sector-2, Noida, UP 201301**. The company specializes in delivering comprehensive IT and engineering services to a diverse client base, ranging from startups to established enterprises. For this internship program, CODEC TECH. . This partnership ensured that interns were not only working on theoretical concepts but were also exposed to industry-standard workflows, code quality checks, and deployment strategies.

1.2 Project Overview

The internship was structured to provide a comprehensive learning curve, moving from basic API integration to full-stack development. This "crawl, walk, run" approach ensured a solid foundation before tackling complex architectural challenges.

Project 1: Word Master: This was an introductory project designed to solidify React fundamentals. It is a utility application designed to provide definitions, phonetics, and audio pronunciations for English words. Beyond basic functionality, it features a user-friendly interface with Dark/Light mode toggles and font customization, teaching the importance of user preference state management.

Project 2: Movie Matrix: Building on the basics, this project introduced external styling libraries and complex data handling. It is a dynamic movie search engine that interacts with the Open Movie Database (OMDB) to fetch and display movie details, ratings, and posters. The focus here was on creating a visually immersive experience similar to modern streaming platforms.

Project 3: Auth Flow System : The final capstone project was a robust full-stack user authen tication platform. It allows users to sign up, authenticate securely, and exchange messages instantly. This required integrating a custom backend with a React frontend, managing database connections, and handling websocket events for live communication.

1.3 Objectives

The internship was driven by several key technical and professional objectives:

- **Mastering React.js:** To understand the component-based architecture, virtual DOM rendering, and the lifecycle of React components.
- **Advanced Styling:** To learn styling with utility-first frameworks like **Tailwind CSS**, moving away from traditional CSS files to speed up development.
- **State Management:** To understand the limitations of local state (useState) and implement global state management using **Zustand** for complex applications.
- **Real-Time Communication:** To implement **Socket.io** for bi-directional event-based communication, essential for modern interactive apps.
- **Backend & Security:** To build and test RESTful APIs using **Postman** and secure them with **JWT** (JSON Web Tokens) to understand authentication flows.
- **Professional Development:** To adapt to remote work protocols, manage time effectively, and communicate technical blockers clearly.

CHAPTER 2: LITERATURE SURVEY (TECHNOLOGY STACK)

2.1 Frontend Technologies

The frontend selection focused on the modern JavaScript ecosystem, prioritizing performance and developer experience.

- **React.js (JSX):** React was used as the core library for building the user interfaces across all three projects. Its Virtual DOM mechanism ensures high performance by only updating changed parts of the UI. The use of JSX (JavaScript XML) allowed for writing HTML structures directly within JavaScript code, making component logic selfcontained.
- **Tailwind CSS:** For the "Streamflix" project, Tailwind CSS was chosen over Bootstrap. As a utility-first CSS framework, it allows for rapid UI development by applying predefined classes directly in HTML. This approach reduces the bundle size by removing unused CSS and ensures a highly responsive design mobile-first approach.
- **Axios:** While the native Fetch API is available, Axios was preferred for its automatic JSON transformation, better error handling, and support for interceptors. It was used to fetch data from external APIs (Dictionary API, OMDB API) and the custom backend.
- **Zustand:** For the Chat App, managing state across multiple components (e.g., user list, current chat, online status) became complex. Zustand was selected as a lightweight alternative to Redux. It provides a simple, hook-based API to manage global state without the boilerplate code often associated with other state management libraries.

2.3 Tools & APIs

- **OMDB API:** A RESTful web service to obtain movie information. It was used in Streamflix to retrieve metadata such as plot summaries, release years, and poster URLs.
- **Dictionary API:** A free API used to fetch word meanings, synonyms, and audio files for pronunciation.

CHAPTER 3: METHODOLOGY

3.1 Development Process (Week-wise Breakdown)

The internship followed an Agile-inspired methodology, broken down into weekly sprints focusing on specific technologies and deliverables.

Week 1: React Fundamentals & Word Master

- **Focus:** The first week was dedicated to understanding React Hooks (specifically useState for local state and useEffect for side effects) and API consumption.
- **Activity:** I built the Dictionary App from scratch. The primary challenge was handling the complex JSON structure returned by the API, which included nested arrays for definitions and phonetics.
- **Key Features Implemented:**
    - **Search Functionality:** An input field that triggers an API call on submission**.**
    - **Audio Playback:** Integrated HTML5 Audio API to play phonetic pronunciations returned by the server.
    - **Theme Switcher:** Implemented logic to toggle between Light and Dark modes using CSS variables and React state.
    - **Font Selection:** A dropdown menu allowing users to switch between Serif, SansSerif, and Monospace fonts, dynamically updating the CSS of the entire application.

