Android App Development-2

Health +

# 1. Introduction and Motivation:

The "Doctor's Appointment App" is an Android application designed to streamline the process of scheduling doctor's appointments for users in their localities.

With busy lifestyles and limited time, visiting doctors can become a cumbersome task. This app aims to bridge the gap by providing a platform where users can instantly schedule appointments with nearby doctors, ensuring timely medical care and convenience.

# 2. Problem Statement:

The traditional process of scheduling doctor's appointments often involves phone calls, long waiting times, and uncertain availability. This can lead to delays in receiving medical attention and inconvenience for patients.

The "Health +" seeks to solve this problem by allowing users to effortlessly book appointments with nearby doctors, eliminating the need for physical visits or prolonged waiting times.

# 3. Methodology:

The project involved several key steps:

* Research: Understanding the existing healthcare appointment systems and analyzing user preferences.
* Design: Creating a user-friendly interface for easy navigation and appointment scheduling.
* Development: Implementing the app's frontend and backend, integrating user authentication, doctor availability, and appointment scheduling logic.
* Testing: Rigorous testing to ensure a smooth user experience, robust functionality, and accurate data handling.
* Deployment: Releasing the app on the Google Play Store for users to download and use.

# 4. Code Implementation:

Class: DatabaseHelper

onCreate(SQLiteDatabase db):

Purpose: Called when the database is created for the first time.

Explanation: Executes SQL statements to create the required database tables (users and appointments).

onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion):

Purpose: Called when the database needs to be upgraded, typically when the DATABASE\_VERSION changes.

Explanation: Drops the existing tables (users and appointments) and recreates them with updated structure.

insertUser(String username, String email, String password):

Purpose: Inserts user details into the users table.

Explanation: Inserts provided username, email, and password into the users table and returns the inserted row's ID.

checkUser(String username):

Purpose: Checks if a user with the given username exists.

Explanation: Queries the users table to find if there's a user with the provided username and returns a boolean indicating existence.

getUserDetails(String username):

Purpose: Retrieves details of a user by username.

Explanation: Queries the users table for details of a user with the provided username and returns a User object with user ID, username, and email.

insertAppointment(String name, String email, long appointmentDate, String medicalIssue):

Purpose: Inserts appointment details into the appointments table.

Explanation: Inserts provided appointment details (name, email, appointmentDate, medicalIssue) into the appointments table and returns the inserted row's ID.

getAllAppointments():

Purpose: Fetches all appointments from the appointments table.

Explanation: Queries the appointments table and constructs a list of Appointment objects with details like name, email, appointmentDate, and medicalIssue. Returns the list

[11:23 p.m., 2023-08-14] Sanath Tuli: Class: Apointment Activity

onCreate(Bundle savedInstanceState):

Purpose: Entry point when the activity is created.

Explanation: Sets up the activity's layout, initializes UI elements, and sets a click listener for the submit button.

btnSubmit.setOnClickListener(new View.OnClickListener()):

Purpose: Handles the click event of the submit button.

Explanation: When the submit button is clicked, it retrieves input data (name, email, medicalIssue) and the selected date from the date picker. Then, it creates a Calendar object to convert the selected date into epoch time (milliseconds). The appointment details are then inserted into the database using the insertAppointment method of the DatabaseHelper class. A toast message is displayed to inform the user whether the insertion was successful or not.

Calendar calendar = Calendar.getInstance();:

Purpose: Create a Calendar object with the current date and time.

Explanation: This is used to convert the selected date from the date picker into epoch time (milliseconds). It provides a way to manipulate and retrieve date and time components.

long appointmentDate = calendar.getTimeInMillis();:

Purpose: Get the selected appointment date in milliseconds (epoch time).

Explanation: Converts the selected date (year, month, day) into epoch time, which is a long value representing the number of milliseconds since January 1, 1970.

long rowId = databaseHelper.insertAppointment(name, email, appointmentDate, medicalIssue);:

Purpose: Insert appointment details into the database.

Explanation: Calls the insertAppointment method from the DatabaseHelper class to insert the gathered appointment details (name, email, appointment date, medical issue) into the database's appointments table. It returns the ID of the newly inserted row.

if (rowId != -1):

Purpose: Check if the insertion was successful.

Explanation: If the returned rowId is not equal to -1, it indicates that the insertion was successful. In such a case, a success toast message is shown and the activity is finished (closed). If the rowId is -1, it implies that the insertion failed, and an error toast message is shown.

