Team 5 Project
AppointmentsOnTheGo

By Adam Inga, Jonathan Pacheco, Shiv Bhakta, Shreya Balaji, Ashwin Indurti

Project overview

The purpose of this program is to connect small businesses to people in the area. This will help users be able to get a good insight on the services close to them and set up appointments in a centralized manner, so they do not have to do so much research. This will also help small business get some extra exposure.

This project was developed using plan-driven approach

Usage of requirements: Requirement overview

The system shall let users search businesses by location, specialty, and next available time

The system shall show a profile for each employer with specialties, experience, and a small portfolio.

The system shall show a calendar for each appointment with open time slots.

The system shall let a user book a slot by entering name, email, and phone number.

The system shall create a unique appointment id for every booking.

The system shall provide a message thread tied to the appointment so the customer and business can talk before the service.

The system shall let the customer upload up to three photos for the appointment with a total size limit of twenty-five megabytes.

The system shall allow businesses to approve or decline a pending booking and add a short note if needed.

The system shall let the customer reschedule or cancel an appointment.

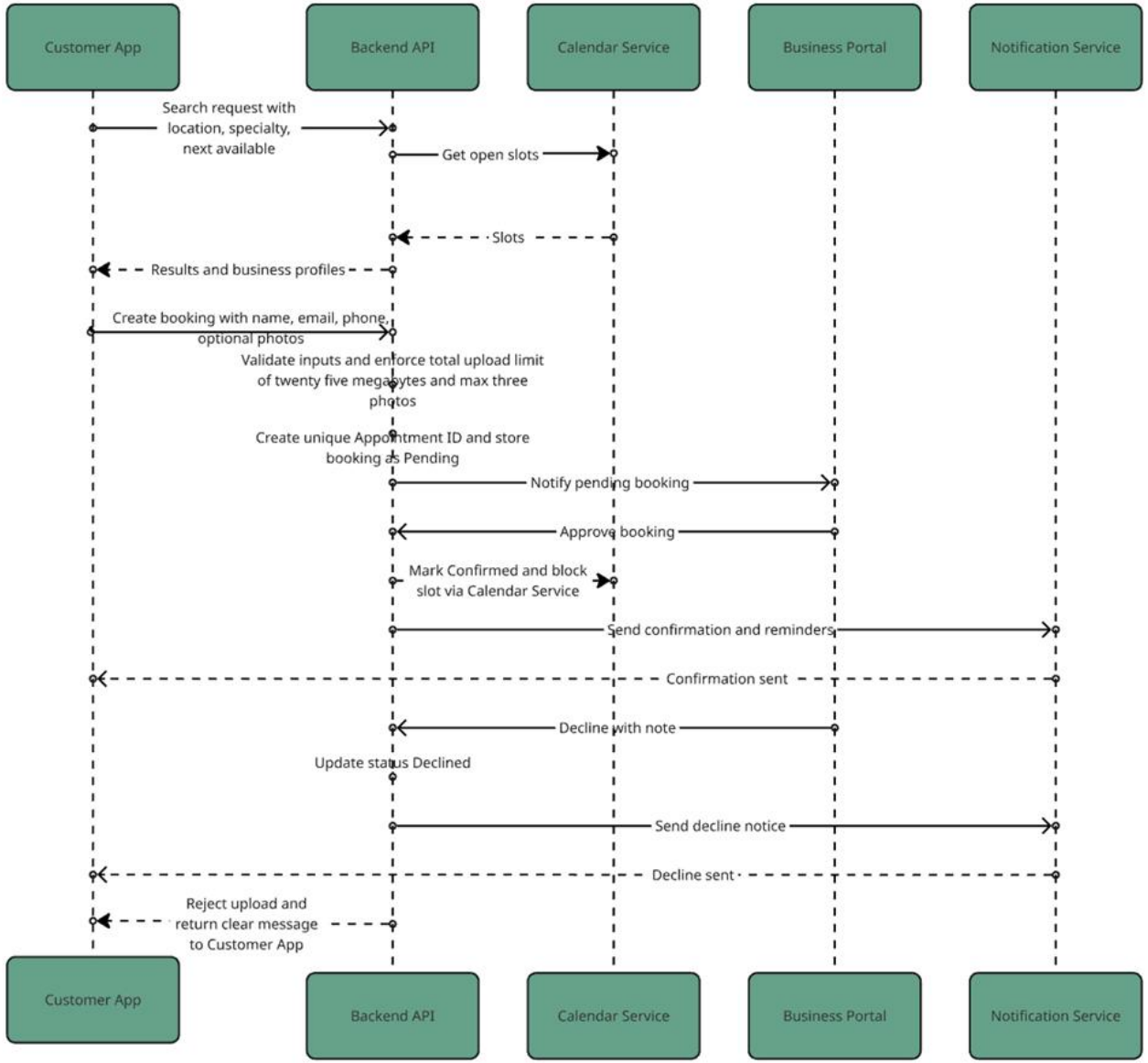
The system shall let businesses mark an appointment as confirmed, completed, or no show.