Week 2: Advanced Styling & Movie Matrix Project

- **Focus:** The second week shifted focus to UI/UX design, responsive layouts, and handling larger datasets.
- **Activity:** I developed **Streamflix**, a movie search engine. The goal was to replicate the look and feel of a professional streaming site.
- **Key Features Implemented:**

  - **API Integration:** Connected to the **OMDB API** using a unique API key to fetch movie data based on search queries.
  - **Asynchronous Handling:** Used **Axios** with async/await syntax to handle HTTP requests cleanly, including error handling for "Movie Not Found" scenarios.
  - **Modern Styling:** Styled the application entirely with **Tailwind CSS**. This involved using grid layouts (grid-cols-1 md:grid-cols-3) to display movie posters responsively across mobile and desktop devices.
  - **Dynamic Components:** Created reusable "MovieCard" components to efficiently render lists of search results.

Week 3 & 4: Auth Flow System

- **Focus:** The final two weeks were the most intensive, focusing on full-stack development, security protocols, and real-time logic.
- **Activity:** I learned MERN stack concepts and built the **Chat Application**. This required setting up a server, a database, and connecting them to the React frontend.
- **Key Features Implemented:**
  - **Backend Setup:** Configured a Node/Express server and established a connection to a MongoDB Atlas cluster.
  - **Authentication System:** Implemented a secure Signup/Login flow. Passwords were hashed using bcrypt before storage, and **JWTs** were issued upon successful login to maintain user sessions.
  - **Real-time Logic:** Integrated **Socket.io** on both client and server. This allowed the server to listen for "new message" events and broadcast them to the specific receiver instantly.
  - **State Management:** Adopted **Zustand** to handle complex global states, such as the list of online users and the currently selected chat conversation.
  - **Media Handling:** Integrated **Cloudinary** for uploading user profile images, ensuring optimized delivery of assets.
  - **API Testing:** Verified all backend routes using **Postman** to ensure data integrity before frontend integration.

