

Ikigai



by:

Mavra Mehak 22I-0798 Scrum Master

Maryum Tanvir 22I-0751 Developer/ Analyst

Salman Khan 22I-1285 QA/Design

**COMING
SOON**

Table Of Contents

1. Introduction.....	5
1.1. Problem Statement.....	5
2. Functional Requirements.....	6
2.1. Admin Management.....	6
2.2. Doctor Management.....	6
2.3. Patient Management.....	6
3. Non-Functional Requirements.....	7
3.1. Performance Requirements.....	7
3.2. Safety Requirements.....	7
3.3. Security Requirements.....	7
3.4. Software Quality Attributes.....	7
3.5. Business Rules.....	7
4. User Stories.....	8
5. Product Backlog.....	8
6. Sprint 1 Backlog.....	10
7. Sprint 2 Backlog.....	11
8. Sprint 3 Design.....	12
8.1. Sprint 3 Items.....	14
8.2. Design Details.....	15
8.2.1. Assign Shifts to Doctors:.....	15
8.2.2. Schedule an Appointment:.....	15
8.2.3. Cancel an Appointment:.....	15
8.2.4. Give Feedback on Appointment:.....	15
8.2.5. View Feedback Related to Appointments:.....	15
8.2.6. Reschedule an Appointment:.....	15
8.2.7. Update Appointment Completion Status and Availability:.....	16
9. System Architecture and Project Plan.....	16
10. Actual Implementation.....	17
10.1. Common Pages.....	17
10.1.1 Login Page.....	17
10.1.2 Sign Up Page.....	17
10.2. Admin's Pages.....	18
10.2.1 Admin's Dashboard.....	18
10.2.2 Admin Manage Doctors Page.....	18
10.2.3 Admin Manage Doctors Request.....	19
10.2.4 Admin Assign Shift Page.....	19
10.2.5 Admin Settings Page.....	20
10.3. Patient's Pages.....	20
10.3.1. Patient Dashboard.....	20

10.3.2. Patient Appointments Page.....	21
10.3.3. Patient Doctors/Book Appointments.....	21
10.3.4. Patient Payments.....	22
10.3.5. Patient Feedback.....	22
10.3.6. Patient Profile.....	23
10.4. Doctor's Pages.....	23
10.4.1. Doctor Dashboard.....	23
10.4.2. Doctor Appointments Page.....	24
10.4.3. Doctor Shifts Page.....	24
10.4.4. Doctor Feedback Page.....	25
10.4.5. Doctor Profile Page.....	25
11. Sprint 3 Burndown Chart.....	26
12. Trello Board.....	27
12.1. Sprint 1 Start.....	27
12.2. Sprint 2 End.....	27
12.3. Sprint 3 End.....	28
13. Black Box Testing.....	30
14. White Box Testing.....	31
15. Work Division.....	31
15.1. Mavra Mehak Gill - 22i-0798.....	31
15.1.1. Contribution Area.....	31
15.1.2. Details.....	31
15.2. Maryum Tanvir - 22i-0751.....	32
15.2.1. Contribution Area.....	32
15.2.2. Details.....	32
15.3. Salman Khan - 22i-1285.....	32
15.3.1. Contribution Area.....	32
15.3.2. Details.....	32
16. Retrospective.....	33
16.1. What went well.....	33
16.1.1. Modular Architecture:.....	33
16.1.2. Robust Backend Implementation:.....	33
16.1.3. Frontend Useability:.....	33
16.2. What could be Improved.....	33
16.2.1. Testing Coverage.....	33
16.2.2. Performance Optimization.....	33
16.3. Actions.....	34
16.3.1. Expand Testing.....	34
16.3.2. Optimize Performance.....	34
17. Review.....	34
17.1. System Quality.....	34

17.2. Alignment with Requirements.....	34
17.3. Strengths.....	34
17.4. Future Optimization Recommendations.....	35
18. Lesson Learnt.....	35
19. Links.....	35
19.1. Github Version Control.....	35
19.2. Vercel Live Demo.....	36
19.3. Presentation.....	36

1. Introduction

The Ikigai/I-CARE Clinic Appointment Management System is a full-stack web application developed to streamline hospital operations by centralizing functions, reducing manual errors, and improving communication. Designed with the needs of admins, doctors, and patients in mind, the system provides a seamless platform for appointment scheduling, doctor management, patient registration, shift assignments, and feedback collection.

Built using the MERN stack — MongoDB, Express.js, React.js, and Node.js, the application follows the MVC architecture to ensure a clean separation of concerns and scalable development practices. The frontend offers an intuitive and responsive interface for users, while the backend provides robust APIs to handle business logic and database operations.

1.1. Problem Statement

The problem of	Clinics generate significant amounts of operational data but often rely on manual processes for managing appointments, doctor schedules, patient registration, and feedback. This leads to scheduling errors, patient dissatisfaction, inefficient use of staff time, and revenue losses.
affects	Doctors, Administrators, and Patients.
the impact of which is	Reduced quality of patient care, delayed treatments, increased costs, scheduling conflicts, and inefficient inventory management.
a successful solution would be	I-CARE is a comprehensive, centralized Clinic Appointment Management System designed to digitize and streamline all clinic operations. It integrates patient registration, appointment scheduling, doctor management, shift assignment, payment processing, and feedback collection into a single easy-to-use platform. By providing real-time access to available doctors, upcoming appointments, and pending actions, I-CARE ensures

	that clinics run efficiently, patients are cared for promptly, and doctors' time is utilized optimally.
--	---

I-CARE addresses common challenges faced in traditional hospital setups, such as inefficient scheduling, lack of transparency in doctor availability, and delayed feedback loops. The system incorporates features like user authentication, doctor profile management, shift scheduling, appointment tracking, payment integration, and feedback collection.

The project was developed using an agile methodology in a team of three students, progressing through multiple sprints. Each sprint focused on delivering key features iteratively, with continuous integration and testing to ensure quality and usability. I-CARE not only digitizes hospital management but also demonstrates the team's understanding of modern web development practices and software engineering principles.

2. Functional Requirements

2.1. Admin Management

- Admins should be able to update or remove doctor profiles.
- Admins should be able to approve doctors with verified credentials.
- Admins should be able to assign shifts to doctors.

2.2. Doctor Management

- Doctors should be able to filter, view & update past, present and future appointments
- Doctors should be able to provide medical notes & update appointment completion status.
- Doctors should be able to update their availability.
- Doctors should be able to filter & view their assigned shifts.
- Doctors should be able to update & view appointment feedback.

2.3. Patient Management

- Patients should be able to register in the system.
- Patients should be able to filter & view available doctors.
- Patients should be able to schedule, cancel, and reschedule appointments.
- Patients should be able to provide feedback.

- Patients should be able to pay & view consultation fees.
- Patients should be able to track appointment history.

3. Non-Functional Requirements

3.1. Performance Requirements

- The system shall respond to 90% of user requests within 2 seconds under normal load conditions.

3.2. Safety Requirements

- The system shall ensure that user data (e.g., patient records, doctor credentials) is stored securely to prevent unauthorized access

3.3. Security Requirements

- All patient data should be encrypted using JWT authentication and bcrypt encryption.
- The system shall implement user roles and permissions to restrict access to sensitive data and functions (e.g., only admins can manage doctor profiles)

3.4. Software Quality Attributes

- The system shall have a user-friendly interface that is easy to navigate and understand for admins, doctors, and patients.
- The system shall provide clear and helpful error messages to guide users in case of problems.
- The system shall be designed in a modular way to facilitate future maintenance and updates.
- The web application should have a responsive design, supporting desktop, tablet, and mobile devices

3.5. Business Rules

- The system shall enforce that only verified doctors can be granted access to the system.
- The system shall ensure that patients can only book appointments with available doctors and time slots

4. User Stories

(Mentioned in Story Card Format)

Story ID: 1

Story Title: Update or Remove Doctor Profiles

Importance:

Medium

User Story:

As an admin, I want to update or remove doctor profiles so that the system always has accurate records

Estimate:

3

Acceptance Criteria:

Given an admin is logged into the system,
When they navigate to the doctor management section and select a doctor profile,
Then they should be able to update or remove the doctor's details successfully.

• Pre-conditions:

- The admin is logged into the system.
- The system has an existing doctor management section.
- The admin has navigated to the doctor management section.
- The admin has selected a doctor profile to update or remove.

• Post-conditions:

- The doctor's details are successfully updated or removed.
- A confirmation message is displayed to the admin.

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report/View

Story ID: 2

Story Title: Approve Doctors with Verified Credentials

User Story:

As an admin, I want to approve doctors with verified credentials so that only verified doctors can be a part of the system.

Importance:

High

Estimate:

5

Acceptance Criteria:

Given a doctor has submitted their credentials for approval,
When an admin reviews and verifies the credentials,
Then the admin should be able to approve the doctor and grant system access.

- Pre-conditions:
 - The doctor has submitted their credentials for approval.
 - The admin is logged into the system.
 - The system provides a doctor approval section where pending applications are listed.
- Post-conditions:
 - The doctor gains access to the system.
 - The doctor is now visible in relevant sections where approved doctors are listed.

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report/View

Story ID: 3

Story Title: View Assigned Shifts As Doctor

User Story:

As a doctor, I want to view my assigned shifts so that I can manage my time effectively.

Importance:

Medium

Estimate:

2

Acceptance Criteria:

Given a doctor is logged into the system,
When they navigate to the shift schedule page,
Then they should see their assigned shifts with date and time details.

- Pre-conditions:
 - The doctor is logged into the system.
 - The doctor navigates to the shift schedule page.
- Post-conditions:
 - The doctor can view their assigned shifts with date and time details.
 - The shift information is accurate and up to date.

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report/View

Story ID: 4

Story Title: Assign Shifts to Doctors

Importance:

High

User Story:

As an admin, I want to assign shifts to doctors so that patient care is properly scheduled.

Estimate:

5

Acceptance Criteria:

Given an admin is logged into the system,
When they access the shift scheduling module,
Then they should be able to assign shifts to doctors based on availability.

- Pre-conditions:
 - The admin is logged into the system.
 - The system has a shift scheduling module.
 - There are doctors registered in the system.
 - The system has access to doctors' availability information.
 - The admin navigates to the shift scheduling module.
- Post-conditions:
 - The selected doctor is successfully assigned a shift.
 - The shift details (date, time, and location) are saved in the system.
 - The assigned shift is reflected in the doctor's schedule.

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report/View

Story ID: 5

Story Title: Search Available Doctors As Patient

Importance:
Medium

User Story:

As a patient, I want to search available doctors so that I can choose the best one for my needs.

Estimate:
3

Acceptance Criteria:

Given a patient is logged into the system,
When they search for doctors by specialty or availability,
Then they should see a list of available doctors with relevant details.

- Pre-conditions:
 - The patient is logged into the system.
 - Doctors have their specialties and availability recorded in the system.
 - The patient navigates to the doctor search page.
- Post-conditions:
 - The patient sees a list of available doctors matching their search criteria (e.g., specialty, availability).
 - Each doctor's profile displays relevant details such as name, specialty, availability, and contact options.
 - The patient can make an informed choice based on the provided information.
 - The system ensures that only up to date availability is shown

Type:
Search
Workflow
Manage Data
Payment
Report/View

Story ID: 6

Story Title: Register in the System

Importance:

High

User Story:

As a patient, I want to register in the system so that I can receive medical services.

Estimate:

3

Acceptance Criteria:

Given a new user wants to access hospital services,
When they fill out the registration form with valid details,
Then they should be successfully registered and receive login credentials.

- Pre-conditions:
 - The user is not already registered in the system.
 - The system has a patient registration module.
 - The user has navigated to the registration page.
 - The user provides all required valid details.
- Post-conditions:
 - The patient is successfully registered in the system.
 - The patient can now log in and access medical services.

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report/View

Story ID: 7

Story Title: Schedule an Appointment

Importance:

High

User Story:

As a patient, I want to schedule an appointment so that I can consult a doctor at a convenient time.

Estimate:

5

Acceptance Criteria:

Given a patient is logged into the system,
When they select a doctor and available time slot,
Then the appointment should be successfully scheduled and confirmed.

- Pre-conditions:
 - The patient is logged into the system.
 - The system has a list of registered doctors with their available time slots.
 - The patient has navigated to the appointment scheduling module.
 - The selected doctor has available time slots.
- Post-conditions:
 - The appointment is successfully scheduled and saved in the system.
 - The doctor's schedule is updated to reflect the new appointment.
 - The patient can view the appointment details.

