COLLEGE CODE : 9623

COLLEGE NAME : Amrita College Of Engineering And Technology

DEPARTMENT : Computer Science and Engineering

STUDENT NM-ID : B481E74927508BB8B0E4489704478AEC

ROLL NO : 962323104117

DATE : 10-09-2025

Completed the project named as Phase_2_ TECHNOLOGY

PROJECT NAME : Login Authentication System

SUBMITTED BY,

NAME : Varshini P

MOBILE NO : 86675 93417

Phase 2 - Solution Design And Architecture

Tech Stack Selection

1. Frontend

The frontend is responsible for providing an intuitive and responsive user interface for users to interact with the system.

- Technology: React.js
- Reason: React.js is a modern JavaScript library that allows for building dynamic and responsive web applications. Its component-based architecture ensures modularity and reusability of code. React also integrates well with authentication flows and supports real-time validation of user input.
- Styling: CSS3 / Bootstrap / Tailwind CSS
- Reason: These tools enable responsive and visually appealing UI design, improving user experience on both desktop and mobile devices.

2. Backend

The backend handles core logic such as user authentication, session management, and database operations.

- Technology: Node.js with Express.js
- Reason: Node.js offers asynchronous, event-driven architecture, making it ideal for handling multiple concurrent login requests efficiently. Express.js simplifies building RESTful APIs and routing, providing a lightweight framework for authentication endpoints.

UI structure/API schema design

UI Structure

The user interface is designed to provide a seamless and intuitive experience for authentication-related tasks. The UI consists of the following main screens/components:

1.Login Page

- Fields: Email/Username, Password
- Buttons: Login, Forgot Password, Sign Up
- Validation: Client-side validation for empty fields, email format, and password strength.

2.Registration Page

- Fields: Full Name, Email, Password, Confirm Password
- Buttons: Sign Up, Login
- Validation: Password strength, matching confirm password, and email uniqueness check.

3.Forgot Password / Reset Password

- Fields: Email (for OTP / reset link), New Password, Confirm Password
- Buttons: Submit, Cancel
- Validation: Email format, password strength, OTP validation.

4.Dashboard / Home Page

- Displays: User information after successful login.
- Buttons: Logout, Edit Profile

5.Notifications / Alerts

- Success messages for login/signup
- Error messages for invalid credentials or server errors

Data Handling Approach

1. User Data Collection

- **Types of Data**: Full name, email/username, password, login history, session tokens.
- **Validation:** Client-side and server-side validation to ensure correct formats (e.g., email validation, password strength).
- **Purpose Limitation:** Only essential data required for authentication and profile management is collected.

2. Data Storagek

Database: MongoDB (NoSQL) or PostgreSQL (SQL)

1.User Table / Collection Fields:

- userId (Primary Key)
- name
- email (Unique)
- passwordHash (hashed using bcrypt)
- createdAt, updatedAt
- loginHistory (optional array of login timestamps and IPs)

Password Security: Passwords are never stored in plain text. They are hashed with bcrypt before saving in the database.

3. Data Transmission

Secure Channels: All data transmitted between client and server uses HTTPS with SSL/TLS encryption.

Sensitive Information: Passwords, tokens, and personal information are encrypted or hashed.

Component or module diagram

The login authentication system is organized into modular components for better maintainability, scalability, and security. Each module has a specific responsibility.

1. Modules Overview

- **1.User Interface (Frontend)**
- Login Module: Handles login form and validation.
- Registration Module: Handles new user signup.
- Forgot Password / Reset Module: Handles password recovery.
- Dashboard Module: Displays user-specific content after authentication.

2.Authentication & Security (Backend)

- Auth Controller: Handles login, logout, registration, and token generation.
- Password Management Module: Handles password hashing, reset, and validation.
- JWT Token Module: Generates and validates authentication tokens.
- Security Module: Implements rate limiting, input sanitization, and vulnerability protection.

3.Database (Data Layer)

- User Table/Collection Module: Stores user credentials and profile info.
- Session / Token Module: Stores active session tokens (optional for stateful sessions).
- Audit / Logging Module: Records login attempts and security events.