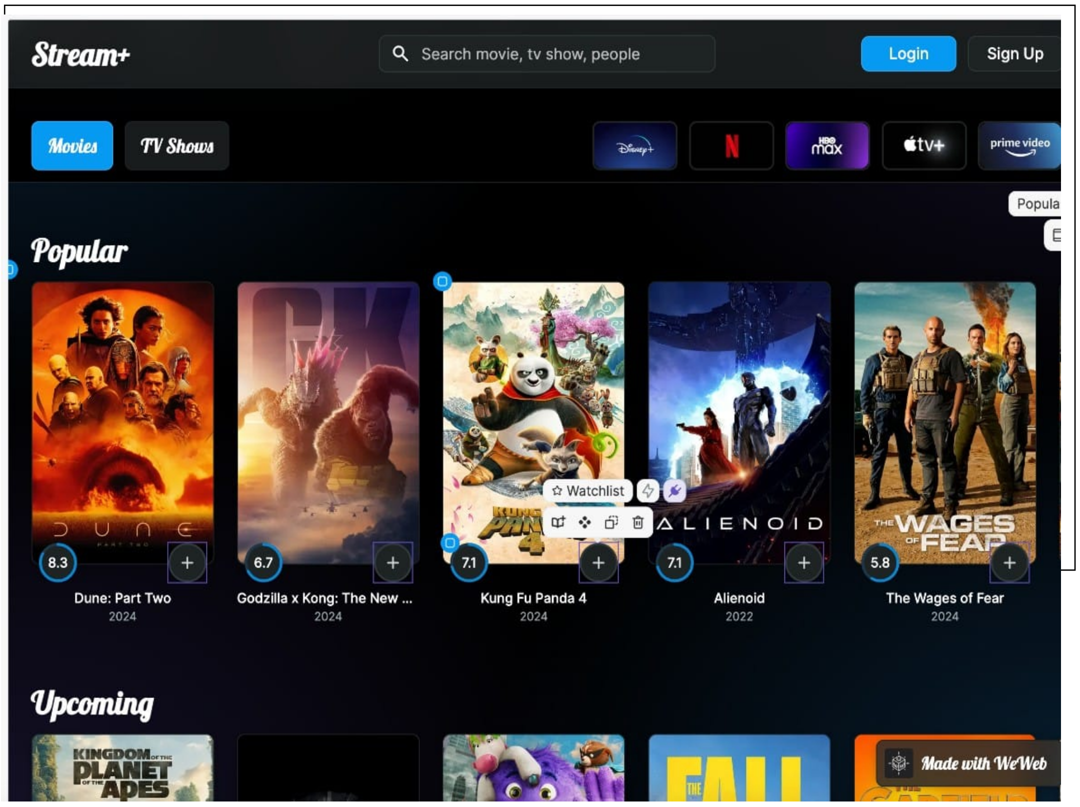
CHAPTER 4: SOFTWARE DESIGN

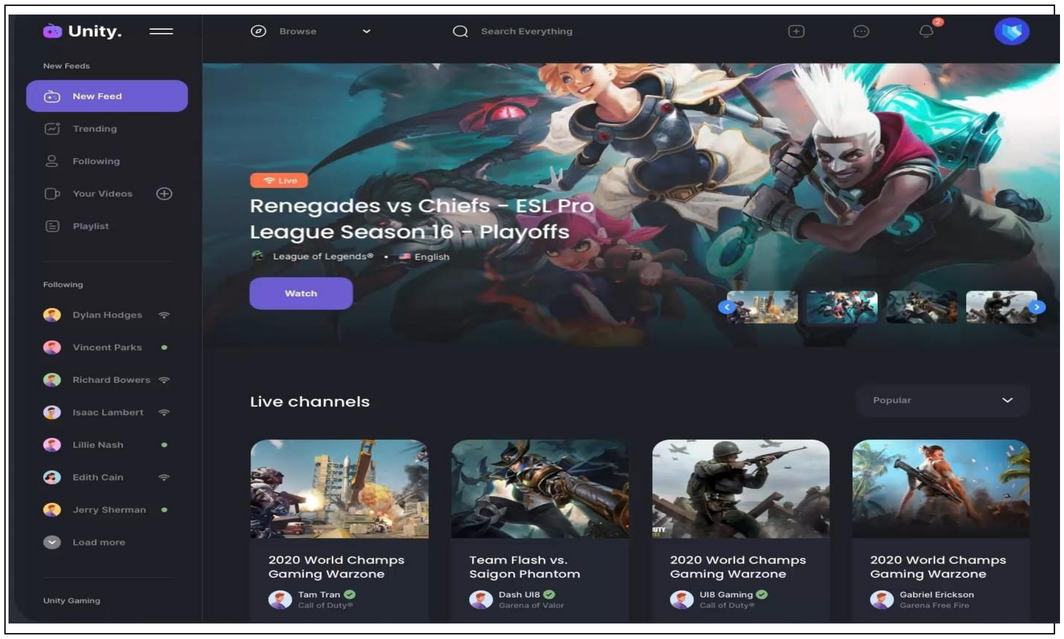
4.2 Data Flow Diagrams

Figure 4.1: Movie matrix Data Flow The data flow in Movie matrix is linear and request-based:

1. **Input:** User enters a Movie Title in the search bar.
2. **Request:** React Component triggers an Axios GET request to the OMDB API endpoint.
3. **Processing:** OMDB API processes the query and returns a JSON object containing details like Title, Poster URL, and IMDb Rating.
4. **Display:** React updates its state with this data, causing the component to re-render and display the Movie Cards.

Figure 4.1: Movie Matrix





4.3 UI Design Principles

- **Dictionary App:** The design philosophy here was "Content First." The focus was on typography and readability, implementing high contrast ratios for the dark mode and offering font toggles to suit user reading preferences.

- **Streamflix:** The design aimed for "Visual Impact." It utilized a dark-themed aesthetic (using Tailwind colors like bg-slate-900 and text-gray-100) to make the movie posters pop, similar to Netflix or Hulu.