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report/View

Story ID: 8

Story Title: Cancel an Appointment

User Story:

As a patient, I want to cancel an appointment so that I can free up my slot for someone else.

Importance:

Medium

Estimate:

2

Acceptance Criteria:

Given a patient has a scheduled appointment,
When they choose to cancel the appointment before the appointment time,
Then the appointment should be removed from the system, and the slot should be available for others.

• Pre-conditions:

- The patient is logged into the system.
- The patient has at least one scheduled appointment.
- The system allows appointment cancellations before the appointment time.
- The patient navigates to the appointment management section.

• Post-conditions:

- The appointment is successfully removed from the system.
- The doctor's availability is updated, making the slot open for other patients.

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report/View

Story ID: 9

Story Title: Filter & View Upcoming Appointments As Doctor

User Story:

As a doctor, I want to view my upcoming appointments so that I can prepare in advance.

Importance:

Medium

Estimate:

2

Acceptance Criteria:

Given a doctor is logged into the system,
When they access their appointment schedule,
Then they should see a list of all upcoming appointments with patient details.

- Pre-conditions:
 - The doctor is logged into the system.
 - The doctor has at least one upcoming appointment.
 - Patient details are stored in the system.
 - The doctor navigates to the appointment schedule section.
- Post-conditions:
 - The doctor sees a list of all upcoming appointments with relevant details.
 - The information is accurate and up to date.

Type:

- Search
- Workflow
- Manage Data
- Payment
- Report/View

Story ID:
10

Story Title: View Past and Upcoming Appointments As Patient

Importance:
Medium

User Story:

As a patient, I want to view my past and upcoming appointments so that I can keep track of my medical history.

Estimate:
3

Acceptance Criteria:

Given a patient is logged into the system,
When they navigate to the appointment history section,
Then they should see a list of past and future appointments.

- Pre-conditions:
 - The patient is logged into the system.
 - The patient navigates to the appointment section.
- Post-conditions:
 - The patient sees a list of all past and upcoming appointments.
 - The information is accurate and up to date.

Type:

Search
Workflow
Manage Data
Payment
Report/View

Story ID:
11

Story Title: Submit Feedback on Appointment

Importance:
Low

User Story:

As a patient, I want to give feedback on my appointment so that I can share my experience with the doctor.

Estimate:
2

Acceptance Criteria:

Given a patient has completed an appointment,
When they submit feedback in the system,
Then the feedback should be recorded and accessible by relevant staff.

- Pre-conditions:
 - The patient is logged into the system.
 - The patient has completed at least one appointment.
 - The patient navigates to the feedback section.
- Post-conditions:
 - The feedback is successfully recorded in the system.
 - The patient may receive a confirmation that their feedback was submitted.

Type:
Search
Workflow
Manage Data
Payment
Report/View

Story ID:
12

Story Title: Generate prescription for the patient

Importance:
High

User Story:

As a doctor, I can view the medical history of the patient and then give them a prescription for their treatment.

Estimate:
2

Acceptance Criteria:

Given a doctor is logged into the system,
When they view patient's medical information
Then they should prescribe a treatment for them

- Pre-conditions:
 - The doctor is logged into the system.
 - The doctor navigates to the appointments section.
- Post-conditions:
 - The doctor can view patient's medical history
 - and generate a prescription for the patient.

Type:
Search
Workflow
Manage Data
Payment
Report/View

Story ID:
13

Story Title: Pay Consultation Fee Online

Importance:
Medium

User Story:

As a patient, I want to pay my consultation fee online so that I can complete the payment process of my appointment.

Estimate:
2

Acceptance Criteria:

Given a patient has booked an appointment,
When they access the payment section,
Then they should be able to view the total bill.

- Pre-conditions:
 - The patient is logged into the system.
 - The patient has a confirmed appointment.
 - The patient navigates to the payment section.
- Post-conditions:
 - The patient pays the fee, and a final bill is generated.

Type:

Search
Workflow
Manage Data
Payment
Report/View

Story ID:
14

Story Title: Reschedule an Appointment

Importance:

Low

User Story:

As a patient, I want to reschedule an appointment so that I can choose a new time if my availability changes.

Estimate:

3

Acceptance Criteria:

Given a patient has a scheduled appointment,
When they choose to reschedule before the appointment time,
Then they should be able to select a new available time slot, and the system should update the appointment details.

- Pre-conditions:
 - The patient is logged into the system.
 - The patient has a confirmed appointment.
 - The system allows rescheduling before the appointment time.
 - The system has available time slots for rescheduling.
 - The patient navigates to the appointment management section.
- Post-conditions:
 - The appointment is successfully updated with the new time slot.
 - The previous appointment time slot becomes available for other patients.
 - The patient receives a confirmation notification about the updated appointment.
 - The doctor's schedule is updated with the new appointment time.

Type:

- Search
 Workflow
 Manage Data
 Payment
 Report/View

Story ID:
15

Story Title: Update Appointment Completion Status and Availability

Importance:
Medium

User Story:

As a doctor, I want to update my appointment completion status and adjust my availability so that patients can see when I am free for new appointments.

Estimate:
3

Acceptance Criteria:

Given a doctor has completed an appointment,
When they mark the appointment as completed in the system,
Then their availability status should be updated automatically based on their configured schedule and remaining appointment slots.

- Pre-conditions:
 - The doctor is logged into the system.
 - The doctor has at least one scheduled appointment.
 - The doctor navigates to the appointment management section.
- Post-conditions:
 - The appointment is marked as completed in the system.
 - The doctor's availability is updated based on their configured schedule.

Type:

Search
 Workflow
 Manage Data
 Payment
 Report/View

5. Product Backlog

Item ID	Type of Backlog Item	Description	Priority	Status	Sprint	Assigned	Acceptance Criteria
1	User Story	Update or remove doctor profiles	Medium	Completed	Sprint 1	Maryum Tanvir	Admin can update/remove doctor profiles and receive confirmation
2	User Story	Approve doctors with verified credentials	High	Completed	Sprint 2	Maryum Tanvir	Admin can review and approve doctors, granting system access
3	User Story	Filter, view assigned shifts as a doctor	Medium	Completed	Sprint 1	Salman Khan	Doctor sees past/current/future assigned shifts with date/time details
4	User Story	Assign shifts to doctors	High	Completed	Sprint 3	Maryum Tanvir	Admin assigns shifts, and system updates doctor's schedule
5	User Story	Search available doctors as a patient	Medium	Completed	Sprint 1	Mavra Mehak	Patient searches for doctors by specialty/availability
6	User Story	Register as a patient	High	Completed	Sprint 2	Mavra Mehak	Patient fills form, gets registered, and receives login credentials

7	User Story	Schedule an appointment	High	Completed	Sprint 3	Mavra Mehak	Patient selects doctor/time, appointment is confirmed
8	User Story	Cancel an appointment	Medium	Completed	Sprint 3	Mavra Mehak	Patient cancels appointment, slot becomes available
9	User Story	Filter, view upcoming appointments as a doctor	Medium	Completed	Sprint 1	Salman Khan	Doctor sees upcoming appointments with patient details
10	User Story	Search and view doctors for managing them	Medium	Completed	Sprint 1	Maryum Tanvir	Admin can search doctors they want to edit to remove
11	User Story	Submit feedback after an appointment	Low	Completed	Sprint 3	Mavra Mehak	Patient submits feedback, stored in system
12	User Story	Generate prescription for the patient	High	Completed	Sprint 3	Salman Khan	Doctor sees patient's medical information and submits a prescription
13	User Story	Pay consultation fee online	Medium	Completed	Sprint 2	Mavra Mehak	Patient sees bill in payment section
14	User Story	Reschedule an appointment	Low	Completed	Sprint 3	Mavra Mehak	Patient selects new time, system updates appointment

15	User Story	Update appointment completion status as a doctor	Medium	Completed	Sprint 3	Salman Khan	Doctor marks appointment completed, availability updates
16	UI	Login/ Register Pages	High	Completed	Sprint 2	Salman Khan	Any User can Register their Account or Login

Key:

High

Medium

Low

6. Sprint 1 Backlog

Item ID	Type of Backlog Item	Description	Priority	Status	Sprint	Assigned	Acceptance Criteria
1	User Story	Update or remove doctor profiles	Medium	Completed	Sprint 1	Maryum Tanvir	Admin can update/remove doctor profiles and receive confirmation
3	User Story	Filter, view assigned shifts as a doctor	Medium	Completed	Sprint 1	Salman Khan	Doctor sees past/current/future assigned shifts with date/time details

Key:

High

Medium

Low

7. Sprint 2 Backlog

Item ID	Type of Backlog Item	Description	Priority	Status	Sprint	Assigned	Acceptance Criteria

2	User Story	Approve doctors with verified credentials	High	Completed	Sprint 2	Maryum Tanvir	Admin can review and approve doctors, granting system access
6	User Story	Register as a patient	High	Completed	Sprint 2	Mavra Mehak	Patient fills form, gets registered, and receives login credentials
13	User Story	Pay consultation fee online	Medium	Completed	Sprint 2	Mavra Mehak	Patient sees bill in payment section
16	UI	Login/ Register Pages	High	Completed	Sprint 2	Salman Khan	Any User can Register their Account or Login

Key:

High

Medium

Low

8. Sprint 3 Design

Item ID	Type of Backlog Item	Description	Priority	Status	Sprint	Assigned	Acceptance Criteria
4	User Story	Assign shifts to doctors	High	Completed	Sprint 3	Maryum Tanvir	Admin assigns shifts, and system updates doctor's schedule

7	User Story	Schedule an appointment	High	Completed	Sprint 3	Mavra Mehak	Patient selects doctor/time, appointment is confirmed
8	User Story	Cancel an appointment	Medium	Completed	Sprint 3	Mavra Mehak	Patient cancels appointment, slot becomes available
11	User Story	Submit feedback after an appointment	Low	Completed	Sprint 3	Mavra Mehak	Patient submits feedback, stored in system
12	User Story	Generate prescription for the patient	High	Completed	Sprint 3	Salman Khan	Doctor sees patient's medical information and submits a prescription
14	User Story	Reschedule an appointment	Low	Completed	Sprint 3	Mavra Mehak	Patient selects new time, system updates appointment
15	User Story	Update appointment completion status as a doctor	Medium	Completed	Sprint 3	Salman Khan	Doctor marks appointment completed, availability updates

Key:

High

Medium

Low

8.1. Sprint 3 Items

The image displays two side-by-side screenshots of a digital task board, likely from a platform like Trello or Asana. Both screenshots show a list of completed items under the heading "Current Sprint - Done".

Screenshot 1 (Left):

- (4) As an admin, I want to assign shifts to doctors so that patient care is properly scheduled.
Due: Apr 17 - Apr 20
Progress: 100%
Last updated: Apr 20
Owner: IT
- (7) As a patient, I want to schedule an appointment so that I can consult a doctor at a convenient time.
Due: Apr 15 - Apr 17
Progress: 100%
Last updated: Apr 17
Owner: MM
- (8) As a patient, I want to cancel an appointment so that I can free up my slot for someone else.
Due: Apr 18 - Apr 20
Progress: 100%
Last updated: Apr 20
Owner: MM
- (11) As a patient, I want to give feedback on my appointment so that I can share my experience with the doctor.
Due: Apr 21 - Apr 22
Progress: 100%
Last updated: Apr 22
Owner: MM

Screenshot 2 (Right):

- (11) As a patient, I want to give feedback on my appointment so that I can share my experience with the doctor.
Due: Apr 21 - Apr 22
Progress: 100%
Last updated: Apr 22
Owner: MM
- (12) As a doctor, I want to review feedback related to my appointments so that I can improve patient care.
Due: Apr 16 - Apr 18
Progress: 100%
Last updated: Apr 18
Owner: SK
- (14) As a patient, I want to reschedule an appointment so that I can choose a new time if my availability changes.
Due: Apr 23 - Apr 25
Progress: 100%
Last updated: Apr 25
Owner: MM
- (15) As a doctor, I want to update my appointment completion status and adjust my availability so that patients can see when I am free for new appointments.