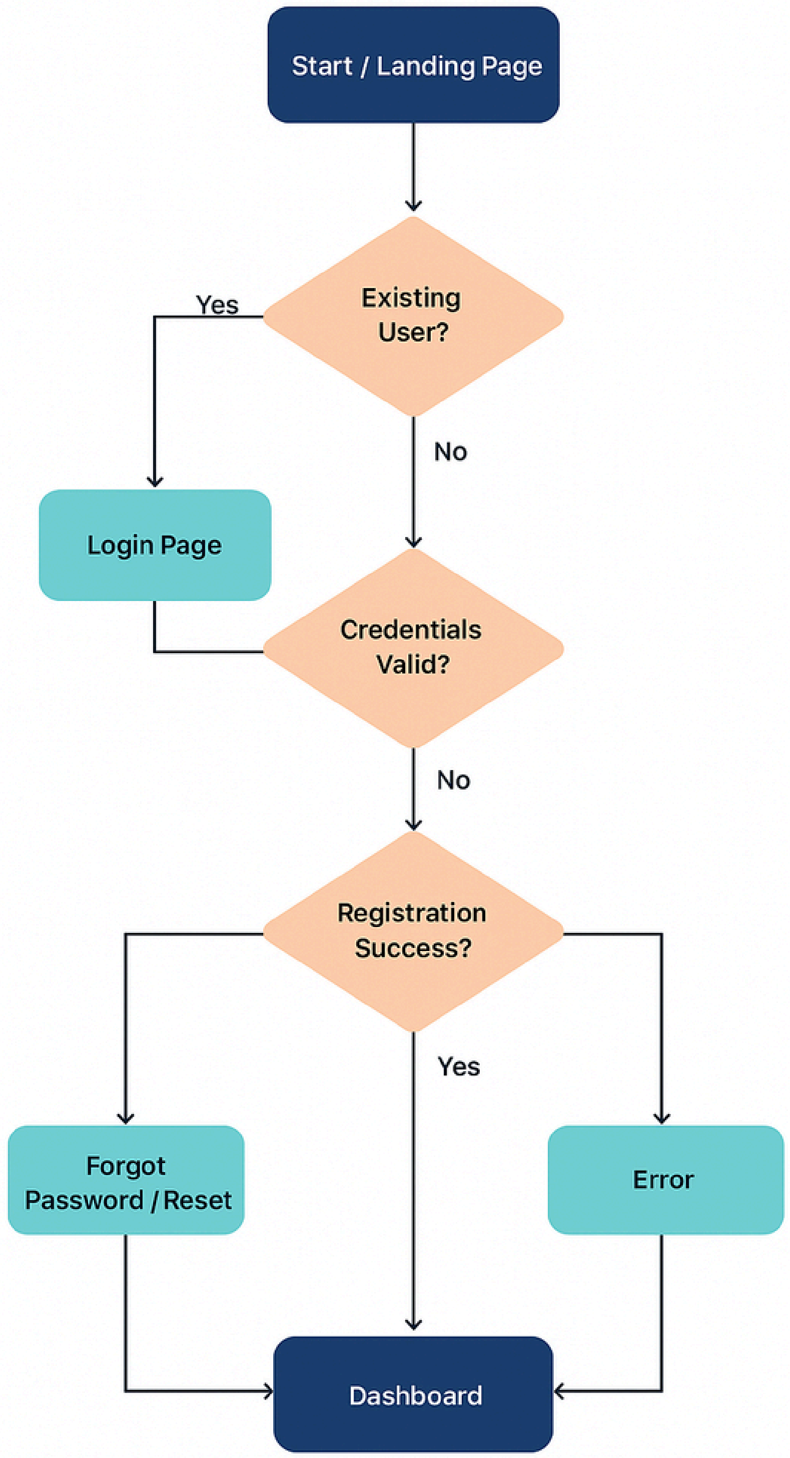
4.API Layer

- Auth APIs: Exposes endpoints for login, signup, logout, and password reset.
- User APIs: Exposes endpoints for profile view and update.

2. Component Diagram

```
+-------------------+
|     Frontend      |
|-------------------|
| Login Module      |
| Registration      |
| Forgot/Reset      |
| Dashboard         |
+---------+---------+
          |
          v
+-------------------+
|      Backend      |
|-------------------|
| Auth Controller   |
| JWT Token Module  |
| Password Module   |
| Security Module   |
+---------+---------+
          |
          v
+-------------------+
|     Database      |
|-------------------|
| User Collection   |
| Session/Token     |
| Audit/Logging     |
+-------------------+
```

Basic flow diagram


Start / Landing Page
Existing User?
Yes
No
Login Page
Credentials Valid?
No
Registration Success?
Yes
Forgot Password / Reset
Error
Dashboard