

DocuHealth

Intern: Marcela Billingslea Durini

Intern Manager: Vidushi Seth

Intern Director: Carl Taylor (AR Experiences)

Peer(s): Melanie Zhao Harshitha Janardan

GitHub Repository Link: <https://github.com/marcebd/DocuHealth.git>

OVERVIEW

My project aims to address the inefficiencies in patient data management at public hospitals in Guatemala. Currently, patient files are kept on paper folders, which can lead to errors and miscommunication between healthcare providers.

To solve these problems, my project proposes a digital solution that streamlines patient data management. By using facial recognition technology, doctors can easily find a patient's file.. This information will be stored in a centralized database that can be accessed by any doctor, improving communication and coordination between healthcare providers.

The benefits of this system include improved patient care, reduced errors, and increased efficiency in doctor visits. Overall, this project has the potential to make a significant impact on the quality of healthcare in Guatemala.

- **Category:** Health Care
- **Story:** Enhance patient data management in Guatemala with an innovative app that utilizes facial recognition for seamless patient sign-in. This app features a centralized database that healthcare providers can access to record visit notes, prescribe medications, upload test results, and document diagnoses. Additionally, the app enables providers to print prescriptions directly, ensuring a streamlined and efficient healthcare process.
- **Market:** Healthcare providers in Guatemala and patients of public hospitals.
- **Habit:** Daily for healthcare providers and every doctor's visit for the patient.
- **Scope:** Improve patient data management in public hospitals in Guatemala by including facial recognition sign-in, centralized database, digital record keeping and prescription printing.

PRODUCT SPEC

REQUIRED

- **Login and Search Patient by Name:** As a doctor, I want to be able to log into the app and search for a patient by their name or face so that I can quickly access their medical records.
 - Healthcare provider can login
 - Patient information can be found through facial recognition ([API](#))
 - Healthcare provider can create an account for self and patients
- **Add Prescribed Medications:** As a doctor, I need to add medications that I prescribe to a patient's record so that the pharmacy can verify and dispense the correct medication.
 - Healthcare provider can create / edit / delete / view prescriptions
- **Write Visit Summary:** As a doctor, I need to write a summary of each visit, including observations and any changes in treatment, to ensure continuity of care.
 - Healthcare provider can create / edit / delete / view visit's notes

OPTIONAL

- **Add Test Results:** As a doctor, I want to record the results of medical tests directly into the patient's file to keep an accurate and up-to-date record of their health status.
 - Healthcare provider can create / edit / delete / view test results
- **Add Vital Signs:** As a doctor, I want to add vital signs such as weight, blood pressure, heart rate, and height during each visit to monitor the patient's health over time.
 - Healthcare provider can create / edit / delete / view vital signs
- **Add Diagnosed Conditions:** As a doctor, I need to record any new diagnosed conditions to a patient's health record to keep a comprehensive history of their health.
 - Healthcare provider can create / edit / delete / view diagnosis
- **Add Instructions:** As a doctor, I want to provide clear instructions regarding treatment plans and home care in the patient's record, ensuring they understand their part in managing their health.
 - Healthcare providers can create / edit / delete / view patient instructions.

SCREEN ARCHETYPES

Doctor's Sign In/ Profile Information-

Create an account

Enter your email to sign up for this app

Sign up with email

By clicking continue, you agree to our [Terms of Service](#) and [Privacy Policy](#)

Create Profile



Doctor Name
Add Profile Photo

Username

Email

Complete Legal Name

Specialty

Add More

ID Number

Date of Birth

Gender

Select One

Language

Select One

Create Profile

Doctor's Home Page View-

Hello DoctorName!

...

Patient 1
Patient 2
Patient 3
+

Q Search....

Today's Visit Date
Past Visit Date

Add Visit Notes

Past Visit's Notes

Date: First sentence of notes user can click to open in another tab

Date: First sentence of notes

Date: First sentence of notes

Date: First sentence of notes

Date: First sentence of notes

Date: First sentence of notes

Current Prescriptions

Medicine Name	Dose
Medicine 1	100mg
Medicine 2	100mg
Medicine 3	100mg

Add New Medication

DATA MODEL

INTERNAL DATABASE MODELS:

- **Patient Table:** Stores information about patients, unique patient ID, name, date of birth, facial recognition data and contact information.
- **Doctor's Table:** Doctor ID, name, specialization, contact details and work schedule.
- **Appointments Table:** Record details about appointments, linking patients and doctors. Appointment ID, patient ID, doctor ID, date, time and status.
- **Visit's Table:** Log each visit a patient makes to a healthcare provider. Including visit ID, patient ID, doctor ID, date, summary notes, and any other information.
- **Medications Table:** Entries of medications prescribed to patients, medication ID, name, description, and possible side effects.
- **Conditions Table:** Medical conditions diagnosed in patients, with condition ID, name, description, and severity.
- **Test Results Table:** Stores results of medical tests, test ID, patient ID, doctor ID, date, type of test and result details.