+ Add a card

8.2. Design Details

8.2.1. Assign Shifts to Doctors:

This feature enables admins to schedule shifts for doctors by accessing a shift scheduling module, ensuring proper patient care coverage through an intuitive interface that displays doctor availability and shift details.

8.2.2. Schedule an Appointment:

This functionality allows patients to book appointments by selecting a doctor and an available time slot, with the system confirming the appointment and updating the doctor's schedule in real time.

8.2.3. Cancel an Appointment:

This feature permits patients to cancel their scheduled appointments, freeing up the slot for others while automatically updating the doctor's availability to reflect the change.

8.2.4. Give Feedback on Appointment:

This feature allows patients to submit feedback after completing an appointment, which is then recorded for doctor review.

8.2.5. View Feedback Related to Appointments:

This functionality enables doctors to access patient feedback on their appointments, displaying comments and ratings to help improve patient care.

8.2.6. Reschedule an Appointment:

This feature allows patients to modify their appointment time by selecting a new available slot, ensuring the system updates the appointment details and doctor's schedule accordingly.

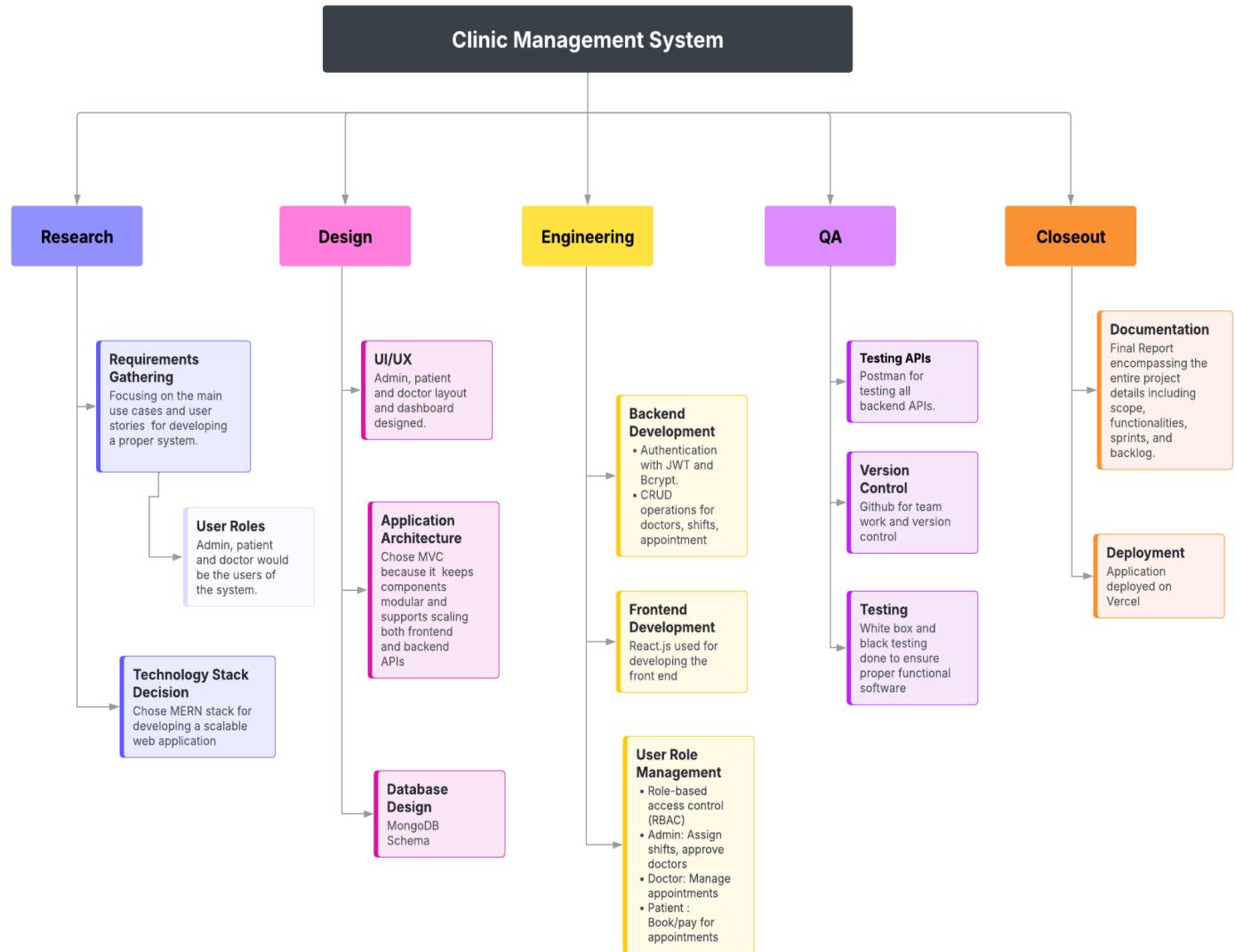
8.2.7. Update Appointment Completion Status and Availability:

This feature enables doctors to mark appointments as completed, automatically adjusting their availability based on their configured schedule to reflect open slots for new appointments.

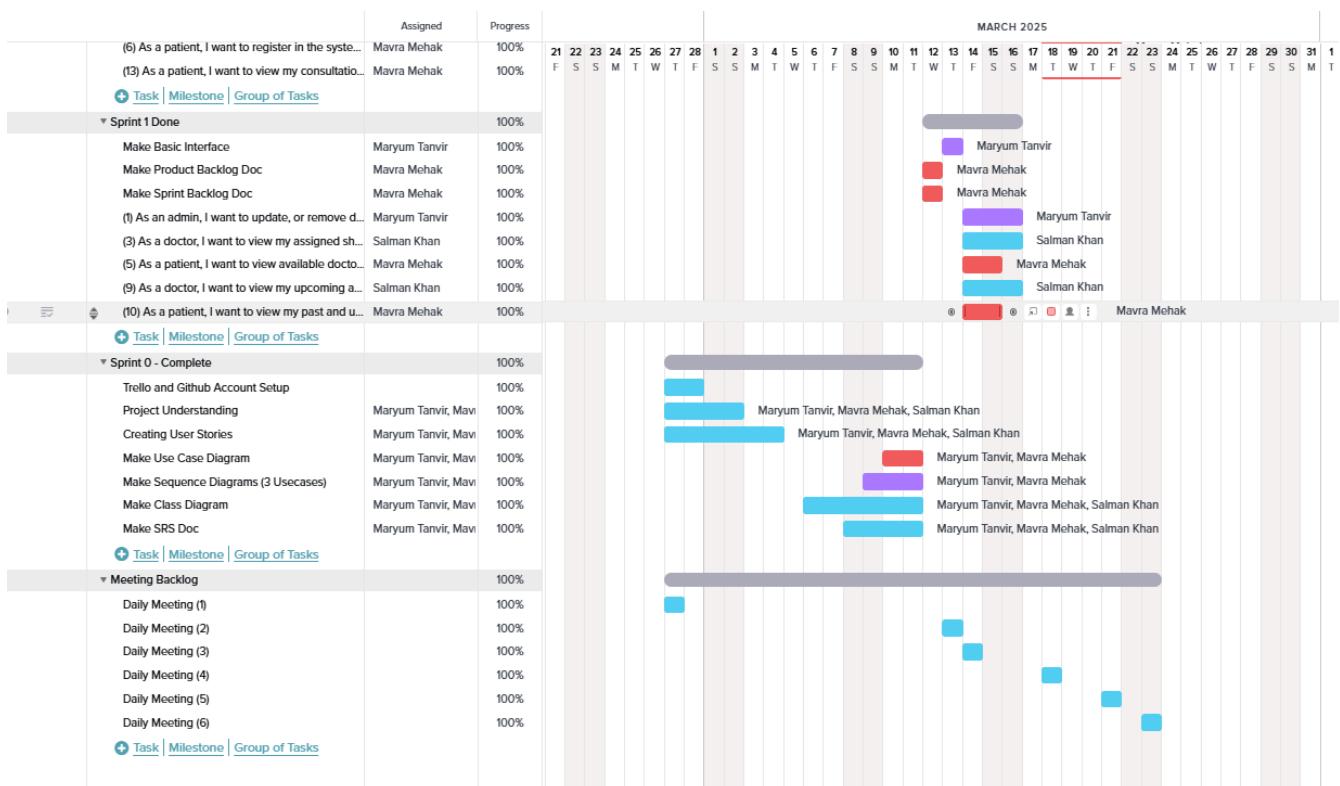
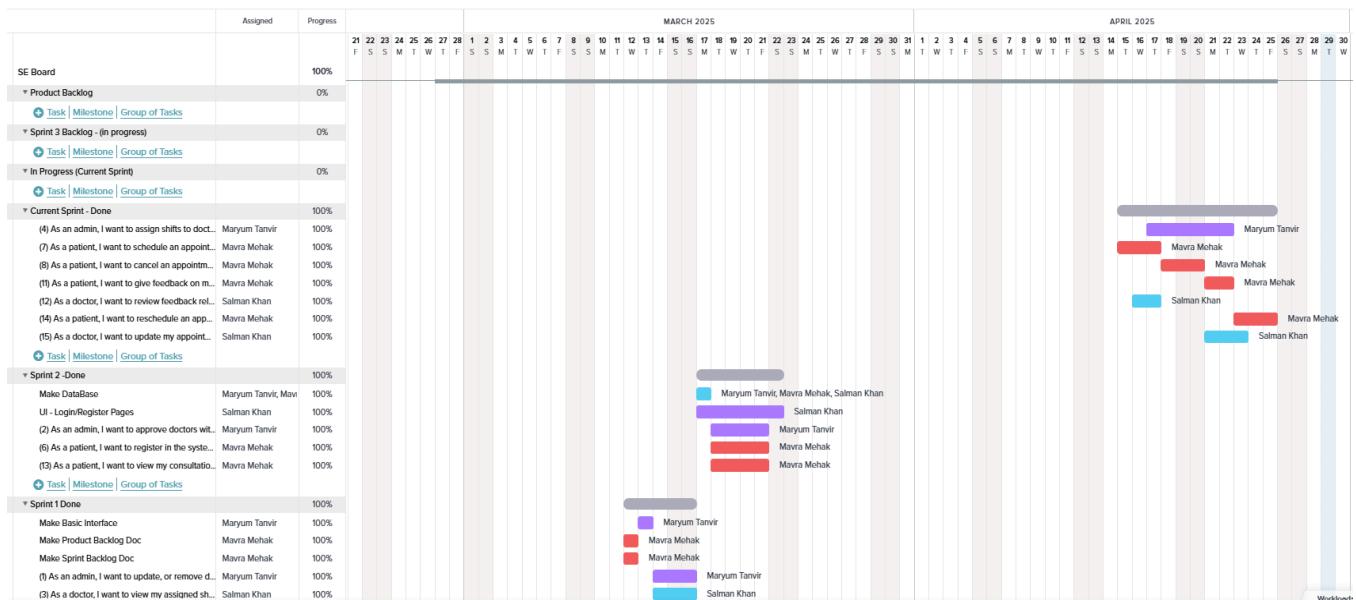
9. System Architecture and Project Plan

(Reference to System Architecture & Project Plan reports)

1) Work Breakdown Structure



2) Gantt Chart



Architecture Style

The Ikigai/I-CARE Clinic Appointment Management System is built using the **MERN** stack (MongoDB, Express.js, React.js, Node.js) and follows the **Model-View-Controller** (MVC) architectural style.

- **Model:** The data layer is handled using MongoDB, where all hospital management-related data (e.g., doctor profiles, patient records, appointments) are stored. The models define the structure of the database collections and are used to interact with the database.
- **View:** The frontend is developed using React.js, which acts as the View in the MVC pattern. React is responsible for presenting data to users and providing a dynamic, responsive user interface. It communicates with the backend via APIs.
- **Controller:** The controllers are implemented in Express.js within the Node.js backend. Controllers handle the business logic, processing incoming HTTP requests, interacting with the models, and sending appropriate responses back to the client.

By using MVC, I-CARE separates concerns clearly, making the system more organized, easier to maintain, scalable, and testable. Each part (Model, View, Controller) is independently developed and maintained, improving collaboration and future extension of the application.