[11:24 p.m., 2023-08-14] Sanath Tuli: Class: Apointments Adapter

onCreateViewHolder(ViewGroup parent, int viewType):

Purpose: Creates the view holder for a single item in the RecyclerView.

Explanation: Inflates the layout for a single item (item\_appointment.xml) and returns a new instance of AppointmentViewHolder that holds the references to the TextViews.

onBindViewHolder(AppointmentViewHolder holder, int position):

Purpose: Binds data to the views within a view holder.

Explanation: Retrieves the Appointment object from the list based on the current position. Sets the data of the appointment (name, email, medical issue, and formatted appointment date) to the appropriate TextViews within the view holder.

getItemCount():

Purpose: Returns the number of items in the data list.

Explanation: Spec…

Class: Doctor List Adapter

getItemCount():

Purpose: Returns the number of doctor items in the list.

Explanation: Specifies the total number of doctor items to be displayed in the RecyclerView, which is determined by the size of the doctorList.

DoctorViewHolder inner class:

Purpose: Holds references to the views within a single doctor item view.

Explanation: Defines the structure of a single doctor item's layout (item\_doctor.xml) and initializes the ImageView (ivDoctorImage), TextViews (tvDoctorName and tvSpecialization), and Button (btnBookAppointment) to represent various doctor details.

Class: Doctor List Adapter

getItemCount():

Purpose: Returns the number of doctor items in the list.

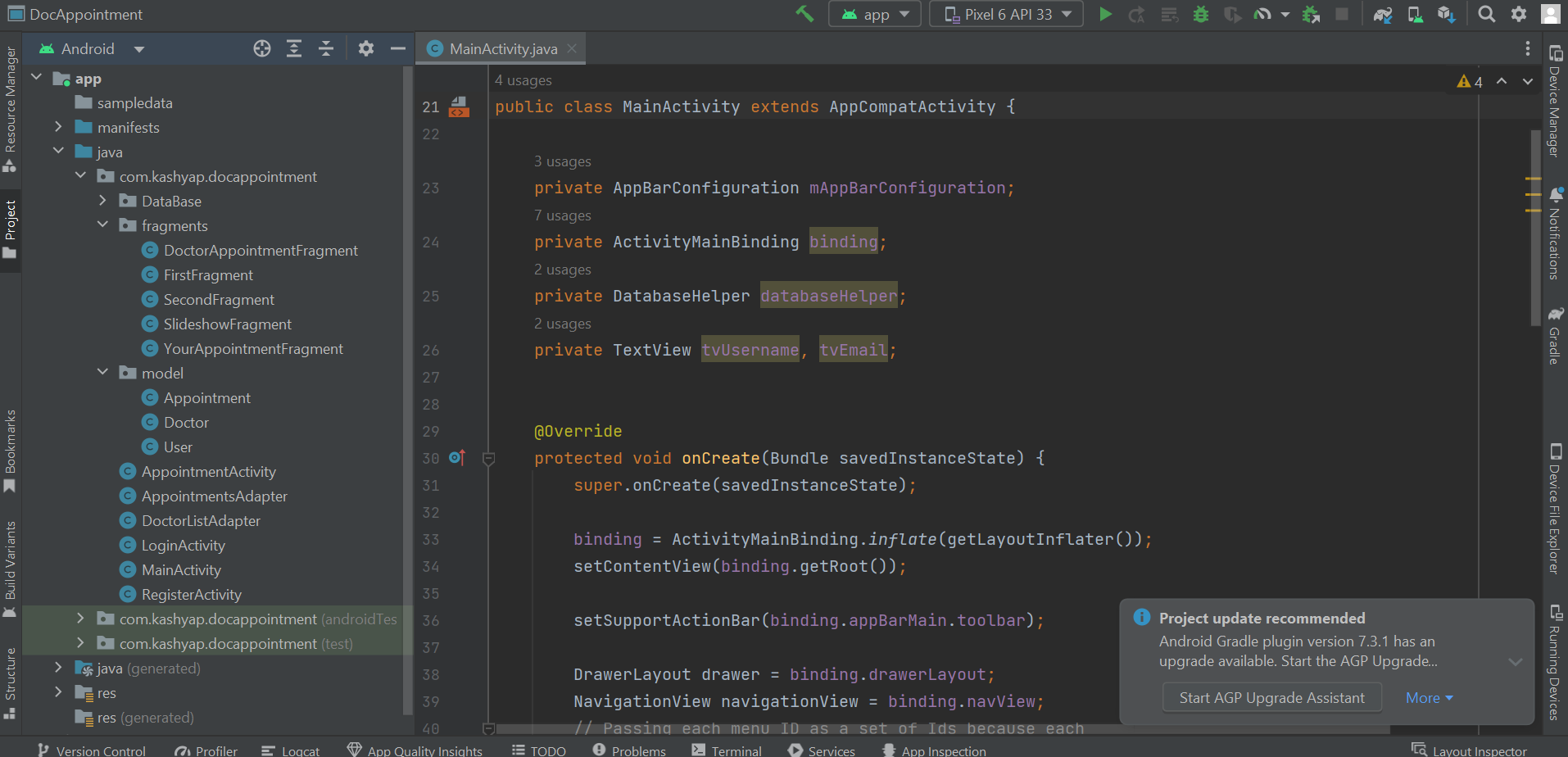
Explanation: Specifies the total number of doctor items to be displayed in the RecyclerView, which is determined by the size of the doctorList.