CHAPTER 5: IMPLEMENTATION

5.1 Project 1: Word Master (Audio Feature)

The audio feature was a key interactive element. Instead of using a heavy external library, I utilized the native HTML5 Audio API. This snippet demonstrates how the audio URL fetched from the API is played when the user clicks the play button.

```
const playAudio = () => {

   // data.phonetics[0].audio contains the URL to the mp3 file
const audio = new Audio(data.phonetics[0].audio);

   audio.play();

};

return (

   <button    onClick={playAudio}        className="play    -      aria-label="Play
btn"
pronunciation">

      <i className="fas fa-play"></i>

   </button>

);
```

5.2 Project 2: Streamflix (Axios & OMDB)

This implementation highlights the asynchronous nature of modern web apps. The searchMovies function waits for the API response before updating the state, ensuring the UI doesn't break while data is loading.

```
import axios from 'axios';


const searchMovies = async (title) => {

   // Constructing the API URL with the secure key

   const          API_URL      =

'[http://www.omdbapi.com/?apikey=](http://www.omdbapi.com/?apikey=)[MY_KEY
]';
  try {
      const response = await axios.get(`${API_URL}&s=${title}`);
if(response.data.Search) {

          setMovies(response.data.Search);
      }
  } catch (error) {
      console.error("Error fetching movies:", error);
  }
};
```

5.3 Project 3: Multi Step Form

This section demonstrates the integration of real-time sockets with global state management. Zustand simplifies the store creation, while the useEffect hook ensures the socket listener is established only once when the component mounts.

```
// store/useChatStore.js (Zustand Store)

import { create } from 'zustand';

export const useChatStore = create((set) => ({

  messages: [],

  // Action to append a new message to the existing array

  addMessage: (msg) => set((state) => ({ messages: [...state.messages, msg] })),

  selectedUser: null,

  setSelectedUser: (user) => set({ selectedUser: user }),

}));

// Socket connection in the Chat Component
```

```
useEffect(() => {
   // Connect to the backend socket server
   const socket = io("http://localhost:5000");
   // Listen for incoming 'newMessage' events from the server
   socket.on("newMessage", (newMessage) => {
      // Update the global store immediately
      addMessage(newMessage);
   });
   // Cleanup listener on unmount
   return () => socket.off("newMessage");
```