1) Deployment Diagram:

Nodes and Artifacts

Client Device

- **Description:** End user's device (desktop, tablet, mobile).
- **Artifact: Browser (View - MVC)**
 - Runs React frontend
 - Built with Vite (vite for dev, vite build for prod).
 - Features:

- **Structure:** public/index.html (entry), src (pages: Dashboard.jsx, Appointments.jsx, Signup.jsx; components with antd, lucide-react icons; context: AuthContext.jsx, ThemeContext.jsx).
 - **Styling:** Tailwind CSS
 - **Data:** axios for API calls
 - **Routing:** react-router-dom for navigation.
 - **Date Handling:** dayjs for dates in Appointments.jsx, Shifts.jsx.
- Communicates with Application Server via HTTP/HTTPS (enabled by cors).

Application Server

- **Description:** Cloud-based or on-premises server hosting Node.js/Express.js backend.
- **Artifact: Node.js Runtime (Controller - MVC)**
 - Runs backend.
 - Features:
 - **Entry:** server.js (express, cors, morgan for logging).
 - **Routes:** API endpoints (admin.routes.js, appointment.routes.js, auth.routes.js).
 - **Controllers:** Business logic (admin_controller.js, appointment_controller.js).
 - **Middlewares:** auth_middleware.js (jsonwebtoken, bcryptjs), error_middleware.js, multer_middleware.js (multer, path, uuid).
 - **Config:** dotenv for .env (e.g., DB URI, JWT secret).
 - Connects to Database Server via mongoose.

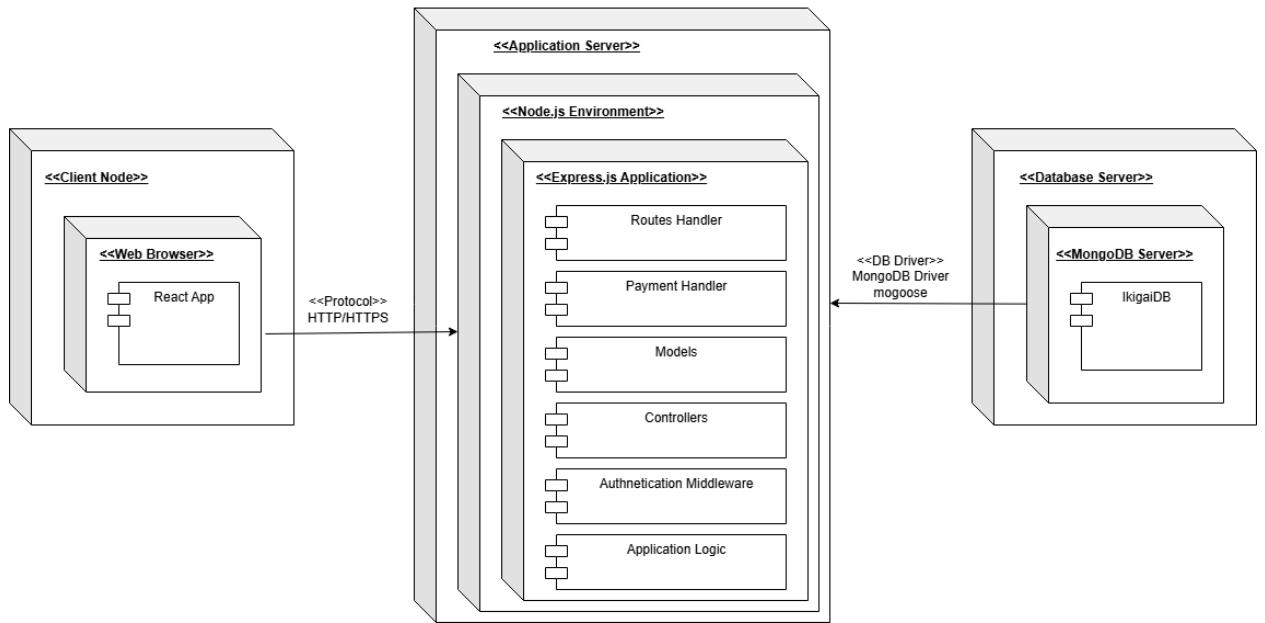
Database Server

- **Description:** Server hosting MongoDB.
- **Artifact: MongoDB Server (Model - MVC)**
 - Stores I-CARE database with collections (AdminProfile.js, Appointment.js, DoctorRequest.js, Feedback.js, PatientProfile.js, Payment.js, Shift.js, User.js).
 - Accessed by Application Server via mongoose for CRUD operations.

Connections

- **Client Device (Browser) ↔ Application Server:** HTTP/HTTPS requests (enabled by cors).
- **Application Server ↔ Database Server:** MongoDB protocol via mongoose.

Deployment Diagram



2) Package Diagram

I-CARE (Top-Level Package)

The main package encompassing the entire application.

Frontend (Package)

Encapsulates client-side code, built with React.js and Vite.

- **Sub-Packages:**
 - `src`: Core application code.

- **pages**: React components for views (e.g., Dashboard.jsx, Appointments.jsx, Signup.jsx) for admin, auth, doctor, patient.
 - **components**: Reusable UI components (e.g., generic/AppointmentCard.jsx), using antd for UI and lucide-react for icons.
 - **context**: Global state management (e.g., AuthContext.jsx, ThemeContext.jsx).
 - **assets**: Static assets (e.g., images/logo.png, styles/global.css).
 - **hooks**: Custom hooks (e.g., useAuth.js) for authentication logic.
- **Configuration:**
 - vite.config.js
 - tailwind.config.js
 - eslint.config.js
- **Dependencies:**
 - react-router-dom for routing.
 - axios for API calls.
 - dayjs for date handling.
- **Dependency Arrow:** src → backend for API access.

Backend (Package)

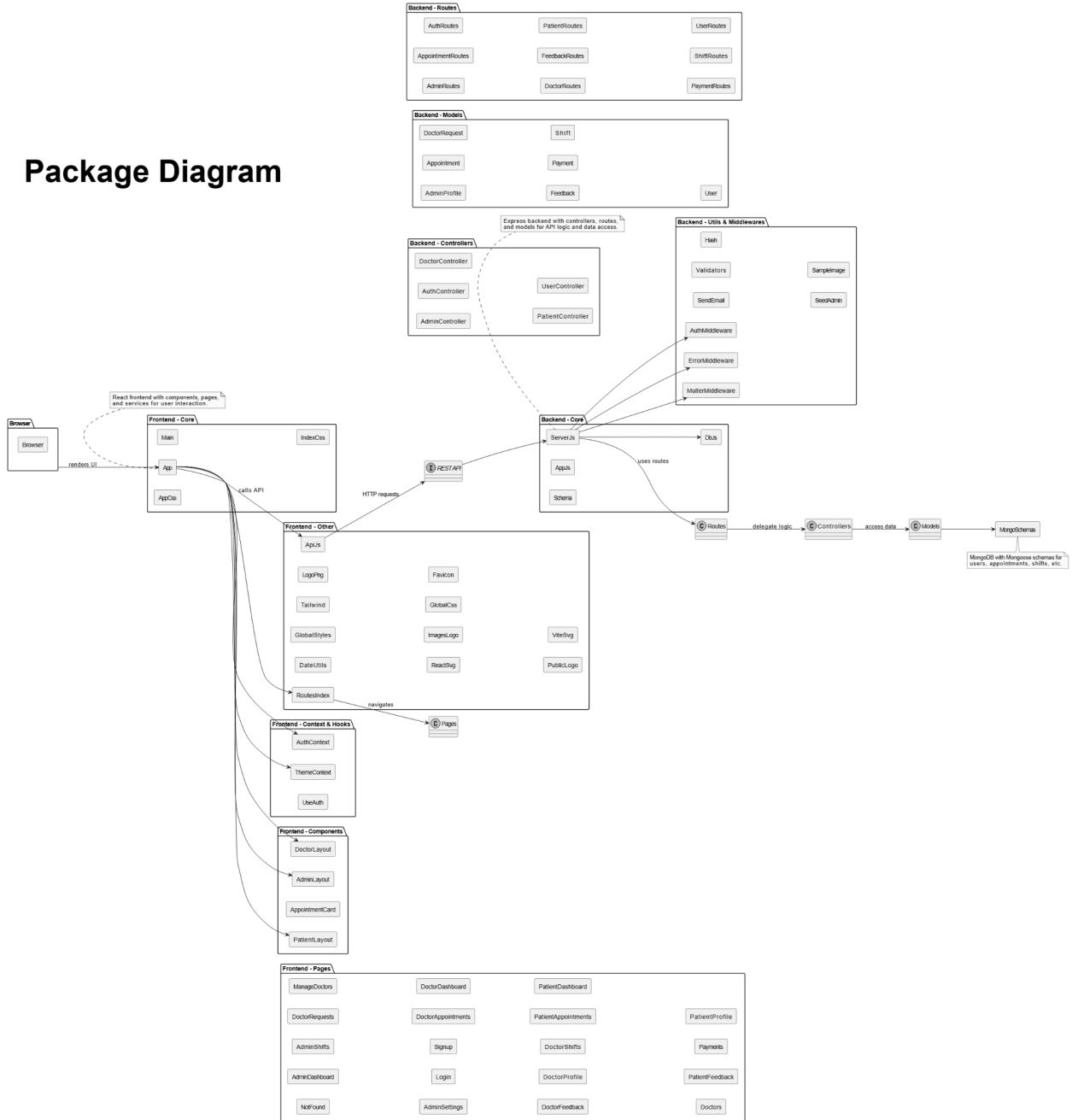
Encapsulates server-side code built with Node.js and Express.js.

- **Sub-Packages:**
 - **routes**: API endpoints (e.g., admin.routes.js, appointment.routes.js), using express.
 - **controllers**: Business logic (e.g., admin_controller.js, appointment_controller.js).
 - **models**: MongoDB schemas (e.g., AdminProfile.js, Appointment.js), using mongoose.
 - **middlewares**:
 - auth_middleware.js (with jsonwebtoken, bcryptjs).
 - error_middleware.js
- **Configuration:**
 - dotenv for environment variables (.env).
- **Dependencies:**
 - cors for cross-origin requests.
 - nodemon for development.

- **Dependency Arrows:**

- routes → models for database access.
- controllers → middlewares for request processing.

Package Diagram



3) Component Diagram

Frontend Component (React.js)

- **Location:** frontend/src
- **Sub-Components:**
 - **Admin:** Dashboard.jsx, DoctorRequests.jsx, ManageDoctors.jsx, Settings.jsx (using antd for UI, lucide-react for icons).
 - **Doctor:** DoctorDashboard.jsx, Appointments.jsx, Feedback.jsx, Profile.jsx, Shifts.jsx (using dayjs for date handling).
 - **Patient:** Dashboard.jsx, Doctors.jsx, Appointments.jsx, Feedback.jsx, Payments.jsx, Profile.jsx.
 - **Auth:** Signup.jsx, Login.jsx (with AuthContext.jsx, useAuth.js).
- **Routing & Data:**
 - react-router-dom for navigation (index.jsx).
 - axios for API requests.
- **Styling:** Tailwind CSS (styles/global.css) for responsive design.
- **Interfaces:** HTTP interface for API calls to Backend Component (e.g., GET /doctors, POST /appointments) via axios.