RELATIONSHIPS:

- **Patients to Doctors:** Many to many (a patient can see multiple doctors and a doctor can see multiple patients. Created in the appointments table).
- **Patients to Medications:** Many to many (a patient can have multiple medications and a medication can be prescribed to multiple patients). Tracked by linking patients and medications.
- **Patients to Conditions:** Many to Many (a patient can have multiple conditions, and a condition can affect multiple patients).
- **Patients to Test Results:** One to many (a patient can have multiple test results).

SERVER ENDPOINTS

AUTHENTICATION AND ACCOUNT MANAGEMENT

- General Login
 - Method: POST
 - Endpoint: /api/auth/login
 - Body Parameters: username, password
 - Description: Allows healthcare providers to log in using traditional credentials.
- Account Creation
 - Method: POST
 - Endpoint: /api/accounts/create
 - Body Parameters: accountType, details
 - Description: Allows healthcare providers to create accounts for themselves or patients.

DASHBOARDS

- Get Doctor Dashboard
 - Method: GET
 - Endpoint: /api/doctors/dashboard?doctorId={id}
 - Query Parameters: doctorId
 - Description: Retrieves information for the doctor's dashboard, including upcoming appointments, patient alerts, and recent activity.
- Facial Recognition Patient file
 - Method: GET
 - Endpoint: /api/auth/patient-facial-login
 - Body Parameters: facialImageData
 - Description: Get patient's data through facial recognition
- Facial Recognition Patient file
 - Method: GET

- Endpoint: /api/auth/patient-login
- Body Parameters: facialImageData
- Description: Get patient's data through name search.

DOCTOR AND VISIT MANAGEMENT

- Manage Doctor Visits
 - Method: GET/ POST/PUT/DELETE
 - Endpoint: /api/visits/manage
 - Body Parameters: visitId, patientId, doctorId, details, action
 - Description: Allows healthcare providers to view, create, edit, or delete details of doctor visits. Doctors will be able to create notes on the visit, can go back to the visit notes for updates, gather information or type after visit summary.

PRESCRIPTION MANAGEMENT

- Manage Prescriptions
 - Method: POST/PUT/DELETE
 - Endpoint: /api/prescriptions/manage
 - Body Parameters: prescriptionId, patientId, medicationId, dosage, duration, action
 - Description: Allows healthcare providers to create, edit, or delete prescriptions.
- View Prescriptions
 - Method: GET
 - Endpoint: /api/patients/prescriptions
 - Query Parameters: patientId
 - Description: Allows health care providers to view their current and past prescriptions.

NOTIFICATIONS AND ALERTS

- Get Notifications
 - Method: GET
 - Endpoint: /api/patients/notifications
 - Query Parameters: patientId
 - Description: Fetches notifications for upcoming appointments, medication reminders, and important health updates for a patient.

Navigation

PAGES

- **Sign In:** Entry point for both patients and healthcare providers to access their respective dashboards using credentials or facial recognition.
- **Dashboard:** A personalized landing page displaying an overview of recent activities, upcoming appointments, and quick access to various sections.

WIDGETS

- **Prescriptions:** View current and past prescriptions. Request refills.
- **Visit Summary:** Access detailed reports and notes from past visits.
- **Doctors:** Search for doctors by name. View doctor profiles and available appointment slots.
- **Reminders:** Set up and receive reminders for medication intake, upcoming tests, or other health-related activities.
- **Settings:** Manage account settings. Adjust privacy and notification preferences.

PROJECT REQUIREMENTS

DATABASE & API INTEGRATION

- **Database Interaction:** The app will interact with a centralized database managed through Prisma, which will store and retrieve data related to patients, doctors, appointments, medications, and medical conditions. This setup allows for efficient data management and ensures that all user interactions with the database are smooth and reliable.
- **API Integration:** The app integrates with the Microsoft Azure Face API, a service not covered in CodePath. This API will be used to implement facial recognition for secure and quick patient login. This integration not only enhances security but also improves user accessibility, particularly for patients who may have difficulty with traditional authentication methods.