CHAPTER 6: RESULTS AND DISCUSSIONS

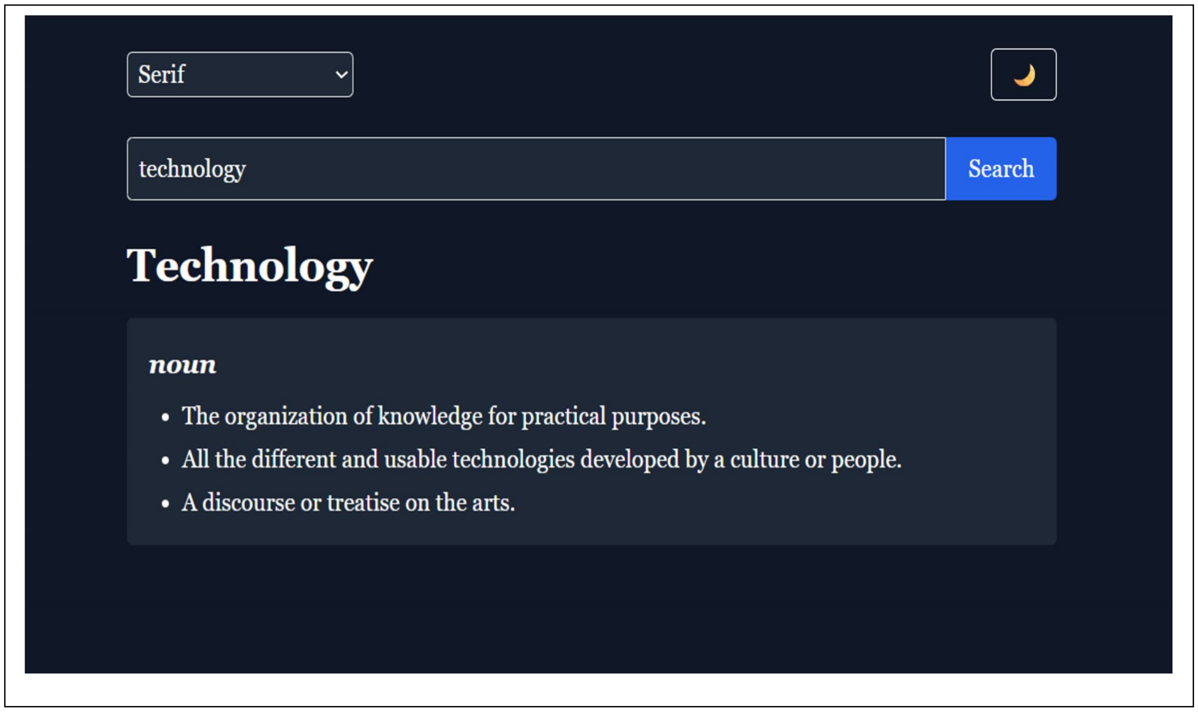
6.1 Testing Strategy

Testing was an integral part of the development lifecycle to ensure robustness.

- **API Testing (Black Box):** All backend routes (Login, Signup, Send Message) were rigorously tested using **Postman** collections. This allowed me to verify HTTP status codes (200 OK, 401 Unauthorized, 500 Server Error) and JSON response structures independent of the frontend.
- **Component Testing:** React components were manually tested for responsiveness across different screen sizes using Chrome DevTools.

6.2 Output Screens

Figure 6.1: Dictionary App Interface (Insert screenshot showing the dictionary interface with the font selector dropdown open and a word definition displayed).



technology

Technology

noun

- The organization of knowledge for practical purposes.
- All the different and usable technologies developed by a culture or people.
- A discourse or treatise on the arts.

Figure 6.2: Movie Matrxi Home Page (Insert screenshot showing a grid of movie posters fetched

from OMDB, demonstrating the responsive grid layout).

