Work Done Report – OTP Authentication Project

Name: Mahendar BS
Qualification: M.Tech, IIT Kanpur
Year of Graduation: 2025

Overview

I designed and implemented a basic full-stack web application that demonstrates OTP-based authentication. The project was developed to understand frontend–backend integration, REST APIs, and authentication flow using modern web technologies.

Backend Development

I created the backend using **Node.js and Express.js**.
The backend runs on **port 5000** and exposes REST API endpoints for authentication.

I implemented an API to generate a **6-digit One-Time Password (OTP)** when a user submits an email ID or phone number. Since this is a demo project, OTP delivery is mocked by printing the OTP in the backend console.

I also implemented OTP verification logic where:

- The user can attempt OTP verification a maximum of **three times**
- After three failed attempts, the user is **blocked for 10 minutes**
- On successful verification, a **mock session token (UUID)** is generated and returned

User data, OTPs, and attempt counts are stored in **in-memory objects** for simplicity.

Frontend Development

I developed the frontend using **React.js**, running on **port 3000**.

The frontend consists of:

- A **Login page** where the user enters email or phone number
- An **OTP verification page** to submit the received OTP
- A **Welcome page** displayed after successful login

The frontend communicates with the backend using the **Fetch API** and handles responses such as success, invalid OTP, and blocked user messages. The authentication token is stored in **localStorage** to maintain login state.

Integration and Testing

I configured **CORS** to allow communication between frontend (port 3000) and backend (port 5000).

I tested the APIs using:

- PowerShell (Invoke-RestMethod)
- Browser-based frontend forms

I verified:

- OTP generation and logging in backend console
- OTP validation and error handling
- Proper redirection to welcome page after login

Challenges Faced and Resolved

- Resolved **port conflict issues (EADDRINUSE)** by identifying running services
- Debugged frontend caching issues by restarting the React server and clearing cache
- Fixed API connection errors by correcting endpoint URLs and request formats
- Understood the difference between **terminal**, **console**, and backend logs

Tools and Technologies Used

- Node.js
- Express.js
- React.js
- JavaScript
- REST APIs
- UUID
- dotenv
- PowerShell
- Visual Studio Code

Conclusion

Through this project, I successfully built and tested an end-to-end OTP authentication system. This work helped me strengthen my understanding of full-stack development, API integration, authentication mechanisms, and real-world debugging practices.