Backend Component (Node.js/Express.js)

- **Location:** backend
- **Sub-Components:**
 - **Controller:** admin_controller.js, appointment_controller.js, auth_controller.js (business logic with express).
 - **Middleware:**
 - auth_middleware.js (jsonwebtoken for JWT, bcryptjs for password hashing).
 - error_middleware.js.
 - multer_middleware.js (multer for file uploads, path, uuid).
 - **Model:** Mongoose schemas (AdminProfile.js, Appointment.js, etc.) for MongoDB via mongoose.
- **Setup:**

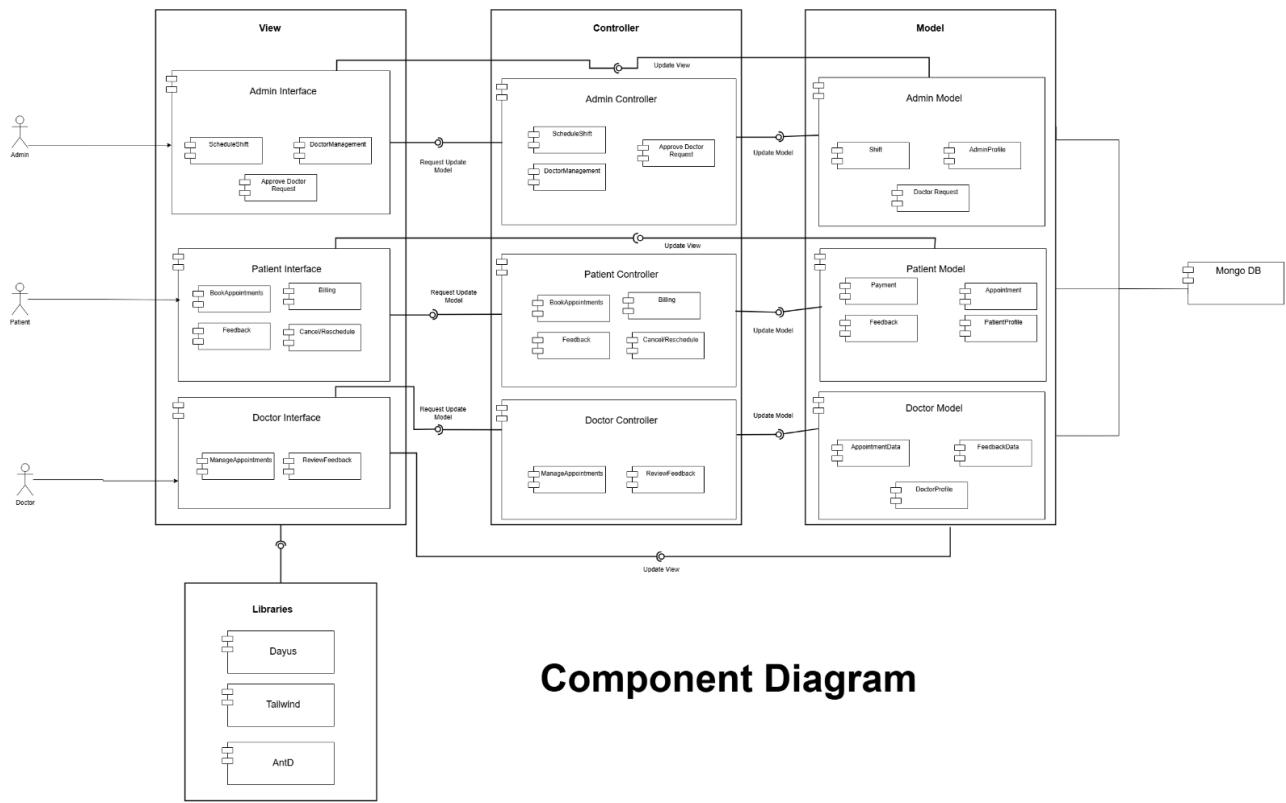
- cors for cross-origin requests.
- **Interfaces:**
 - API endpoints (/admin, /appointments, etc.) via routes for Frontend Component.
 - MongoDB connection via mongoose.

Database Component (MongoDB)

- **Description:** Stores collections (admins, doctors, patients, appointments, feedback, payments, shifts, users).
- **Interfaces:** MongoDB connection interface via mongoose for Backend Component.

Interactions

- **Frontend → Backend:** HTTP requests (via axios) to API endpoints, enabled by cors.
- **Backend (Controller) → Middleware:** Request processing (e.g., authentication with jsonwebtoken, file uploads with multer).
- **Backend (Controller) → Model:** Database access via mongoose to interact with Database Component.



Component Diagram

10. Actual Implementation

10.1. Common Pages

10.1.1 Login Page



Login

* Email

* Password
 

[Not a user? Sign Up here](#)

10.1.2 Sign Up Page



Sign Up

* First Name * Last Name

* Gender * Role

* Email * Password
 

* Confirm Password Phone
 

[Already have an account? Login here](#)

10.2. Admin's Pages

10.2.1 Admin's Dashboard

The screenshot shows the Admin's Dashboard for the 'Ikigai' application. On the left sidebar, there are navigation links: Dashboard (selected), Manage Doctors, Doctor Requests, Shifts, Settings, and Logout. The main content area has a header 'Hello, Fatima Ahmed' and a welcome message 'Welcome to the appointment management system!'. A 'My Information' section displays the user's details: Department (Management), Designation (System Admin), and Email (admin1@clinic.com). Below this are two cards: 'Approval Requests' (17 doctor requests) and 'Total Doctors' (17 doctors in the system).

10.2.2 Admin Manage Doctors Page

The screenshot shows the 'Manage Doctors' page under the 'Ikigai' application. The left sidebar includes links for Dashboard, Manage Doctors (selected), Doctor Requests, Shifts, Settings, and Logout. The main area features a 'Manage Doctors' header and a welcome message 'Welcome to the management system!'. A search bar at the top right allows users to search for specific doctors. Below is a table listing nine doctors with columns for First Name, Last Name, Gender, Specialty, Credentials, Fee, and Actions (Edit and Delete buttons). The table rows have alternating light and dark backgrounds.

First Name	Last Name	Gender	Specialty	Credentials	Fee	Actions
Areeba	Khan	female	Dermatology	MBBS, FCPS	2500	Edit Delete
Bilal	Rana	male	Cardiology	MBBS, MD	3000	Edit Delete
Mehwish	Tariq	female	Neurology	MBBS, MRCP	3500	Edit Delete
Saad	Malik	male	Orthopedics	MBBS, MS	3500	Edit Delete
Hina	Shah	female	Ophthalmology	MBBS, FCPS	2500	Edit Delete
Umer	Iqbal	male	Pediatrics	MBBS, DCH	3000	Edit Delete
Maha	Javed	female	Gynecology	MBBS, MCPS	4000	Edit Delete
Arslan	Malik	male	Orthopedics	MBBS, MS Ortho	4000	Edit Delete
Hira	Tariq	female	Ophthalmology	MBBS, FCPS	3000	Edit Delete

10.2.3 Admin Manage Doctors Request

The screenshot shows the 'Manage Requests' section of the 'Ikigai' management system. On the left sidebar, 'Doctor Requests' is highlighted. The main area displays a table of doctor requests with columns: First Name, Last Name, Gender, Specialty, Credentials, Email, and Actions (Approve/Reject). The table contains five entries:

First Name	Last Name	Gender	Specialty	Credentials	Email	Actions
Zainab	Tariq	female	ENT	MBBS	zainab@clinic.com	<button>Approve</button> <button>Reject</button>
Kamran	Shah	male	Urology	MBBS, MS	kamran@clinic.com	<button>Approve</button> <button>Reject</button>
Iqra	Hassan	female	Gastroenterology	MBBS	iqra@clinic.com	<button>Approve</button> <button>Reject</button>
Sana	Jamil	female	Psychiatry	MBBS	sana@clinic.com	<button>Approve</button> <button>Reject</button>
Danish	Ahmed	male	Dermatology	MBBS	danish@clinic.com	<button>Approve</button> <button>Reject</button>

Pagination at the bottom shows page 1 of 4.

10.2.4 Admin Assign Shift Page

The screenshot shows the 'Manage Shifts' section of the 'Ikigai' management system. On the left sidebar, 'Shifts' is highlighted. The main area displays a table of assigned shifts with columns: Doctor Name, Date, Start Time, End Time, Shift Type, and Location. The table contains five entries:

Doctor Name	Date	Start Time	End Time	Shift Type	Location
Areeba Khan	5/1/2025	09:00	17:00	Morning	Skin Ward
Bilal Rana	5/1/2025	17:00	01:00	Evening	Cardiology Block
Mehwish Tariq	5/1/2025	01:00	09:00	Night	Neuro Wing
Saad Malik	5/2/2025	09:00	17:00	Morning	Ortho Block
Hina Shah	5/2/2025	17:00	01:00	Evening	Eye Center

A '+ Assign Shift' button is located in the top right corner of the main area.

10.2.5 Admin Settings Page

The screenshot shows the 'Settings' page of the Ikigai application. On the left is a sidebar with icons for Dashboard, Manage Doctors, Doctor Requests, Shifts, and Settings (which is highlighted). Below the sidebar are Logout and Settings links. The main area has a light blue header with the text 'Settings' and 'Update your information.' Below this is a form with fields for First Name, Last Name, Gender, Department, Designation, Phone, and Office Location, each with a small edit icon. At the bottom right of the form is a 'Save Changes' button.

10.3. Patient's Pages

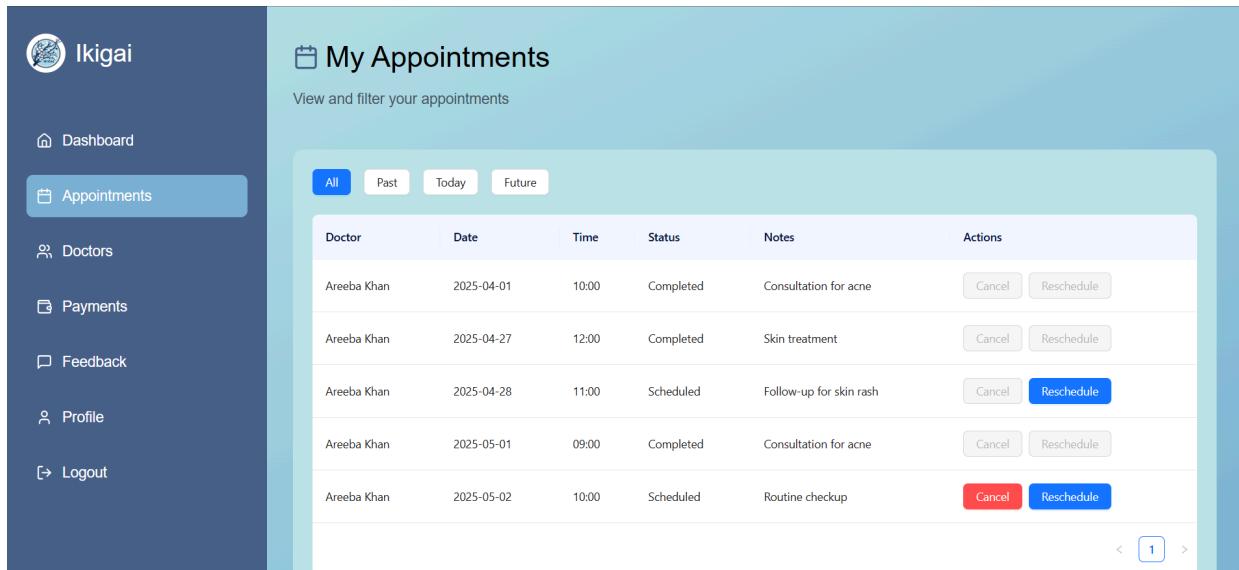
10.3.1. Patient Dashboard

The screenshot shows the Patient Dashboard for Ahmed Khan. The sidebar on the left includes Dashboard, Appointments, Doctors, Payments, Feedback, Profile, and Logout. The main area starts with a greeting 'Hello, Ahmed Khan' and a welcome message 'Welcome to the appointment management system!'. Below this is a 'My Information' section with a table:

Email	Age	Phone
ahmed@clinic.com	29	03001234567

Below the information are two sections: 'Upcoming Appointments' (1 upcoming appointments) and 'Past Appointments' (4 past appointments).