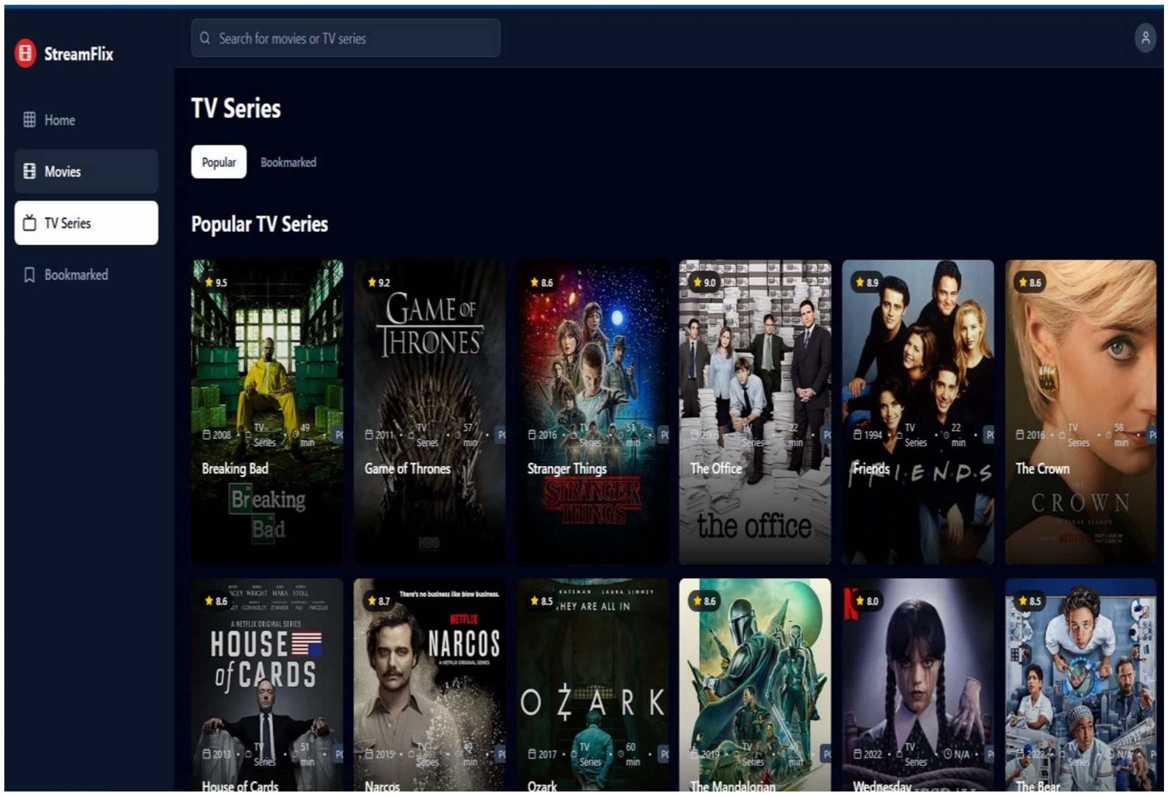
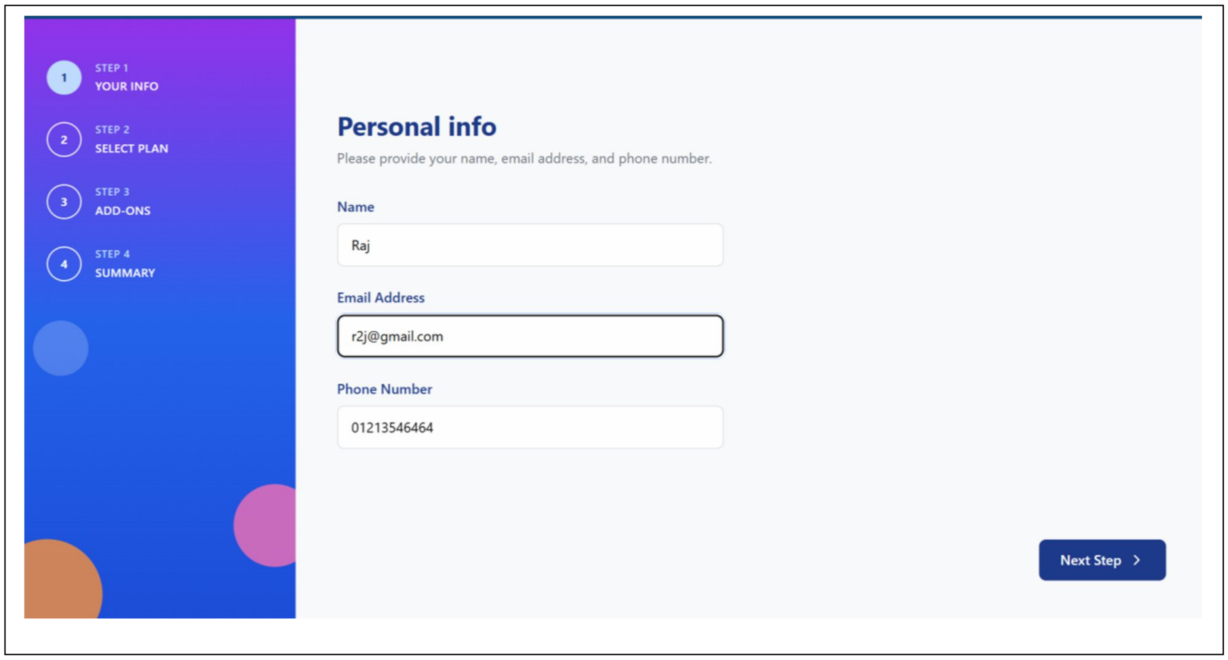


Figure 6.3: Chat App Real-time Interface (Insert screenshot showing the chat window, the active user list on the left, and real-time message bubbles on the right).