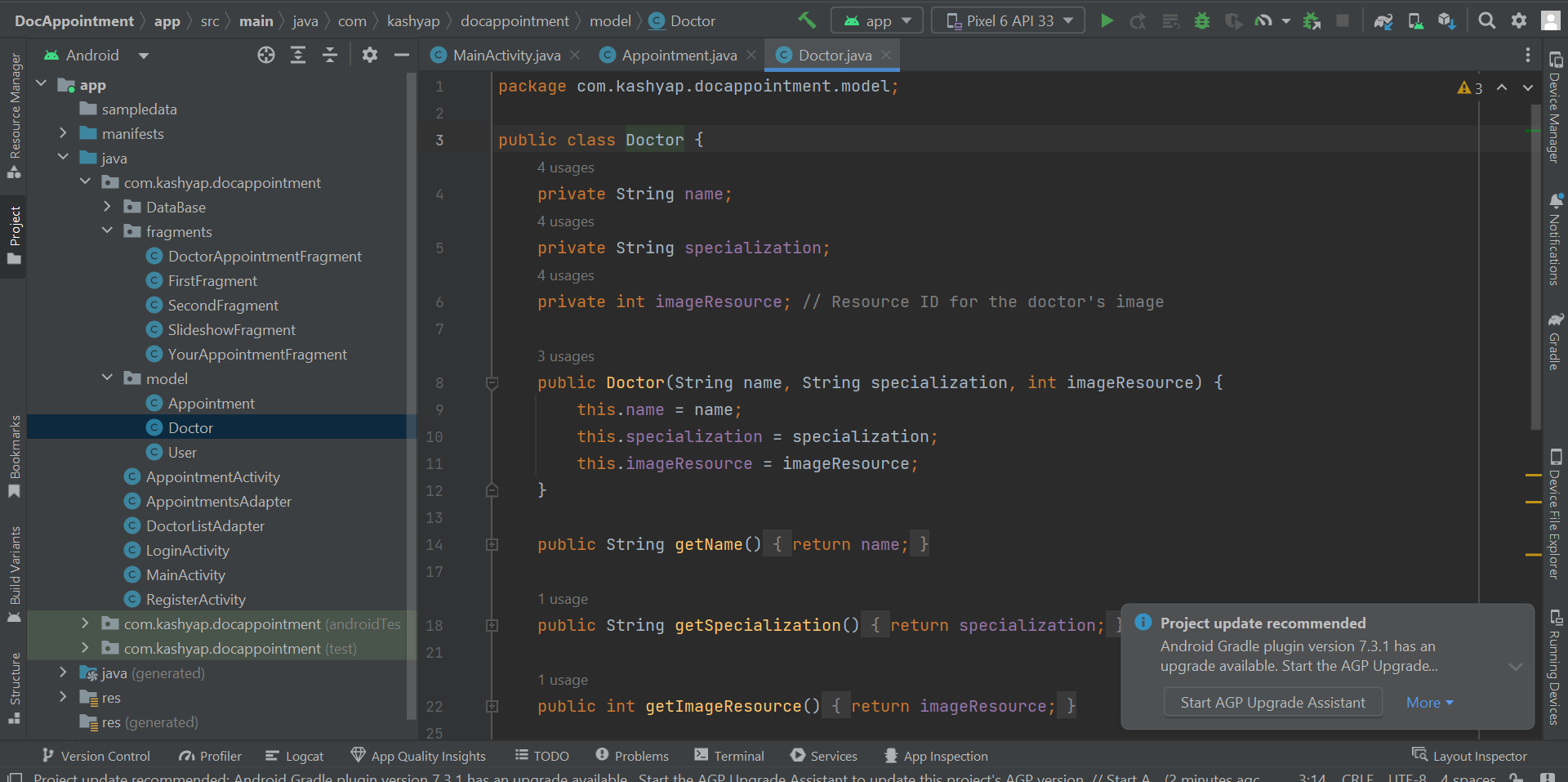
DoctorViewHolder inner class:

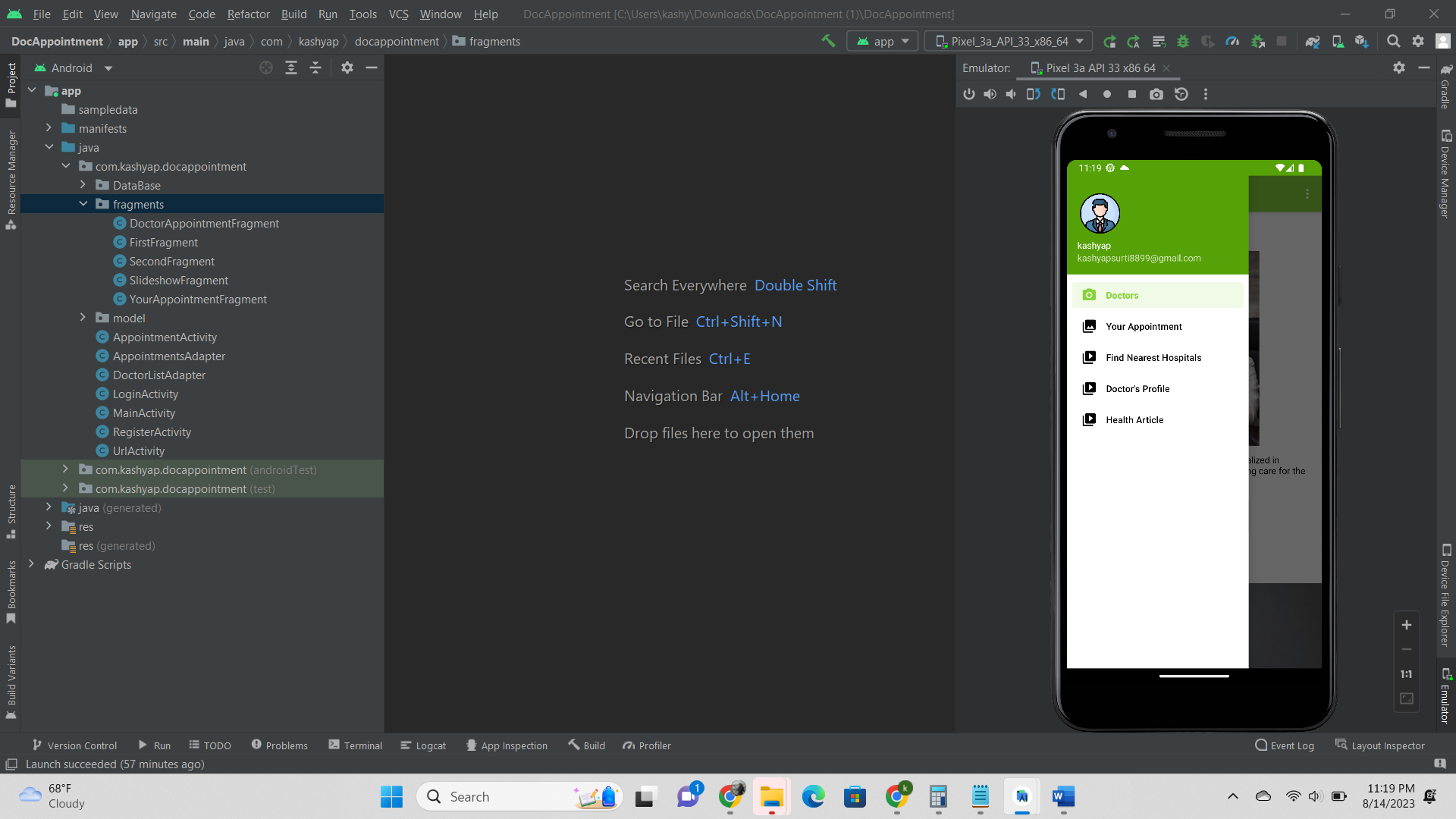
Purpose: Holds references to the views within a single doctor item view.