USER AUTHENTICATION

- **Login/Logout Functionality:** Healthcare providers can log in and log out of the app.
- **Sign Up for New User Profiles:** Healthcare providers have the capability to create new user profiles for themselves. This feature is crucial for maintaining up-to-date records and allows for the expansion of the user base within the healthcare system.

VISUALS & INTERACTIONS

- **Multiple Views:** The app features several distinct views, a dashboard for doctors, detailed views for managing prescriptions and visit summary.
- **Interesting Cursor Interaction:** To enhance user engagement and provide a dynamic interface, custom cursor interactions such as tooltips will be implemented. For example, when users hover over actionable items like buttons or links, the cursor might change to a stylized pointer or a contextual icon, providing immediate visual feedback.
- **Complex Visual Styling:** A key highlight will be the sophisticated dashboard that serves as the central hub for navigation. This dashboard will display critical patient information and offer tabs for managing multiple patients simultaneously, facilitating seamless navigation. Additionally, users will be able to quickly access and preview visit notes, prescriptions, and historical patient records directly from the dashboard. Each of these elements will open in modals, allowing for convenient editing without navigating away from the main interface, thereby streamlining user interactions and improving workflow efficiency.
- **Loading State Visual Polish:** To improve the user experience during data loading or processing times, the app will implement visual elements such as glimmer effects and spinners. These elements keep the user informed about the ongoing processes and enhance the visual appeal of the app. Additionally, action buttons will be disabled during loading to prevent duplicate submissions and ensure data integrity.

TECHNICAL CHALLENGES

For your project, you should demonstrate that you can apply what you've learned so far and expand on that knowledge to write code and implement features that go beyond the scope of the projects you worked on during CodePath.

Based on the general idea and direction of your project requirements, your intern manager will create at least two (2) Technical Challenges for you. This section is all about explaining what they are and how you're planning to tackle them - you'll work together with your manager to fill it out.

Technical Challenge #1 - [Name/Small Description]

What

What problem are you solving, and what parts go beyond what you learned in CodePath?

How

Explain in words how you'll solve this problem.

You're encouraged to expand on this section with pseudo-code, links to external frameworks, architecture / design diagrams, anything that you can use to explain this to others!

Technical Challenge #2

What

How

DATABASE INTEGRATION

Using Prisma and pgAdmin together provides a comprehensive approach to database management for my application, where Prisma offers programmatic control and pgAdmin provides a graphical interface.

- Prisma will primarily be used within my application's development environment to handle database operations programmatically. It allows you to define my database schema in a human-readable format, perform migrations to keep my database schema in sync with my codebase, and interact with my database using Prisma Client. This is particularly useful for implementing and maintaining business logic, data manipulation, and server-side integrations in my application.
- pgAdmin serves as the visual interface to my PostgreSQL database. It is used for tasks such as visually exploring the database structure, manually tweaking data, running SQL queries, and monitoring database performance.

EXTERNAL APIS

I will create a Python script that uses the Amazon Rekognition API to verify user identity through facial recognition. This script will be designed to help me integrate facial recognition-based user verification into my website.

I will collect and store the face images of the people I want to be able to recognize, following the amazon guidelines for collecting and storing images. Then I will create a Rekognition collection to store the face images. And add the collected face images into the Rekognition collection. Now when a user attempts to log in or access a patient file, I will capture their face using the device camera. And will call the script to compare the captured image with the faces stored in the collection. The script will return a response indicating whether the captured images matches any of the face images in the collection. If a match is found, the response will include the matching face's data, such as user ID.

AUTHENTICATION

Handling User Authentication:

- **Login:** Users (healthcare providers) can log in using either traditional credentials (username and password)
- **Sign Up:** Healthcare providers can create accounts for themselves. This process involves collecting necessary information and verifying identities to ensure security.
- **Encryption:** As a stretch goal, implementing encryption for stored passwords and other sensitive data using algorithms.

Navigation Management:

- Authentication affects navigation by controlling access to various sections of the app. After logging in, users are directed to their respective dashboards based on their role (patient or healthcare provider).

VISUALS AND INTERACTIONS

Interesting Cursor Interaction:

- Implement custom cursor designs or effects when hovering over interactive elements. For example, changing the cursor to a stylized hand when hovering over buttons or links.