Usage of requirements

- Non-functional
- Performance: low sizes for file
- Availability: Database

- The requirements we had set up forced us to have a centralized place in order to store information for appointments, small business profiles, and customer accounts.
- Making us consider the **"repository" design pattern**. Leading for us to use a database to manage all of this information in one place.

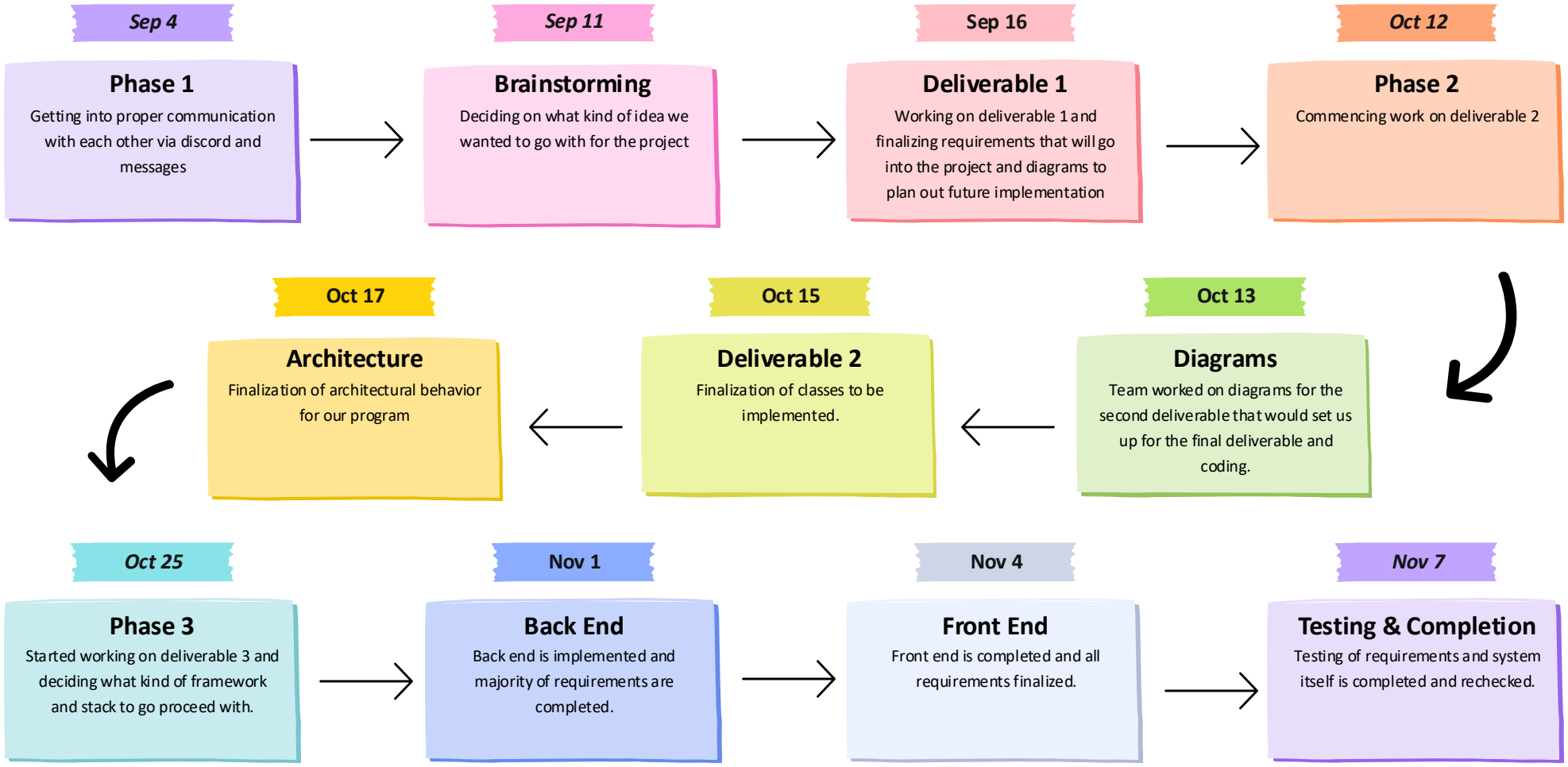
Connection to Deliverable 2



Development Process

- For this system, we followed a plan-driven process, best described by the waterfall model.
- Development moved through separate phases that followed the due dates on the deliverables, where the output of one phase became the input to the next.
- This fit our goal because the platform needed clear up front definition: connecting nearby users to small businesses, showing local services, and enabling appointments in one place.

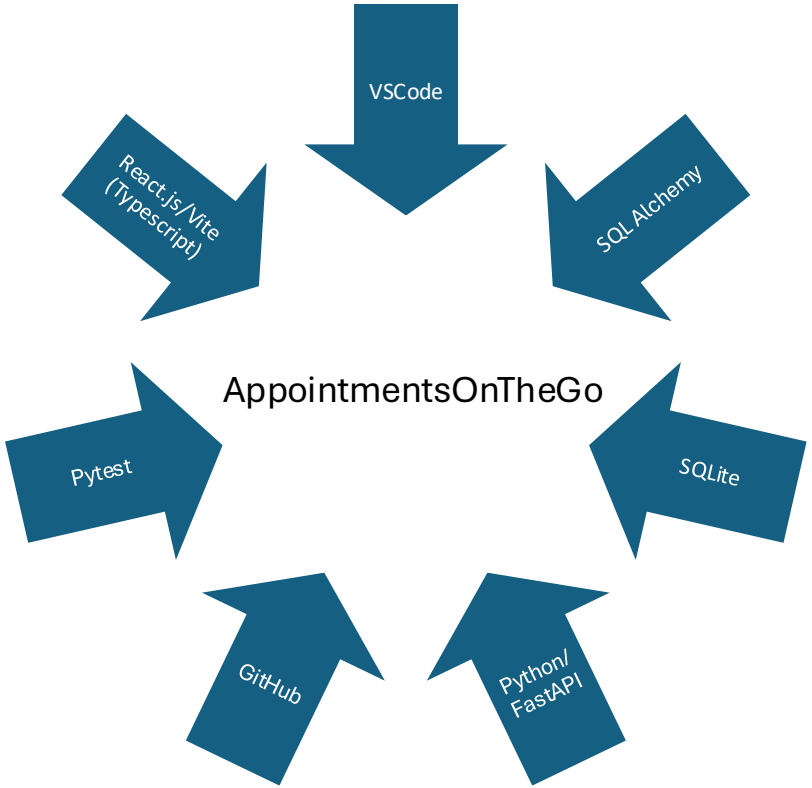
Time Line


Sep 4
Phase 1
Getting into proper communication with each other via discord and messages
Sep 11
Brainstorming
Deciding on what kind of idea we wanted to go with for the project
Sep 16
Deliverable 1
Working on deliverable 1 and finalizing requirements that will go into the project and diagrams to plan out future implementation
Oct 12
Phase 2
Commencing work on deliverable 2
Oct 13
Diagrams
Team worked on diagrams for the second deliverable that would set us up for the final deliverable and coding.
Oct 15
Deliverable 2
Finalization of classes to be implemented.
Oct 17
Architecture
Finalization of architectural behavior for our program
Oct 25
Phase 3
Started working on deliverable 3 and deciding what kind of framework and stack to go proceed with.
Nov 1
Back End
Back end is implemented and majority of requirements are completed.
Nov 4
Front End
Front end is completed and all requirements finalized.
Nov 7
Testing & Completion
Testing of requirements and system itself is completed and rechecked.