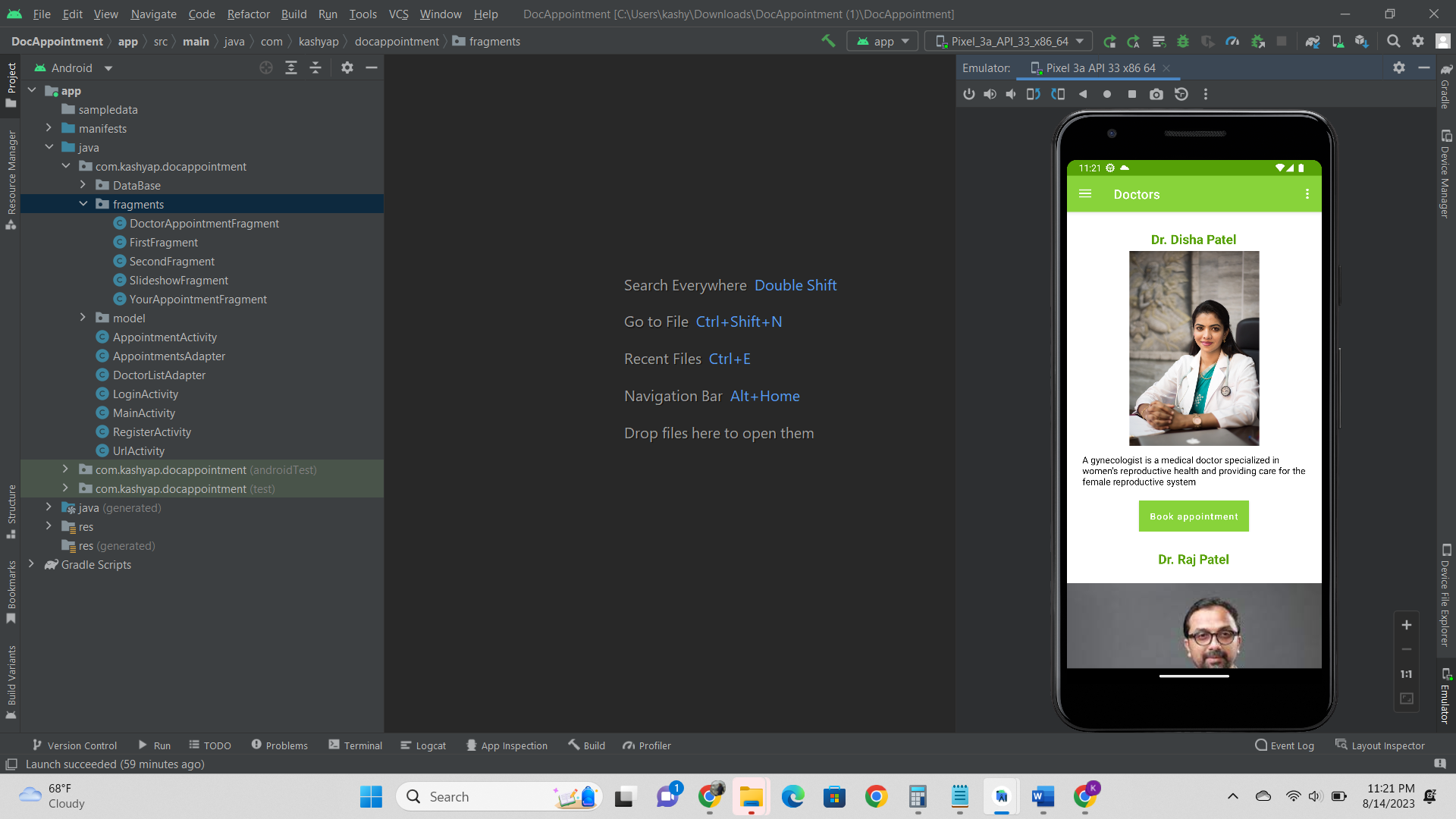
Explanation: Defines the structure of a single doctor item's layout (item\_doctor.xml) and initializes the ImageView (ivDoctorImage), TextViews (tvDoctorName and tvSpecialization), and Button (btnBookAppointment) to represent various doctor details.