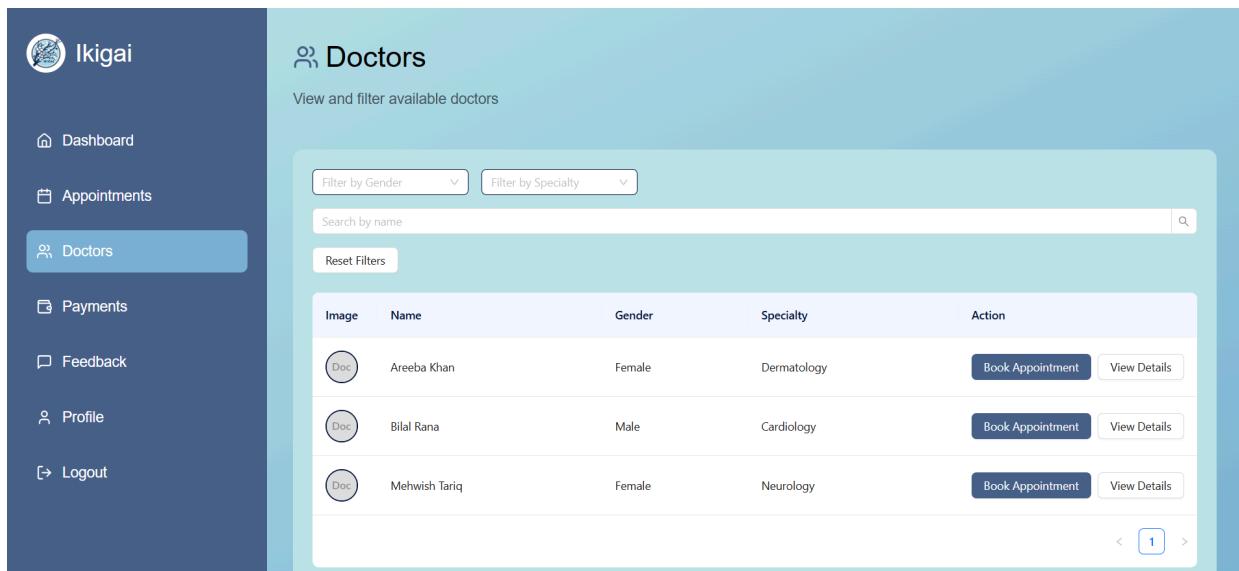
10.3.2. Patient Appointments Page



The screenshot shows the 'My Appointments' page for a patient named Areeba Khan. The left sidebar has a dark blue background with the 'Ikigai' logo at the top. Below it are links: Dashboard, Appointments (highlighted in blue), Doctors, Payments, Feedback, Profile, and Logout. The main content area has a light blue header with the title 'My Appointments' and a subtitle 'View and filter your appointments'. Below this is a table with columns: Doctor, Date, Time, Status, Notes, and Actions. The table lists five appointments for Areeba Khan. Each row includes a 'Cancel' and a 'Reschedule' button. At the bottom right of the table is a pagination control showing page 1 of 1.

Doctor	Date	Time	Status	Notes	Actions
Areeba Khan	2025-04-01	10:00	Completed	Consultation for acne	<button>Cancel</button> <button>Reschedule</button>
Areeba Khan	2025-04-27	12:00	Completed	Skin treatment	<button>Cancel</button> <button>Reschedule</button>
Areeba Khan	2025-04-28	11:00	Scheduled	Follow-up for skin rash	<button>Cancel</button> <button>Reschedule</button>
Areeba Khan	2025-05-01	09:00	Completed	Consultation for acne	<button>Cancel</button> <button>Reschedule</button>
Areeba Khan	2025-05-02	10:00	Scheduled	Routine checkup	<button>Cancel</button> <button>Reschedule</button>

10.3.3. Patient Doctors/Book Appointments



The screenshot shows the 'Doctors' page. The left sidebar has a dark blue background with the 'Ikigai' logo at the top. Below it are links: Dashboard, Appointments (highlighted in blue), Doctors (highlighted in blue), Payments, Feedback, Profile, and Logout. The main content area has a light blue header with the title 'Doctors' and a subtitle 'View and filter available doctors'. Below this are two dropdown filters: 'Filter by Gender' and 'Filter by Specialty'. There is also a search bar with placeholder text 'Search by name' and a magnifying glass icon. A 'Reset Filters' button is located below the search bar. The main content is a table with columns: Image, Name, Gender, Specialty, and Action. The table lists three doctors: Areeba Khan (Female, Dermatology), Bilal Rana (Male, Cardiology), and Mehwish Tariq (Female, Neurology). Each doctor row has a 'Book Appointment' button and a 'View Details' button. At the bottom right of the table is a pagination control showing page 1 of 1.

Image	Name	Gender	Specialty	Action
	Areeba Khan	Female	Dermatology	<button>Book Appointment</button> <button>View Details</button>
	Bilal Rana	Male	Cardiology	<button>Book Appointment</button> <button>View Details</button>
	Mehwish Tariq	Female	Neurology	<button>Book Appointment</button> <button>View Details</button>

10.3.4. Patient Payments

The screenshot shows the Ikigai mobile application. On the left is a dark blue sidebar with the following navigation options:

- Dashboard
- Appointments
- Doctors
- Payments** (highlighted with a blue background)
- Feedback
- Profile
- Logout

The main content area has a light blue header with the title "\$ Payments" and a subtitle "View and manage your payments". Below this is a table with the following data:

Doctor Name	Appointment Date	Time	Amount	Status	Action
Areeba Khan	2025-05-01	09:00	PKR 2500	Pending	<button>Pay Now</button>
Areeba Khan	2025-05-02	10:00	PKR 2500	Pending	N/A

At the bottom right of the table are navigation arrows: < (disabled), 1 (selected), and >.

10.3.5. Patient Feedback

The screenshot shows the Ikigai mobile application. On the left is a dark blue sidebar with the following navigation options:

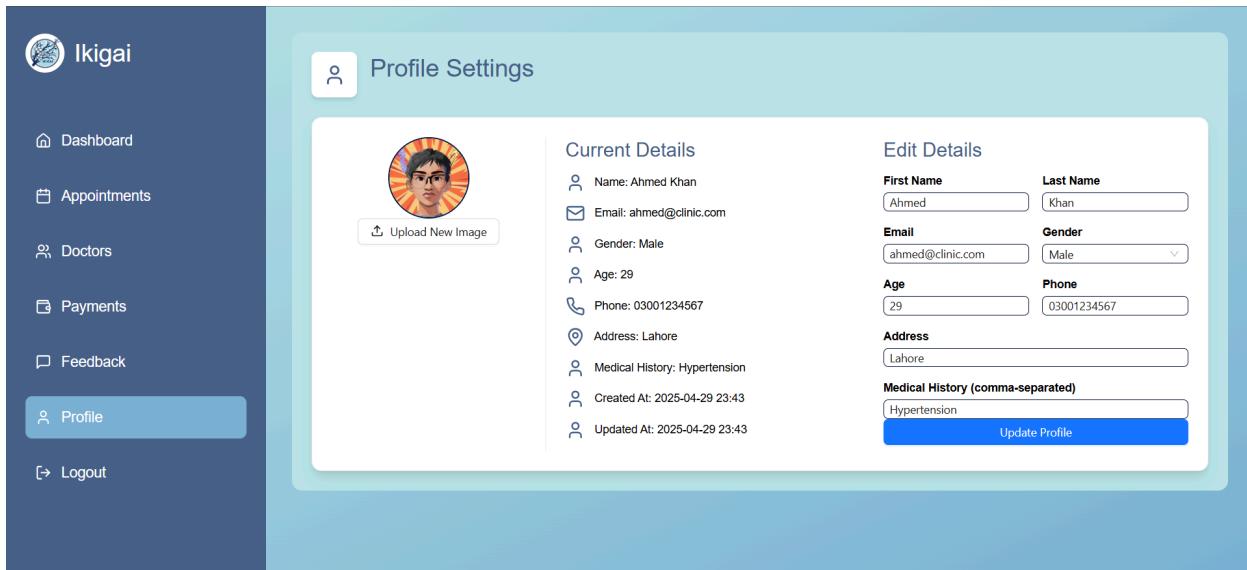
- Dashboard
- Appointments
- Doctors
- Payments
- Feedback** (highlighted with a blue background)
- Profile
- Logout

The main content area has a light blue header with the title "Feedback" and a subtitle "Submit feedback for completed appointments". Below this is a table with the following data:

Doctor Name	Appointment Date	Time	Status	Action
Areeba Khan	2025-04-01	10:00	Completed	<button>Submit Feedback</button>
Areeba Khan	2025-04-27	12:00	Completed	<button>Submit Feedback</button>
Areeba Khan	2025-04-28	11:00	Scheduled	N/A
Areeba Khan	2025-05-02	10:00	Scheduled	N/A

At the bottom right of the table are navigation arrows: < (disabled), 1 (selected), and >.

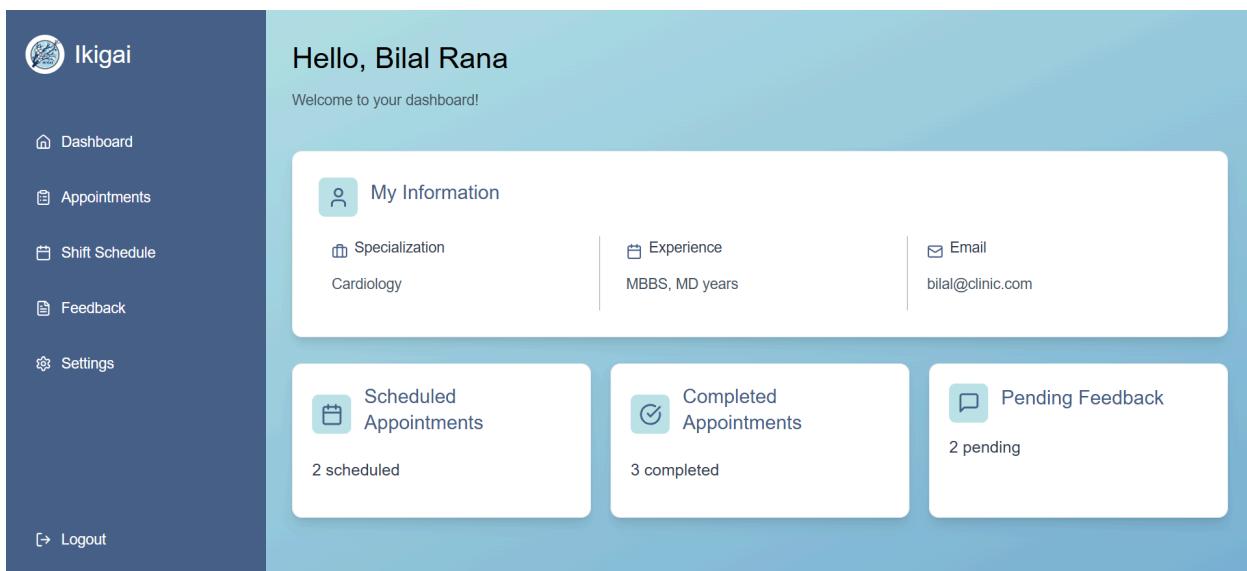
10.3.6. Patient Profile



The screenshot shows the 'Profile Settings' page of the Ikigai application. On the left is a sidebar with navigation links: Dashboard, Appointments, Doctors, Payments, Feedback, Profile (which is highlighted in blue), and Logout. The main content area has a header 'Profile Settings' with a user icon. It features a circular profile picture of a person with a yellow and orange background, and a button to 'Upload New Image'. Below this is a section titled 'Current Details' listing personal information: Name: Ahmed Khan, Email: ahmed@clinic.com, Gender: Male, Age: 29, Phone: 03001234567, Address: Lahore, Medical History: Hypertension, Created At: 2025-04-29 23:43, and Updated At: 2025-04-29 23:43. To the right is a 'Edit Details' form with fields for First Name (Ahmed), Last Name (Khan), Email (ahmed@clinic.com), Gender (Male), Age (29), Phone (03001234567), Address (Lahore), and Medical History (comma-separated) (Hypertension). A blue 'Update Profile' button is at the bottom.

10.4. Doctor's Pages

10.4.1. Doctor Dashboard



The screenshot shows the 'Hello, Bilal Rana' dashboard for a doctor. The sidebar includes links for Dashboard, Appointments, Shift Schedule, Feedback, Settings, and Logout. The main area starts with a 'My Information' section showing Specialization (Cardiology), Experience (MBBS, MD years), and Email (bilal@clinic.com). Below this are three cards: 'Scheduled Appointments' (2 scheduled), 'Completed Appointments' (3 completed), and 'Pending Feedback' (2 pending).

10.4.2. Doctor Appointments Page

The screenshot shows the 'Appointments' section of the Ikigai application. At the top, there are four buttons: 'All', 'Past', 'Today', and 'Future'. Below is a table with columns: Patient Name, Email, Phone, Date, Time, Status, Notes, and Actions. The table contains five rows of appointment data:

Patient Name	Email	Phone	Date	Time	Status	Notes	Actions
Sarah Ali	sarah@clinic.com	03011234567	01/05/2025	17:00	Completed	Chest pain follow-up	No actions available
Rehan Iqbal	rehan@clinic.com	03451230001	03/05/2025	09:30	4908 min left	Heart palpitations checkup	<button>Complete</button> <button>Cancel</button>
Sana Khan	sana.khan@clinic.com	03451230002	04/05/2025	17:30	Completed	Follow-up for chest discomfort	No actions available
Bilal Zaman	bilal@clinic.com	03451230003	05/05/2025	02:00	Cancelled	Blood pressure consultation	No actions available
Iqra Latif	iqra@clinic.com	03451230004	06/05/2025	10:00	9258 min left	Cardiac screening	<button>Complete</button> <button>Cancel</button>

At the bottom right, there are navigation buttons for page 1, 2, and >.