Implementation Summary

Implementation Summary

Tools Used

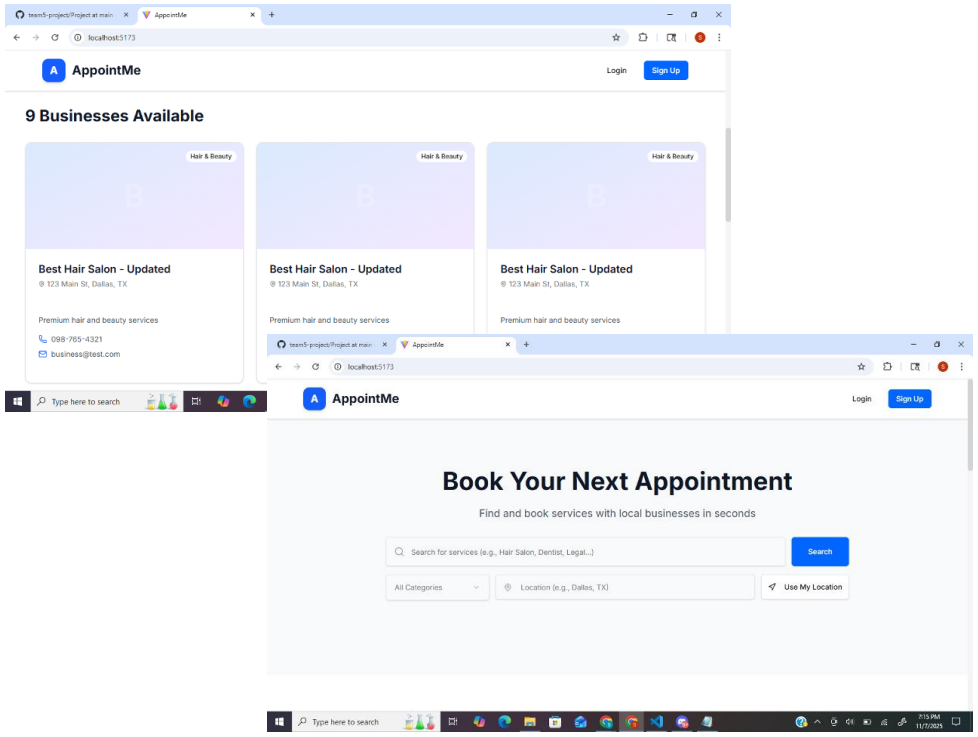


Modularity, Simplicity, and Reusability

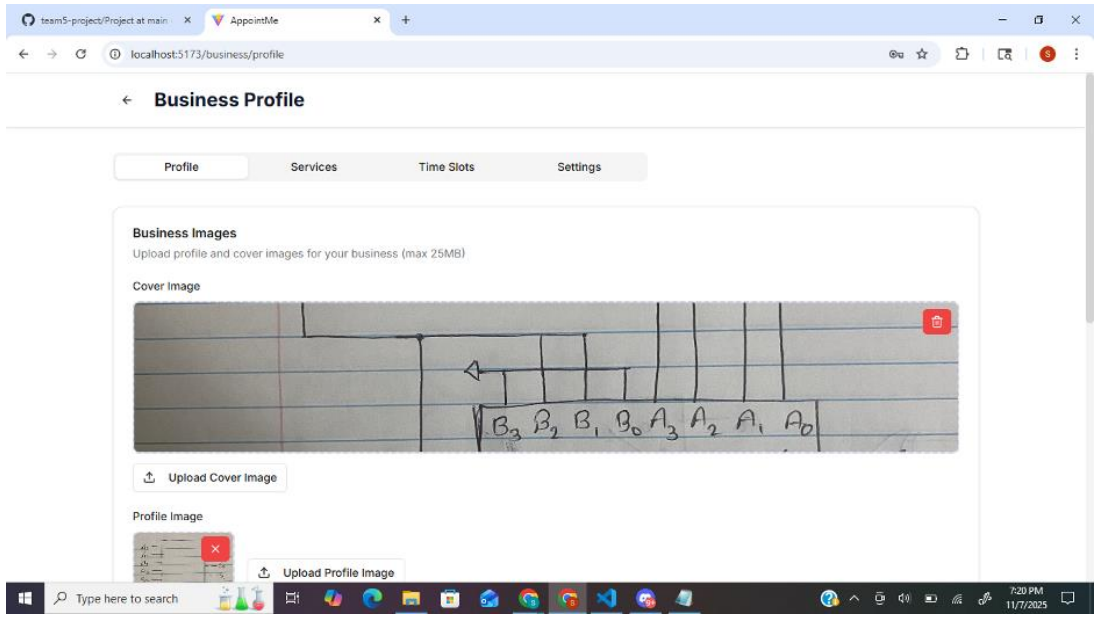
- React.js - Component based
- Only Necessary Features
- Reusable components

