






MediConnection







Test Results

Key: Pass - , Fail - 






Book Appointment

1. Verify that a patient can request an appointment. 
2. Verify that a physician can reject an appointment request. 
3. Verify that a physician can approve an appointment request. 
4. Verify that a physician or patient can mark an appointment as done. 
5. Verify that the user can view their upcoming appointments in the waiting room. 





Contact Doctor

1. Verify that the patient can start a chat with a physician with whom they have an appointment. 
2. Verify that messages are rendered properly. 
3. Verify that users can delete messages. 
4. Verify that the user can send and receive messages in real-time. 
5. Verify that the user can resume a previous chat session with a physician. 
6. Verify that messages come with notifications. 


Access medication

1. Verify that physicians can add medication. 
2. Verify that patients can view medications assigned to them. 
3. Verify that physicians can delete medications. 
4. Verify that updates to medication on physician side reflects on patient side. 
5. Verify that patients cannot mutate medication list. 

Search Physician

1. Verify that the search bar on the homepage returns relevant results based on the user's search query. 
2. Verify that the user can filter the search results based on physician's name. 
3. Verify that the user can filter the search results based on specialization. 
4. Verify that the user can view the filtered physician's profile. 

Alternate scenarios:

5. Verify that the search bar returns results even if the user misspells the physician's name or specialty. 

Summary of Bugs Encountered

Error 1:

Error encountered: Bug encountered during the compilation of the Project.

Test code: Unit Testing

Root cause: SocketIO was initially placed in the MongoDB creation, which flagged multiple errors.

Solution: Isolate the SocketIO code (Extracted it from local Schema creation).

Error 2:

Error encountered: Patient profile data not being displayed correctly

Test code: Unit test for rendering patient profile component

Root cause: Incorrect data mapping from the API response to the component's state

Solution: Modify the data mapping to match the expected structure of the patient profile component

Error 3:

Error encountered: Unable to create a new appointment

Test code: End-to-end test for the appointment creation workflow and integration testing

Root cause: Missing validation for required appointment details (e.g. date, time, doctor)

Solution: Implement validation checks for required fields and display error messages when necessary

Error 4:

Error encountered: Unable to save prescription data to database.

Test code: Integration test for saving prescription data to the database

Root cause: Incompatible data types between the application and the database schema

Solution: Update the data types in the database schema and ensure proper data conversion during save operations

Error 5:

Error encountered: Chat messages not updating in real time

Test code: End-to-end test for chat system functionality

Root cause: Inefficient use of polling for message updates

Solution: Implement real-time updates using WebSockets to ensure smooth communication between users

Error 6:

Error encountered: available time slots for appointments not displaying correctly

Test code: Unit test for rendering appointment time slots, interface testing

Root cause: Incorrect filtering of available time slots based on doctor's availability

Solution: Improve filtering logic to accurately display available time slots

Error 7:

Error encountered: Inconsistent UI/UX across different pages

Test code: Interface testing

Root cause: Inconsistent usage of Bootstrap components and custom CSS

Solution: Standardize the usage of Bootstrap components and create a shared style guide to maintain consistency

Error 8:

Error encountered: Application crashes when a user tries to access an unauthorized page

Test code: Integration test for user authorization and access control

Root cause: Missing error handling for unauthorized access attempts

Solution: Implement proper error handling and redirect unauthorized users to a suitable error or login page

Error 9:

Error encountered: Slow loading times for patient profiles and page content

Test code: Performance testing

Root cause: Inefficient data fetching and rendering patterns

Solution: Implement lazy loading, and/or caching to improve performance

Error 10:

Error encountered: Security vulnerabilities in the authentication system

Test code: Security testing and penetration testing

Root cause: Weak encryption methods and lack of validation checks

Solution: Use secure encryption algorithms (e.g., bcrypt) and implement server-side validation to prevent unauthorized access and data breaches