

**CHANDIGARH  
UNIVERSITY**  
Discover. Learn. Empower.

**NAAC  
GRADE A+**  
Accredited University

## **UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

**(BE-CSE/IT-6<sup>th</sup> Sem)**



**Subject Name: Advanced Programming Lab - 2**

**Subject Code: 22CSP-351**

**Submitted to:**

Er.Nandita Malik  
(E9599)

**Submitted by:**

**Name:** utkarsh Kumar

**Uid:** 22BCS14658

**Section:** 22EPAM-802

**Group:** B



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment 1

**Student Name:** Utkarsh Kumar

**UID:** 22BCS14658

**Branch:** BE- CSE

**Section/Group:** 22EPAM-802

**Semester:** 6<sup>th</sup>

**Date of Performance:** 06/01/2025

**Subject Name:** Advanced Programming LAB 2

**Subject Code:** 22CSP-351

**1. Aim:** The primary aim of this experiment is to provide students or developers with an understanding of full-stack development involving MongoDB, Node.js, React, and Express.

**2. Objective:** Learn about-

- MongoDB
- ExpressJS
- ReactJS
- NodeJS

**3. Implementation/Code:**

Before starting, ensure the following are installed on your computer: - **Node.js** (v16 or higher) - **npm** (v8 or higher) - **MongoDB** (local or cloud-based) - **Git** (for cloning the repository)

### **Step 1: Clone the Repository**

1. Open your terminal or command prompt.
2. Clone the repository using the following command:  
`git clone https://github.com/lonethinker/AP2-LoginPage-MERN.git`
3. Navigate to the project folder:  
`cd AP2-LoginPage-MERN`

### **Step 2: Set Up the Backend**

1. Navigate to the backend folder:  
`cd backend`
2. Install the backend dependencies:  
`npm install`
3. Set up environment variables:
  - Create a .envfile in the backend folder.

- Add the following variables to the .envfile:  
MONGO\_URI=mongodb://localhost:27017/mern-login-signup  
JWT\_SECRET=your\_jwt\_secret\_key  
PORT=5000  
Replace your\_jwt\_secret\_key with a secure secret key for JWT.
- 4. Start the backend server:  
npm start  
The backend server will run on http://localhost:5000.

### Step 3: Set Up the Frontend

1. Open a new terminal window and navigate to the frontend folder:  
cd ../frontend
2. Install the frontend dependencies:  
npm install
3. Start the frontend development server:  
npm start  
The frontend will run on http://localhost:3000.

### Step 4: Access the Application

1. Open your browser and navigate to http://localhost:3000.
2. You should see the **Login/Signup App** homepage.

### Step 5: Test the Application

1. **Signup:**
  - Click on the **Signup** button.
  - Enter your name, email, and password.
  - Click **Signup** to create a new account.
2. **Login:**
  - Click on the **Login** button.
  - Enter your email and password.
  - Click **Login** to access your account.

### Step 6: (Optional) Test the API Using Postman

If you want to test the backend API directly, follow these steps:

1. **Signup:**
  - Method: POST
  - URL: http://localhost:5000/signup
  - Body (JSON):

```
{
  "name": "John Doe", "email":
  "john@example.com",
  "password": "password123"
}
```
2. **Login:**



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

- Method: POST
  - URL: `http://localhost:5000/login`
  - Body (JSON):