Requirements Implementation

The system shall let users search businesses by location, specialty, and next available time (Requirement 1)



The system shall show a profile for each employer with specialties, experience, and a small portfolio. (Requirement 3)

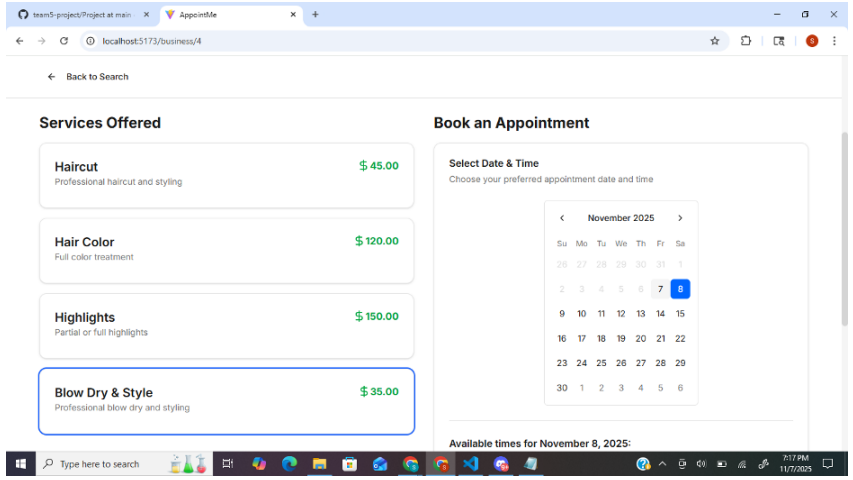


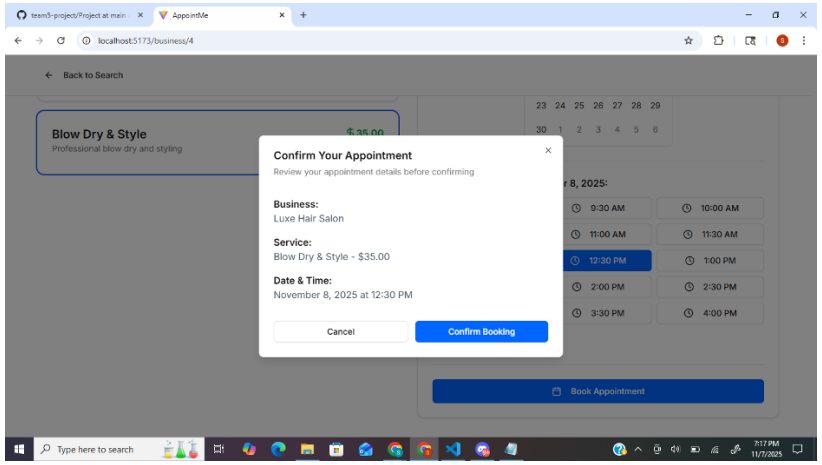
Requirements Implementation (cont'd)

- The system shall show a calendar for each appointment with open time slots. (Requirement 4)

- The system shall let a user book a slot by entering name, email, and phone number. (Requirement 5)
- The system shall create a unique appointment id for every booking. (Requirement 6)

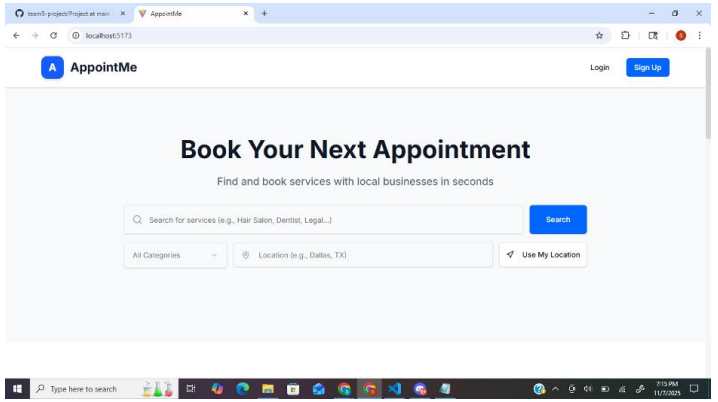
- The system shall provide a message thread tied to the appointment so the customer and business can talk before the service. (Requirement 8)