Tooltip:

- Provide tooltips on icons and buttons to enhance user understanding and accessibility. Tooltips appear when the user hovers over an element, offering additional information about its function.

UI Component with Custom Visual Styling:

- Design custom-styled components such as buttons, input fields, and toggles that align with the app's theme. Use CSS and JavaScript to create unique visual and interactive experiences. A key highlight will be the sophisticated dashboard that serves as the central hub for navigation. This dashboard will display critical patient information and offer tabs for managing multiple patients simultaneously, facilitating seamless navigation. Additionally, users will be able to quickly access and preview visit notes, prescriptions, and historical patient records directly from the dashboard. Each of these elements will open in modals, allowing for convenient editing without navigating away from the main interface, thereby streamlining user interactions and improving workflow efficiency.

Expand/Collapse Dashboard Layout:

- Implement an expandable and collapsible dashboard layout to allow users to customize their view. This can be achieved using JavaScript to dynamically adjust the CSS properties of dashboard panels.

Loading State:

- Glimmer Effect: Implement a shimmer effect as a placeholder for content as it loads, enhancing the perceived performance of the app.
- Spinners: Display spinners during create/edit operations to indicate that a process is ongoing. This helps manage user expectations and prevents multiple submissions.
- Disabling Action Buttons: During loading states, action buttons are disabled to prevent duplicate operations. This is managed by toggling the disabled attribute of buttons based on the loading state.

TIMELINE

MU Week	Project Week	Focus	User Stories
4	1	Create the coding environment, sign in requirements and encryption.	<p>Code Environment:</p> <ul style="list-style-type: none">• Set up backend and frontend folders and packages <p>Frontend:</p> <ul style="list-style-type: none">• Create wireframes for the sign in page• Route the website to start on sign in page• Send sign in information as a post to the backend <p>Backend:</p>

			<ul style="list-style-type: none"> • Create database/models for the sign in • Create server endpoints for the sign in • Use bcrypt to encrypt passwords <p>Main goals:</p> <ul style="list-style-type: none"> • Healthcare provider can sign in (normal password and username)
5	2	User authentication, dashboard, doctor add visit summary and prescriptions.	<p>Continue:</p> <ul style="list-style-type: none"> • Password encryption <p>Frontend:</p> <ul style="list-style-type: none"> • Create dashboard and components <ul style="list-style-type: none"> ◦ Individual components and modals <ul style="list-style-type: none"> ■ Search bar • Fetch data from the database and send it to the database. <p>Backend:</p> <ul style="list-style-type: none"> • Create database/model for doctors and patients • Create database/model for visit notes, prescriptions • Create server endpoints (CRUD).
6	3	Technical Challenge and Stretch features	<p>Frontend:</p> <ul style="list-style-type: none"> • Start implementing the facial recognition with the API • Create page and components for the check-in queue that updates real time • Patient facial sign in components and API fetch • Message doctor that patient has signed in (potentially open patient file). <p>Backend:</p> <ul style="list-style-type: none"> • Create database/models for the queue • Look patient up with facial recognition • Set up server end points • Message doctor patient has checked in
7	4	Technical Challenge and Stretch features	<p>Frontend:</p> <ul style="list-style-type: none"> • Create a new page where patients can message doctors • Route that page in dashboard page • Return message to doctor <p>Backend:</p> <ul style="list-style-type: none"> • Create database/models for the messages • Return message to doctor
8	5	Technical Challenge and Stretch features	<p>Frontend:</p> <ul style="list-style-type: none"> • Create print component in dashboard

			<ul style="list-style-type: none"> • Create pdf print preview with prescriptions and doctors name • Create new page for patient notification settings • Fetch patients notifications and decide the message method and time to send prescriptions and visit summary <p>Backend:</p> <ul style="list-style-type: none"> • Fetch prescriptions • Create database/models for the patient notifications
9	6	Technical Challenge and Stretch features Presentation	<p>Continue:</p> <ul style="list-style-type: none"> • Send notifications based on user preferences via email, message, etc. <p>Stretch:</p> <ul style="list-style-type: none"> • Look into data encryption feasibility <p>Presentation:</p> <ul style="list-style-type: none"> • Presentation • Create a doctor account and patient account for demo • Render the website
10	7	Stretch features.	<p>Stretch:</p> <ul style="list-style-type: none"> • TBD: Family members can view other patient files