10.4.3. Doctor Shifts Page

The screenshot shows the 'Shifts' section of the Ikigai application. At the top, there are four buttons: 'All', 'Past', 'Today', and 'Future'. Below is a table with columns: Date, Start Time, End Time, Shift Location, Shift Type, and Status. The table contains five rows of shift data:

Date	Start Time	End Time	Shift Location	Shift Type	Status
01/05/2025	17:00	01:00	Cardiology Block	evening	2478 min left
03/05/2025	09:00	17:00	Cardiology Block	morning	4878 min left
04/05/2025	17:00	01:00	Cardiology Block	evening	6798 min left
05/05/2025	01:00	09:00	Cardiology Block	night	7278 min left
06/05/2025	09:00	17:00	Cardiology Block	morning	9198 min left

At the bottom right, there are navigation buttons for page 1, 2, and >.

10.4.4. Doctor Feedback Page

The screenshot shows the 'Feedback' section of the Ikigai application. On the left is a sidebar with navigation links: Dashboard, Appointments, Shift Schedule, Feedback (which is currently selected), and Settings. Below these are Logout and a user profile icon. The main content area has a title 'Feedback' with a comment icon. It displays a table of patient reviews:

Patient Name	Feedback	Rating	Appointment Date	Status	Actions
Sana Khan	Thorough and attentive care.	4.9	04/05/2025	reviewed	Reviewed Pending
Taimoor Saeed	Clear explanation of treatment plan.	4.6	07/05/2025	reviewed	Reviewed Pending
Bilal Zaman	Cancelled due to scheduling conflict.	3.5	05/05/2025	pending	Reviewed Pending
Rehan Iqbal	Professional but session felt rushed.	4.3	03/05/2025	pending	Reviewed Pending
Iqra Latif	Very knowledgeable cardiologist.	4.7	06/05/2025	reviewed	Reviewed Pending

At the bottom right of the table, there are navigation arrows and a page number '1'.

10.4.5. Doctor Profile Page

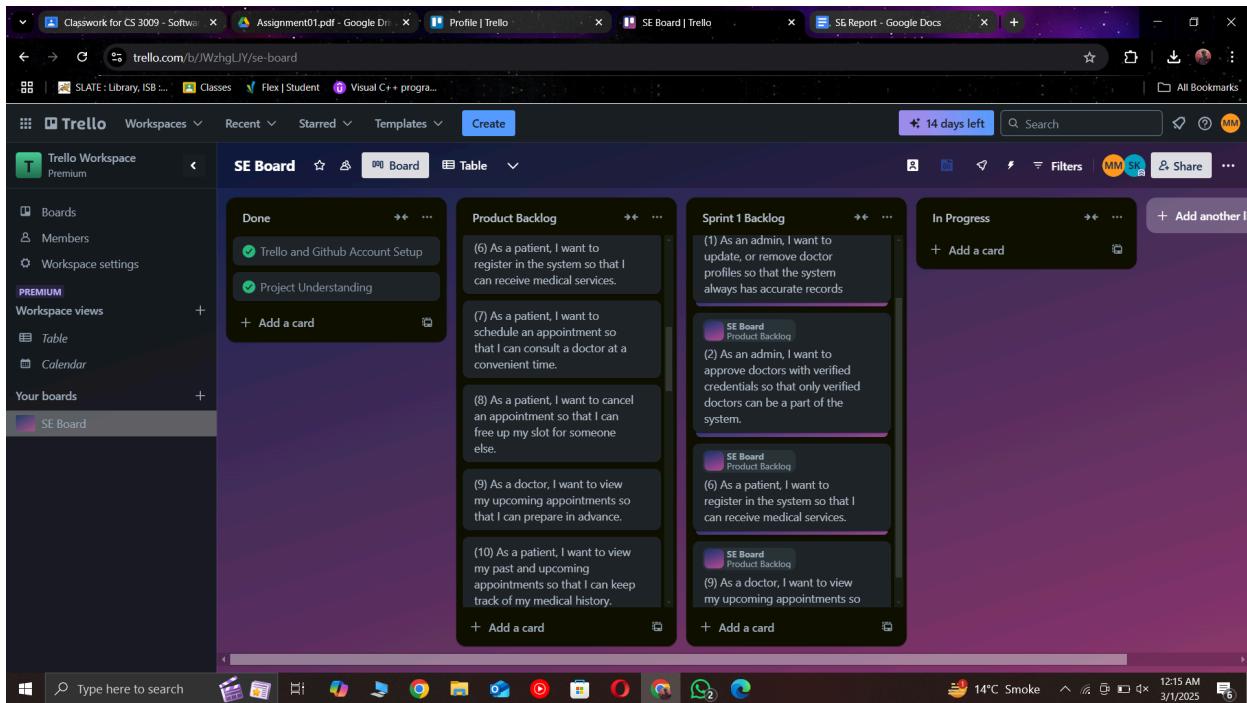
The screenshot shows the 'Profile Settings' section of the Ikigai application. The sidebar on the left includes: Dashboard, Appointments, Shift Schedule, Feedback, Settings, and Logout. The main content area has a title 'Profile Settings' with a person icon. It features a circular placeholder for a profile picture labeled 'Doctor' with a 'Upload New Image' button. To the right are two columns: 'Current Details' and 'Edit Details'. The 'Current Details' column lists: Email (bilal@clinic.com), Speciality (Cardiology), Consultation Fee (3000), Phone (03221234567), Location (Cardiology Block), and Gender (male). The 'Edit Details' column contains form fields for: First Name (Bilal), Last Name (Rana), Email (bilal@clinic.com), Speciality (Cardiology), Credentials (MBBS, MD), Consultation Fee (3000), Phone (03221234567), Location (Cardiology Block), Gender (Male), and a 'New Password (optional)' field with a placeholder '*****'. A blue 'Update Profile' button is at the bottom right.

11. Sprint 3 Burndown Chart

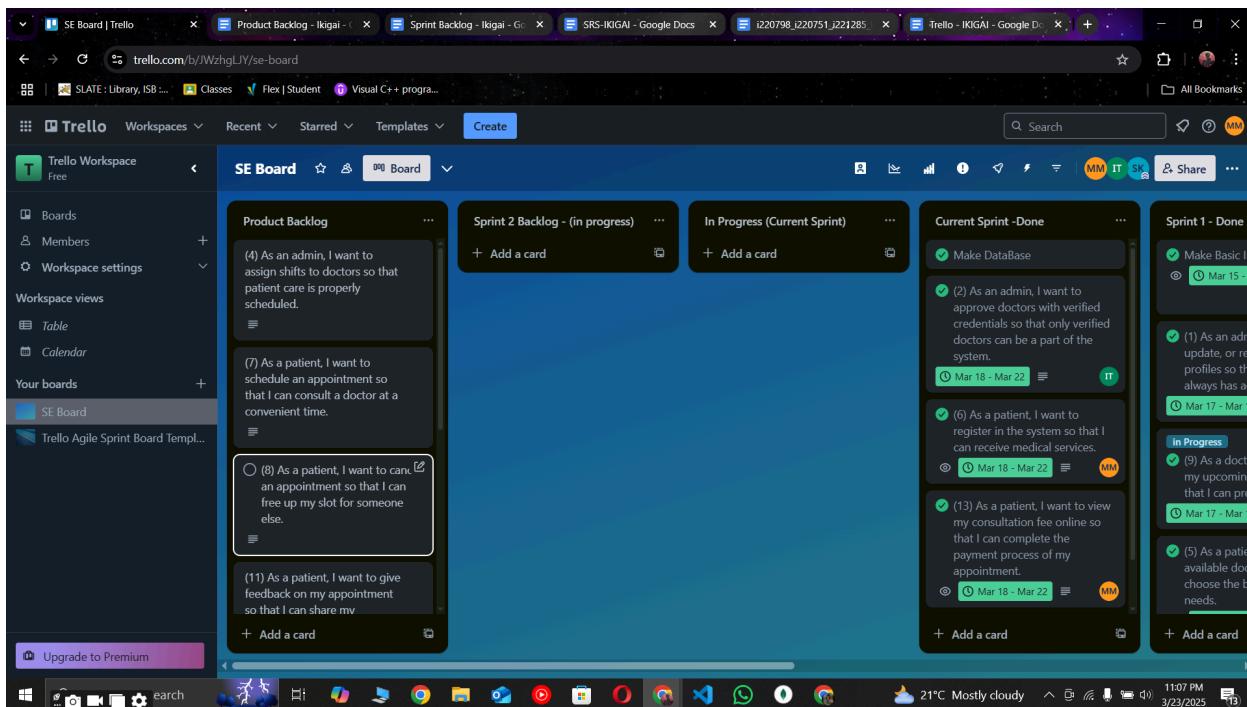


12. Trello Board

12.1. Sprint 1 Start



12.2. Sprint 2 End



SE Board | Trello

trell.com/b/WzghUV/se-board

SLATE : Library, ISB... Classes Flex Student Visual C++ program...

Trello Workspaces Recent Starred Templates Create

Search Share

MM IT SKA

SE Board

Boards Members Workspace settings

Table Calendar

Your boards SE Board Trello Agile Sprint Board Temp...

Upgrade to Premium

Current Sprint - Done

- Make DataBase
- (2) As an admin, I want to approve doctors with verified credentials so that only verified doctors can be a part of the system.
- (6) As a patient, I want to register in the system so that I can receive medical services.
- (13) As a patient, I want to view my consultation fee online so that I can complete the payment process of my appointment.

Sprint 1 - Done

- Make Basic Interface
- (1) As an admin, I want to update, or remove doctor profiles so that the system always has accurate records
- In Progress
- (9) As a doctor, I want to view my upcoming appointments so that I can prepare in advance.
- (5) As a patient, I want to view available doctors so that I can choose the best one for my needs.

Sprint 0 - Complete

- Trello and Github Account Setup
- Project Understanding
- Creating User Stories
- Make Use Case Diagram
- Make Sequence Diagrams (3 Usecases)
- Make Class Diagram

Meeting Backlog

- Daily Meeting (1)
- Daily Meeting (2)
- Daily Meeting (3)
- Daily Meeting (4)
- Daily Meeting (5)

+ Add a card

12.3. Sprint 3 End

Beta Search

SE Board

Product Backlog Sprint 3 Backlog - (in progress) In Progress (Current Sprint) Current Sprint - Done Sprint 2 - Done Sprint 1 - Done Sprint 0 - Complete

+ Add a card + Add a card

Get started with Persons Burndowns by Corrello View in TeamGantt Burndown Charts for Trello Corrello Beta

11:07 PM 3/23/2025

As an admin, I want to assign shifts to doctors so that patient care is properly scheduled.

As a patient, I want to schedule an appointment so that I can consult a doctor at a convenient time.

As a patient, I want to cancel an appointment so that I can free up my slot for someone else.

As a patient, I want to give feedback on my appointment so that I can share my experience with the doctor.

Make DataBase

As an admin, I want to approve doctors with verified credentials so that only verified doctors can be a part of the system.

As a patient, I want to register in the system so that I can receive medical services.

As a patient, I want to view my consultation fee online so that I can complete the payment process of my appointment.

Make Sprint Backlog Doc

Make Product Backlog Doc

Make Basic Interface

As an admin, I want to update, or remove doctor profiles so that the system always has accurate records

As a doctor, I want to view my assigned shifts so that I can manage my time effectively.

As a patient, I want to view available doctors so that I can choose the best one for my needs.