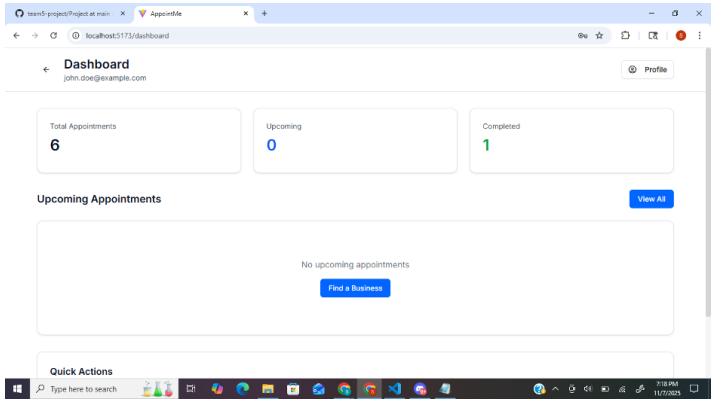


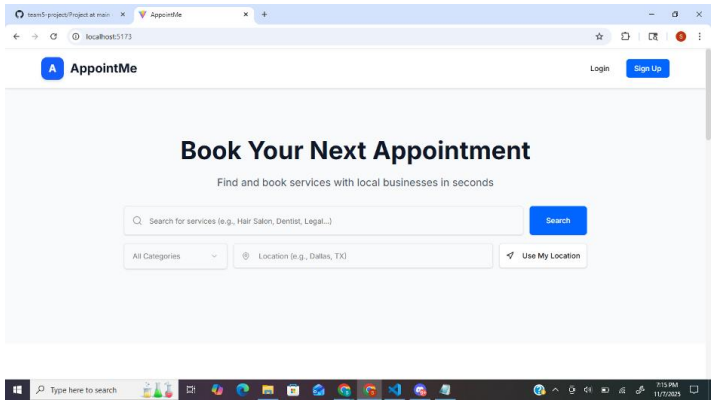


Requirements Implementation (cont'd)

- The system shall let the customer upload up to three photos for the appointment with a total size limit of twenty-five megabytes. (Requirement 9)
- The system shall allow businesses to approve or decline a pending booking and add a short note if needed. (Requirement 10)
- The system shall let the customer reschedule or cancel an appointment. (Requirement 11)
- The system shall let businesses mark an appointment as confirmed, completed, or no show. (Requirement 13)







Automated Testing Approach

Tools & Framework

- Used **pytest** for backend automated testing (Python)
- Unit testing focused on authentication logic and migration scripts

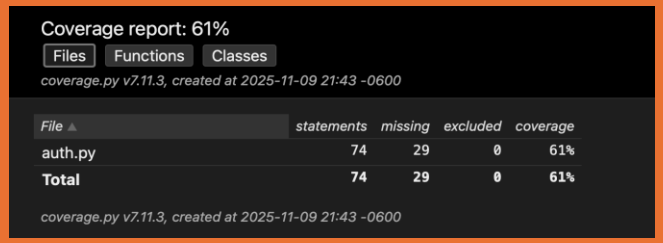
Test Scope

- 33 total unit tests
- Covered:
    - JWT token generation
    - Password hashing
    - Customer/Business login authentication
    - Database migration integrity

Testing Strategy

- Focus on security-critical and logic-heavy functions
- Database-dependent components mocked for reliability and consistency across environments

61% Coverage



Impact of Testing

Early Bug Detection	• Caught edge cases before deployment
Regression Prevention	• Adding new features no longer risked breaking existing logic
Safer Refactoring	• Test safety net allowed confident code improvements
Tests as Documentation	• The suite clarifies how each function is expected to behave

Demo & Results

Landing Page: Search businesses using filters; view business cards.

User Authentication: Customer & business login.

Business Portal: Manage services, availability, and update profile.

Customer Portal: Upcoming Appointments

Booking Update: Real-time slot removal after booking.

Reflection & Lessons Learned

Handling two portals (Customer & Business) was challenging.

Bugs with real-time availability and time zones.

Reusable UI components helped keep the design clean.

AI tools helped speed up debugging and coding.

Future improvements: payments, reviews, and notifications.