Project Report

Parimal Mehta

Here is the writeup for the Conversational Booking Agent Project.

## Project Objective

To build a three-party booking system involving a Human user, an AI Assistant, and a Browser Agent. In this case, we are going to book a meeting on Calendly.

## Architecture

A diagram of a server

AI-generated content may be incorrect.

System Design

### Technology Stack

* TypeScript
* React
* Python
* FastAPI
* Playwright
* Gemini
* Node/npm
* Git

#### Why use React/Python/FastAPI/TypeScript?

* I am familiar with these technologies. So, I would be able to finish the project faster.
* Frontdesk uses these technologies.

#### Why use Playwright?

* It would be faster and easier to understand than Selenium.
* It supports multiple browsers. i.e a useful future skill to have.
* Frontdesk uses Stagehand. Since Stagehand uses Playwright, I decided to use it. I did not use Stagehand because AI can be finicky and I did not want to spend time learning it. Playwright would be more efficient/faster.
* It is more reliable and is well documented.
* Why not browserless.io – I did not want to rely on an external service. This is easier to test.

#### Why use Gemini?

* I was initially planning to use OpenAI until I found out it was paid.
* Out of the free options I checked, the Gemini API seems to be very well documented and had a robust free tier with a lot of features and a large request limit.

### Separation of Logic

All frontend logic is within App.tsx.

A .env file in the backend folder is used to save the Calendly booking url and the Gemini API key.

Backend logic is divided into multiple files:

* main.py – Has all logic related to exposing REST APIs for the frontend to call.
* model.py – Has all logic related to communicating with the Gemini API. (dialog management)
* booking.py – Has all logic related to browser automation using Playwright to book a meeting on Calendly.

## Future Work

1. Write automated tests.
2. Make the UI more presentable.
3. Clean up code and debug logs.
4. Rename files, variables, and functions to be more understandable.
5. Make chats from AI and user in the frontend have different colors.