Trello and Github Account Setup

Project Understanding

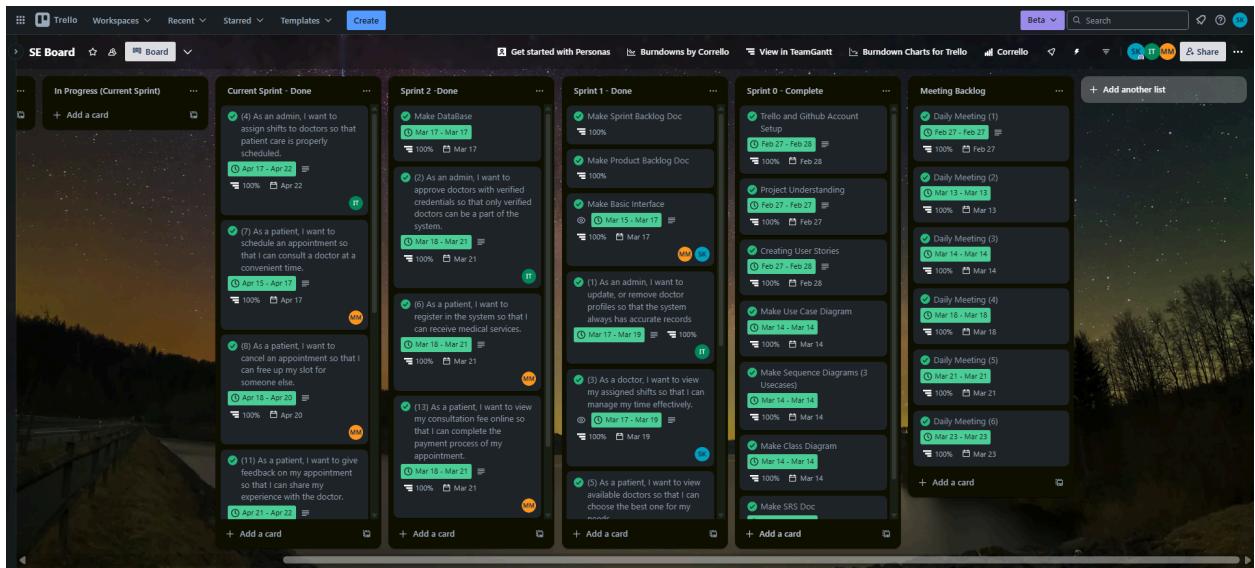
Creating User Stories

Make Use Case Diagram

Make Sequence Diagrams (3 Usecases)

Make Class Diagram

Make SRS Doc



13. Black Box Testing

The screenshot shows the VS Code interface with multiple tabs open. The left sidebar has a tree view of files under 'FRONTEND-JS'. The main area displays the output of a Cypress E2E test run. The terminal shows the following test results:

```
Healthcare System E2E Tests
  User Story 2: Approve Doctors with Verified Credentials
    ✓ TC_04: Approve Doctor with Valid Credentials (397ms)
    ✓ TC_05: Reject Doctor with Invalid Credentials (Empty) (1240ms)
  User Story 3: Filter, View Assigned Shifts as a Doctor
    ✓ TC_06: View Past Shifts (1532ms)
    ✓ TC_07: View Future Shifts with Boundary Date (1222ms)
  User Story 4: Assign Shifts to Doctors
    ✓ TC_08: Assign Shift with Valid Date/Time (2902ms)
    ✓ TC_09: Assign Shift with Invalid Date (Past) (2182ms)
    ✓ TC_46: Assign Shift with Min Date (Tomorrow) (3072ms)
  User Story 5: Filter, View Available Doctors as a Patient
    ✓ TC_11: Filter Doctors with Invalid specialty (1990ms)
    ✓ TC_49: Filter Doctors by Gender (1590ms)
    ✓ TC_50: Search Doctors by Name (2032ms)
  User Story 6: Register as a Patient
    ✓ TC_12: Register with Valid Data (6275ms)
    ✓ TC_13: Register with Invalid Email (4679ms)
    ✓ TC_14: Register with Short Password (3892ms)
    ✓ TC_36: Register with Valid Name Characters (5061ms)
    ✓ TC_37: Register with Invalid Name Characters (4660ms)
    ✓ TC_41: Register with Min Age (0) (4616ms)
    ✓ TC_42: Register with Max Age (120) (4761ms)
    ✓ TC_43: Register with Invalid Age (158) (4340ms)
  User Story 7: Schedule an Appointment
    ✓ TC_16: Schedule Appointment (2704ms)
  User Story 8: Cancel an Appointment
    ✓ TC_17: Cancel Upcoming Appointment (1337ms)
    ✓ TC_18: Cancel Non-Existent Appointment (1184ms)
```

At the bottom, it says "21 passing (1m)".

The screenshot shows the VS Code interface with multiple tabs open. The left sidebar displays the project structure under 'FRONTEND-JS'. The 'TERMINAL' tab shows the output of a Cypress E2E test run. The terminal output includes:

```
21 passing (1m)
(Results)
Tests: 21
Passing: 21
Failing: 0
Pending: 0
Skipped: 0
Screenshots: 0
Video: false
Duration: 1 minute, 8 seconds
Spec Ran: Tests.cy.js
```

=====
(Run Finished)

Spec	Tests	Passing	Failing	Pending	Skipped
Tests.cy.js	01:08	21	21	-	-
All specs passed!	01:08	21	21	-	-

D:\University\Semester6\SE\CMS-front\frontend-js\|

14. White Box Testing

The screenshot shows the VS Code interface with the project 'clinic-appointment-system-backend' open. The Explorer sidebar shows files like 'patient.controller.js', 'jest.config.js', 'doctor.controller.test.js', and 'package.json'. The terminal tab displays the command 'npm run test:coverage' and its output, which includes a coverage report table and summary statistics.

File	% Stats	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	75.4	65.75	83.67	75.52	
controllers	74.39	65.75	83.67	74.48	59-51,107-108,124,187-188,223-224,243,279-280,300-301,321,355-356,386-387,423-424,485-486
admin.controller.js	84.76	77.77	100	83.91	29-68,81-124,172-173,286-229,258-259,301-302,353-354,385-427
doctor.controller.js	58.24	54.05	61.53	58.85	..,547,558-560,576-577,587,618-619,632,635,641,666-667,676,691-692,701,707,725-726,732-733,744,763-764
patient.controller.js	78.77	67.07	85.71	79.2	
models	100	100	100	100	
AdminProfile.js	100	100	100	100	
Appointment.js	100	100	100	100	
DoctorProfile.js	100	100	100	100	
DoctorRequest.js	100	100	100	100	
Feedback.js	100	100	100	100	
PatientProfile.js	100	100	100	100	
Payment.js	100	100	100	100	
Shift.js	100	100	100	100	
User.js	100	100	100	100	

Test Suites: 3 passed, 3 total
Tests: 76 passed, 76 total
Snapshots: 0 total
Time: 9.477 s
Ran all test suites.

15. Work Division

15.1. Mavra Mehak Gill - 22i-0798

15.1.1. Contribution Area

Scrum Master Duties, Frontend + Backend Development (Patient Features), Testing, UI Design

15.1.2. Details

Mavra focused on developing all patient-related pages, including patient registration, viewing available doctors, scheduling, canceling, rescheduling appointments, viewing consultation fees, tracking appointment history, and submitting feedback (**User Stories 5, 6, 7, 8, 10, 11, 13, 14**). She conducted extensive testing, writing and executing black-box and white-box test cases to ensure the reliability of patient-facing features. Mavra also contributed to 50% of the overall UI design, creating a user-friendly and responsive interface. As the Scrum

Master, she managed the project timeline, divided tasks, facilitated daily meetings, and ensured timely sprint deliveries using agile practices.

15.2. Maryum Tanvir - 22i-0751

15.2.1. Contribution Area

Frontend + Backend Development (Admin Features), Database Schema Design, UI Design, API Testing, GitHub Maintenance

15.2.2. Details

Maryum focused on implementing all admin-related user stories, including updating or removing doctor profiles, approving doctors with verified credentials, and assigning shifts to doctors (**User Stories 1, 2, 4**). She designed the database schema for the application using MongoDB, ensuring efficient storage and retrieval of data for doctors, patients, and appointments. Maryum also contributed the remaining 50% of the overall UI design, collaborating with Mavra to create a cohesive and responsive interface. She assisted in testing by developing test cases to validate admin functionalities and maintained the GitHub repository, managing commits and ensuring version control after each sprint.

15.3. Salman Khan - 22i-1285

15.3.1. Contribution Area

Frontend + Backend Development (Doctor Features), Authentication Module (Signup/Login), Documentation, Trello Board Maintenance

15.3.2. Details

Salman focused on implementing all doctor-related user stories, including viewing assigned shifts, viewing upcoming appointments, viewing feedback, and updating appointment completion status and availability (**User Stories 3, 9, 12, 15**). He also developed the signup and login pages for all user roles, ensuring secure authentication and proper form validation. Additionally, Salman was responsible for creating and

maintaining the project documentation, ensuring all reports were comprehensive and up to date. He also managed the Trello board, keeping tasks organized and tracking sprint progress

16. Retrospective

16.1. What went well

16.1.1. Modular Architecture:

The project itself followed **Module Pattern & MVC architecture** which promotes modularity. The system was separated into backend (controllers, models, routes) and frontend (components, pages, services).

16.1.2. Robust Backend Implementation:

Controllers (admin.controller.js, patient.controller.js) effectively handled CRUD operations, leveraging Mongoose schemas (Appointment.js, User.js) via the **Repository Pattern**.

16.1.3. Frontend Usability:

React dashboards (AdminDashboard.jsx, PatientDashboard.jsx) provided role-specific UIs, using AuthContext.jsx and useAuth.js for state management.

16.2. What could be Improved

16.2.1. Testing Coverage

The tests made do not cover every functionality the project has to offer.

16.2.2. Performance Optimization

The project works well may not work properly when scaled for really high appointment volumes

16.3. Actions

16.3.1. Expand Testing

Add more tests using Cypress & Jest to Test the project completely.

16.3.2. Optimize Performance

Implement MongoDb indexes for frequent queries and work more on user-friendly GUI.

17. Review

17.1. System Quality

- **Functionality:** Delivers core features like appointment management, role-based dashboards, authentication, and feedback/payment handling via patient.controller.js, AuthContext.jsx, and MongoDB schemas.
- **Reliability:** Robust backend with unit tests (patient.controller.test.js) and middleware.
- **Maintainability:** Modular MVC structure (controllers/ models) eases updates, though clear dependencies
- **Scalability:** MongoDB and Express support growth, but large datasets require query indexing for performance.

17.2. Alignment with Requirements

- **User Roles:** Fully supports admin (doctor approvals), doctor (shift management), and patient (appointment booking) workflows.
- **Appointment System:** Handles booking, rescheduling, and cancellation, validated by tests
- **Security:** JWT authentication and input validation ensure endpoint safety.
- **UI/UX:** Intuitive dashboards, with reusable components..
- **Documentation:** Elaborative and clean UML diagrams and README is provided

17.3. Strengths

- Clean backend with separated controllers, models, and routes.
- Effective frontend state management using React context (AuthContext.jsx) and hooks (useAuth.js).

- Comprehensive MongoDB schema
- Solid testing foundation with unit and end-to-end tests.

17.4. Future Optimization Recommendations

- Further enhancements in concurrent user load handling and React performance.
- Strengthen security with file upload limits and rate limiting.
- Further expansion of the business logic to cater more user requirements

18. Lesson Learnt

Throughout the development of the Ikigai/I-CARE project, the team gained valuable experience in both technical and collaborative aspects. On the technical side, we learned the importance of modular architecture and clean code practices, especially when working with MERN stack, as it ensured that the project remained scalable and maintainable. We also gained a deeper understanding of frontend-backend integration. Another key lesson was the significance of unit testing and test coverage—we realized that writing comprehensive test cases requires accounting for edge cases, interactions between components, and asynchronous behaviors, which was more challenging than anticipated.

On the collaborative side, we learned the importance of clear communication and task distribution, ensuring that each team member's strengths were leveraged effectively. Time management and consistent project planning through tools like Trello and Gantt charts played a crucial role in keeping the development process on track. Using GitHub taught us how to collaborate better, from managing commits to resolving merge conflicts, making teamwork smoother.

This project taught us how to balance individual and team responsibilities, tackle challenges together, and grow as developers. We're proud of what we achieved and we hope to continue refining our skills to craft even more optimized, robust and complex web applications.

19. Links

19.1. Github Version Control

Front End:

<https://github.com/MaryumTanvir/clinic-appointment-system-frontend.git>

Back End:

<https://github.com/MaryumTanvir/clinic-appointment-system-backend.git>

19.2. Vercel Live Demo

<https://clinic-appointment-management-system-frontend.vercel.app/>

19.3. Presentation

<https://www.canva.com/design/DAGS6t6nNVY/eYhEsgE7rmcZjZe9ZflXaA/edit>