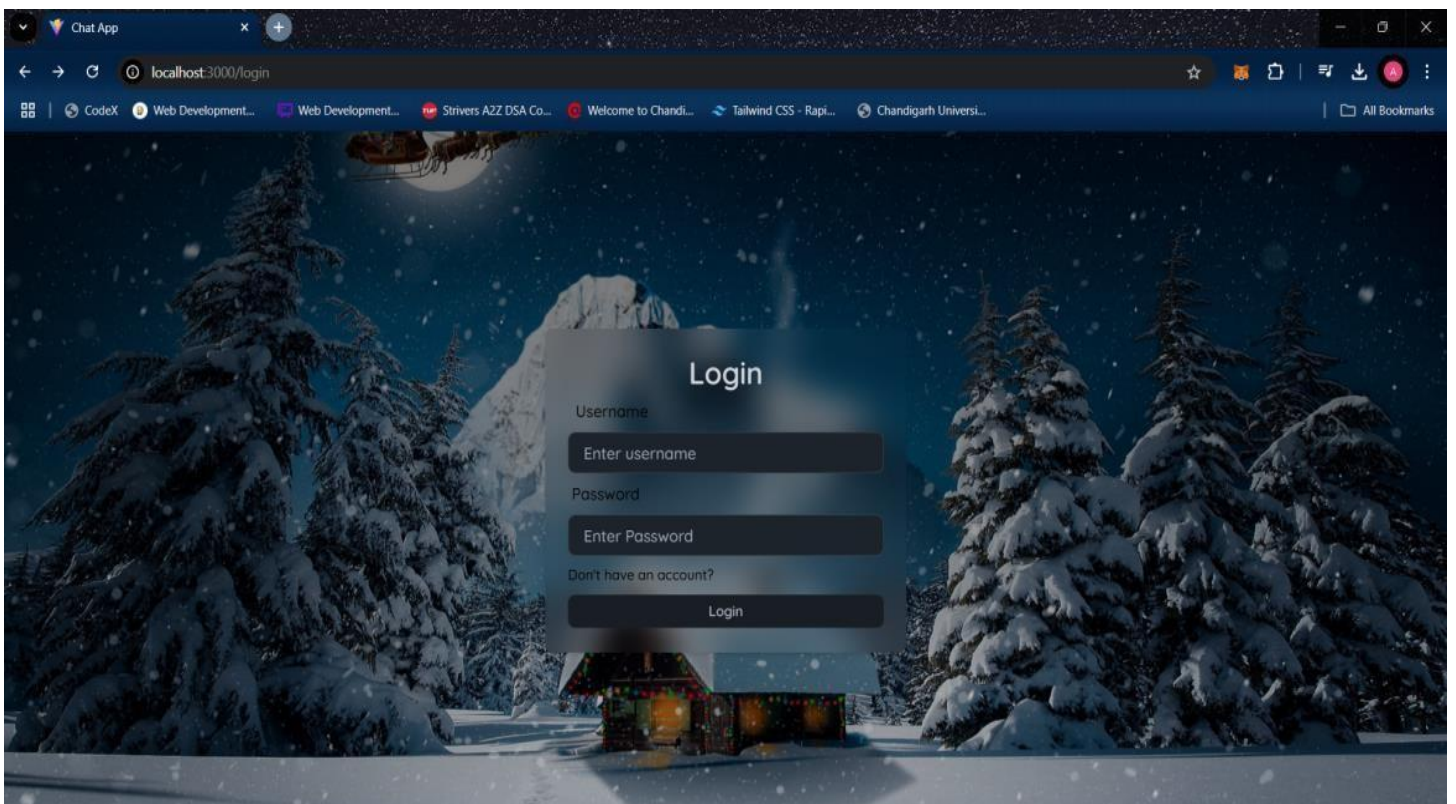
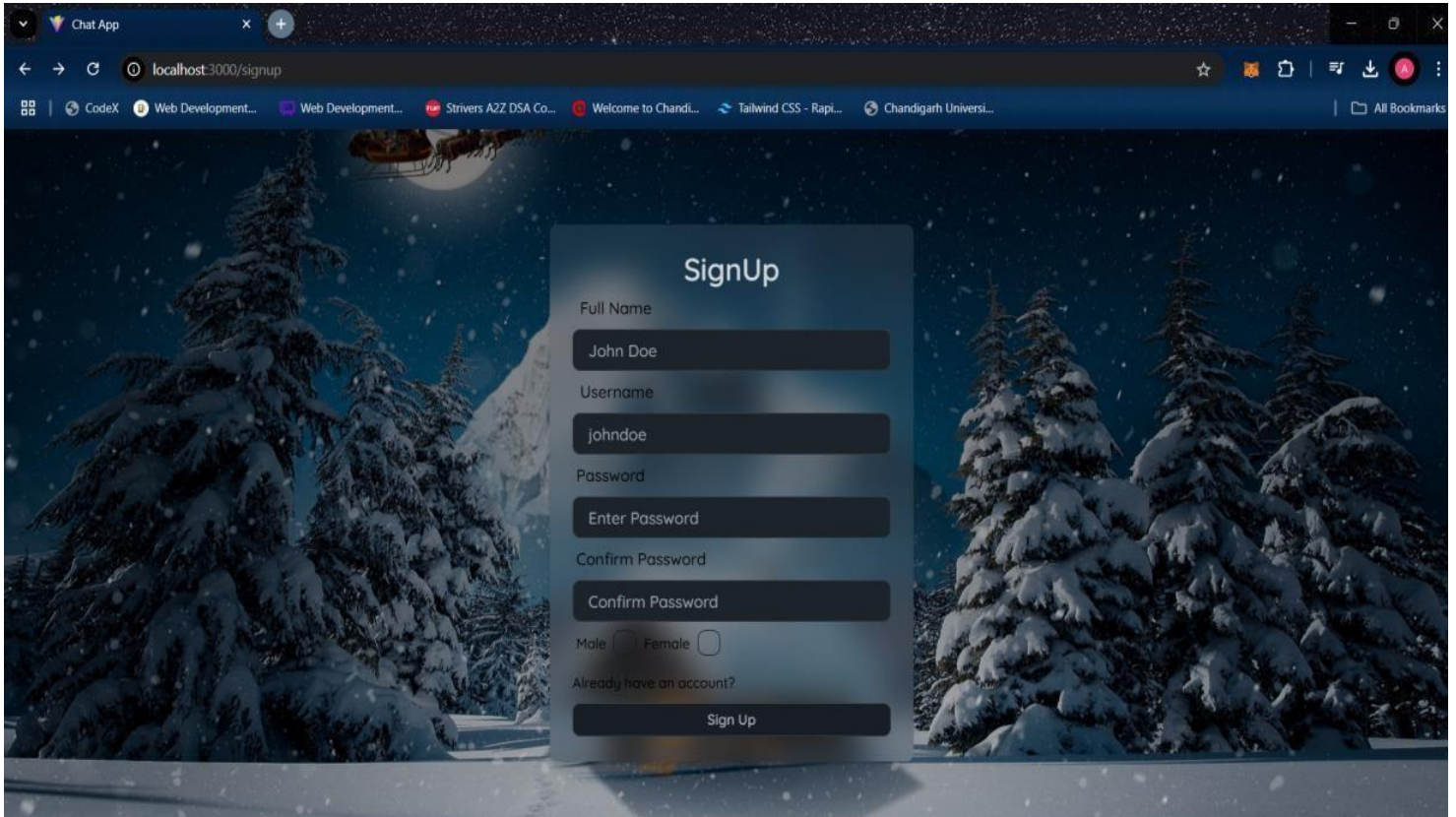
```
{  "email": "john@example.com",  "password": "password123"}
```
3. **Protected Route** (e.g., Dashboard):
- Method: GET
  - URL: `http://localhost:5000/dashboard`
  - Headers:
    - Authorization: Bearer <your-jwt-token>

