# Project Design Phase Solution Architecture

| Date | 27th February 2025 |
|------|--------------------|
| Team ID | SWUID20250152937 |
| Project Name | DocSync |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

**Example - Solution Architecture Diagram:**
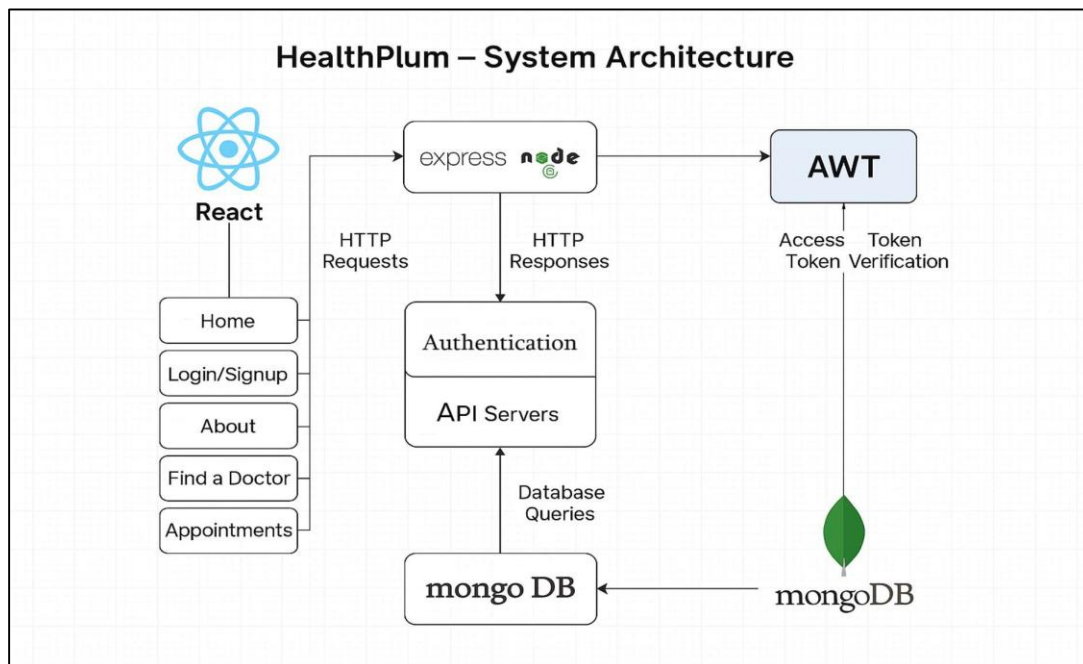
**Architecture Breakdown:**

1. **Client Side (React.js)**
   - The user (doctor or patient) interacts via the browser.
   - Pages like Home, About, Find a Doctor, Appointment Booking, Dashboards.
   - Uses Axios/Fetch to communicate with the backend via RESTful APIs.
   - Stores JWT tokens.

1. **API Gateway (Express + Node.js)**
   - **Routes:**
     - /api/authentication – login, signup for both roles.
     - /api/doctors – list, filter, profile view.
     - /api/appointments – create, update, cancel, fetch.
     - /api/patients – patient data, dashboard info.
   - **Middleware:**
     - Auth middleware verifies JWTs before granting access.
     - Role-based access control (Doctor/Patient).
     - Input validation & error handling.

1. **MongoDB Database**

- Collections:
  - **Users** (doctors & patients, roles, hashed passwords)
  - **Doctors** (name, department, schedule, ratings)
  - **Appointments** (time, doctorId, patientId, status)
- Relations via ObjectIds – efficient querying and filtering.

1. **Authentication Layer (JWT)**
   - On login/signup, JWT is issued and returned to client.
   - Frontend stores token and includes it in the Authorization header.
   - Backend verifies token for protected routes.