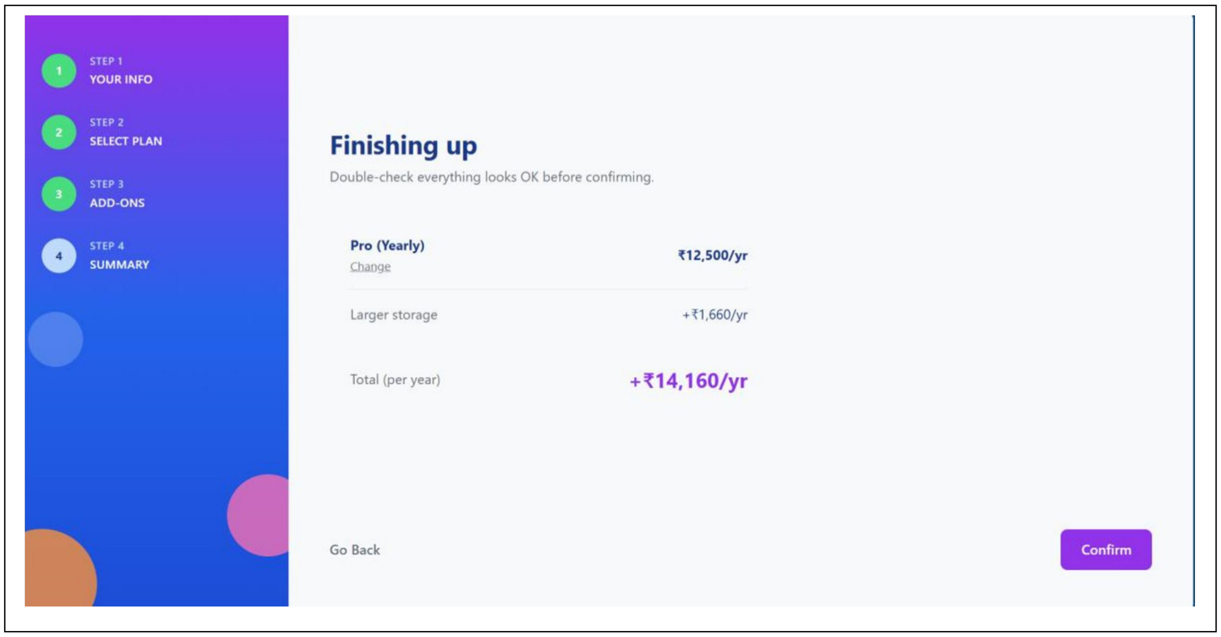
Personal info

Please provide your name, email address, and phone number.

Name: Raj

Email Address: r2j@gmail.com

Phone Number: 01213546464



Finishing up

Double-check everything looks OK before confirming.

Pro (Yearly)	₹12,500/yr
Larger storage	+₹1,660/yr
Total (per year)	+₹14,160/yr

CHAPTER 7: CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

The 30-day internship at CODEC TECH. was an intensive and transformative learning experience. Starting from a simple Dictionary App, I progressed to complex API handling in Streamflix, and finally mastered full-stack concepts with the Real-time Chat App.

- **State Management Mastery**: I successfully learned to manage state globally with Zustand, moving beyond simple prop-drilling. This skill is essential for building scalable, large-scale applications.
- **Backend Proficiency:** I gained confidence in backend logic using Node.js and Express. Understanding how to secure APIs with JWT gave me a holistic view of web security.
- **Real-Time Capabilities:** The exposure to Socket.io demystified how live applications (like WhatsApp or Discord) function, adding a valuable skill to my repertoire.
- **Soft Skills**: Working remotely taught me self-discipline, the importance of clear documentation, and how to communicate technical issues effectively.

7.2 Challenges Faced

- **Socket.io CORS Errors**: Initially, I faced connection refusals between the frontend (port 3000) and backend (port 5000). This was solved by configuring Cross-Origin Resource Sharing (CORS) correctly on the Express server.
- **State Management Paradigm:** Transitioning from React's useState to Zustand required a shift in thinking, but it ultimately simplified the code structure and improved performance by preventing unnecessary re-renders.

7.3 Future Scope

The projects developed during this internship have significant potential for expansion:

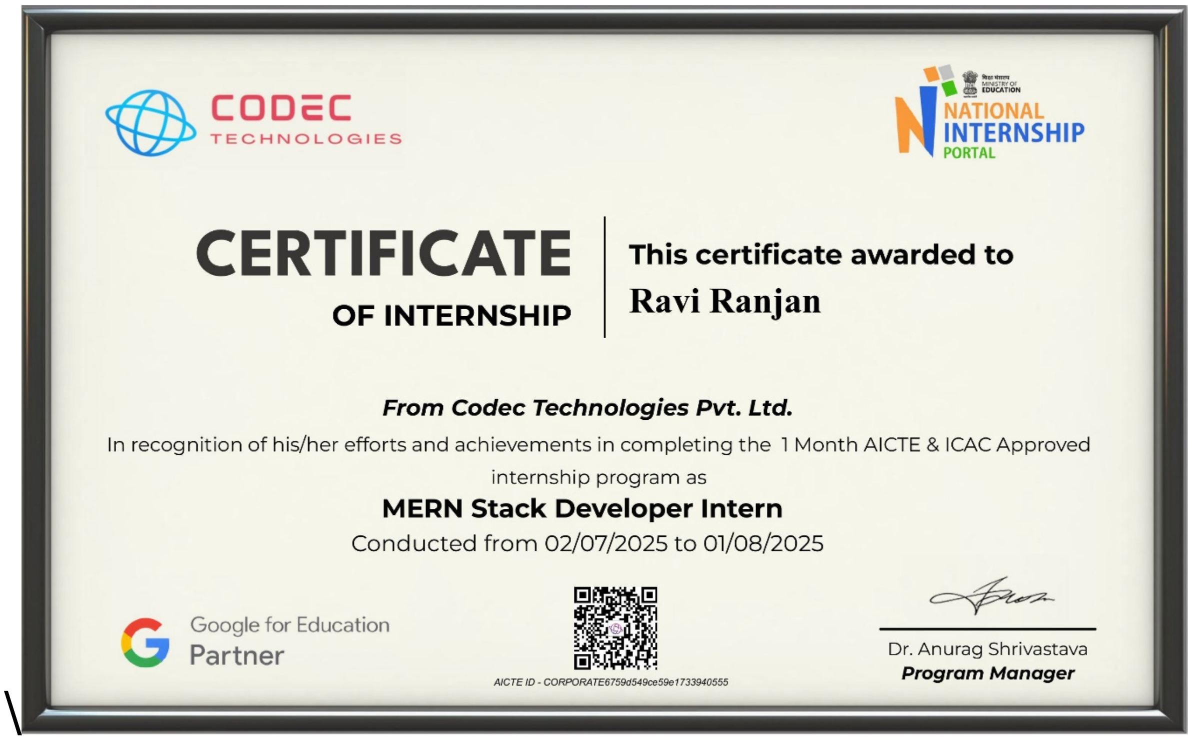
- **Streamflix Enhancements:** I plan to add a "Watchlist" feature using localStorage or a database, allowing users to save movies for later. Additionally, integrating a trailer playback feature would improve user engagement.
- **Chat App Upgrades:** Future versions could implement **End-to-End encryption** for messages to ensure privacy. I also aim to add support for group chats and file sharing (PDFs, docs) beyond just images.

REFERENCES

- React Documentation: *react.dev* - Used for understanding Hooks, Context API, and Component Lifecycle methods.
- Tailwind CSS Documentation: *tailwindcss.com* - Referenced for utility class names, responsive design patterns, and configuration.
- OMDB API Documentation: *omdbapi.com* - Used to understand API parameters, query structures, and JSON response formats for the movie application.
- Socket.io Documentation: *socket.io/docs/v4* - Essential for implementing real-time event listeners and broadcasting events between client and server.
- Zustand GitHub Repository: *github.com/pmndrs/zustand* - Used for implementing global state management and understanding store configuration.
- Node.js Official Documentation: *nodejs.org/en/docs* - Referenced for understanding the runtime environment, file system modules, and asynchronous event handling.
- Express.js Routing Guide: *expressjs.com/en/guide/routing.html* - Used to structure the backend REST API routes and implement middleware functions.
- MongoDB Atlas Documentation:

  https://www.google.com/search?q=www.mongodb.com/docs/atlas - Consulted for database connection strings, collection management, and CRUD operations.
- JSON Web Token (JWT) Introduction: *jwt.io/introduction* - Used to understand the structure of JWTs (Header, Payload, Signature) and secure authentication flows.
- Axios HTTP Client: *axios-http.com/docs/intro* - Referenced for making asynchronous HTTP requests, handling interceptors, and error processing.
- MDN Web Docs (Mozilla): *developer.mozilla.org* - The primary reference for standard HTML5, CSS3 properties, and modern JavaScript (ES6+) syntax.
- Pressman, Roger S. "Software Engineering: A Practitioner's Approach," 8th Edition, McGraw-Hill Education. (Referenced for Agile SDLC and Testing strategies).

Certificate



CODEC TECHNOLOGIES

MINISTRY OF EDUCATION

NATIONAL INTERNSHIP PORTAL

CERTIFICATE

OF INTERNSHIP

This certificate awarded to

Ravi Ranjan

From Codec Technologies Pvt. Ltd.

In recognition of his/her efforts and achievements in completing the 1 Month AICTE & ICAC Approved internship program as

MERN Stack Developer Intern

Conducted from 02/07/2025 to 01/08/2025

Google for Education Partner

AICTE ID - CORPORATE6759d549ce59e1733940555

[illegible]

Dr. Anurag Shrivastava

Program Manager

\