## Project Structure

### AP2-LoginPage-MERN/

└─ backend/	
├─ models/	# MongoDB models
├─ routes/	# API routes
├─ controllers/	# Route controllers
├─ middleware/	# Custom middleware (e.g., authentication)
├─ config/	# Database configuration
├─ .env	# Environment variables
├─ server.js	# Backend entry point
└─ package.json	# Backend dependencies
└─ frontend/	
├─ public/	# Static assets
├─ src/	
│ └─ components/	# React components
│ └─ App.js	# Main React component
│ └─ index.js	# Frontend entry point
│ └─ App.css	# Global styles
└─ package.json	# Frontend dependencies
└─ .gitignore	# Files and folders to ignore
└─ README.md	# Project documentation

## 4. Output:







## 5. Learning Outcomes:

- Develop a comprehensive understanding of MongoDB's role as a NoSQL database, including how to design and manage databases, perform CRUD operations, and utilize advanced features like aggregation and indexing within the context of full-stack applications.
- Gain proficiency in using ExpressJS to design and implement RESTful APIs, manage middleware, handle routing, and build the server-side architecture required to support dynamic web applications.
- Master the fundamentals of ReactJS, focusing on building interactive, reusable components, managing application state with hooks, and effectively utilizing React's virtual DOM to create efficient and responsive user interfaces.
- Build a solid foundation in NodeJS by understanding its non-blocking, event-driven architecture, and how to leverage its asynchronous capabilities to create scalable and high-performance server-side applications.
- Develop hands-on experience in integrating MongoDB, ExpressJS, ReactJS, and NodeJS (the MERN stack) to construct end-to-end full-stack applications, including setting up communication between the front-end and back-end, managing user authentication, and deploying a complete application.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