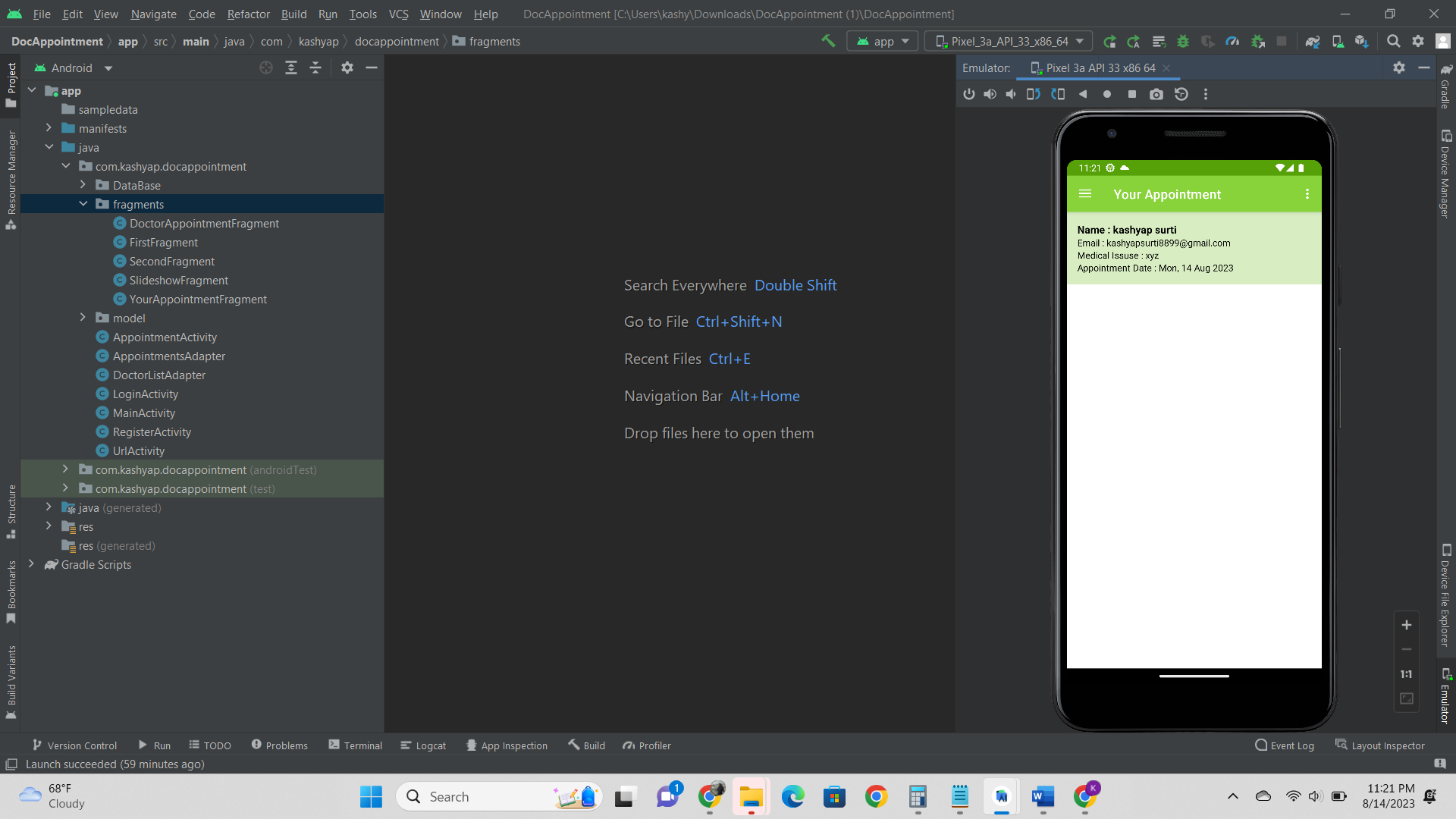
# 5. Snippet of the Application:

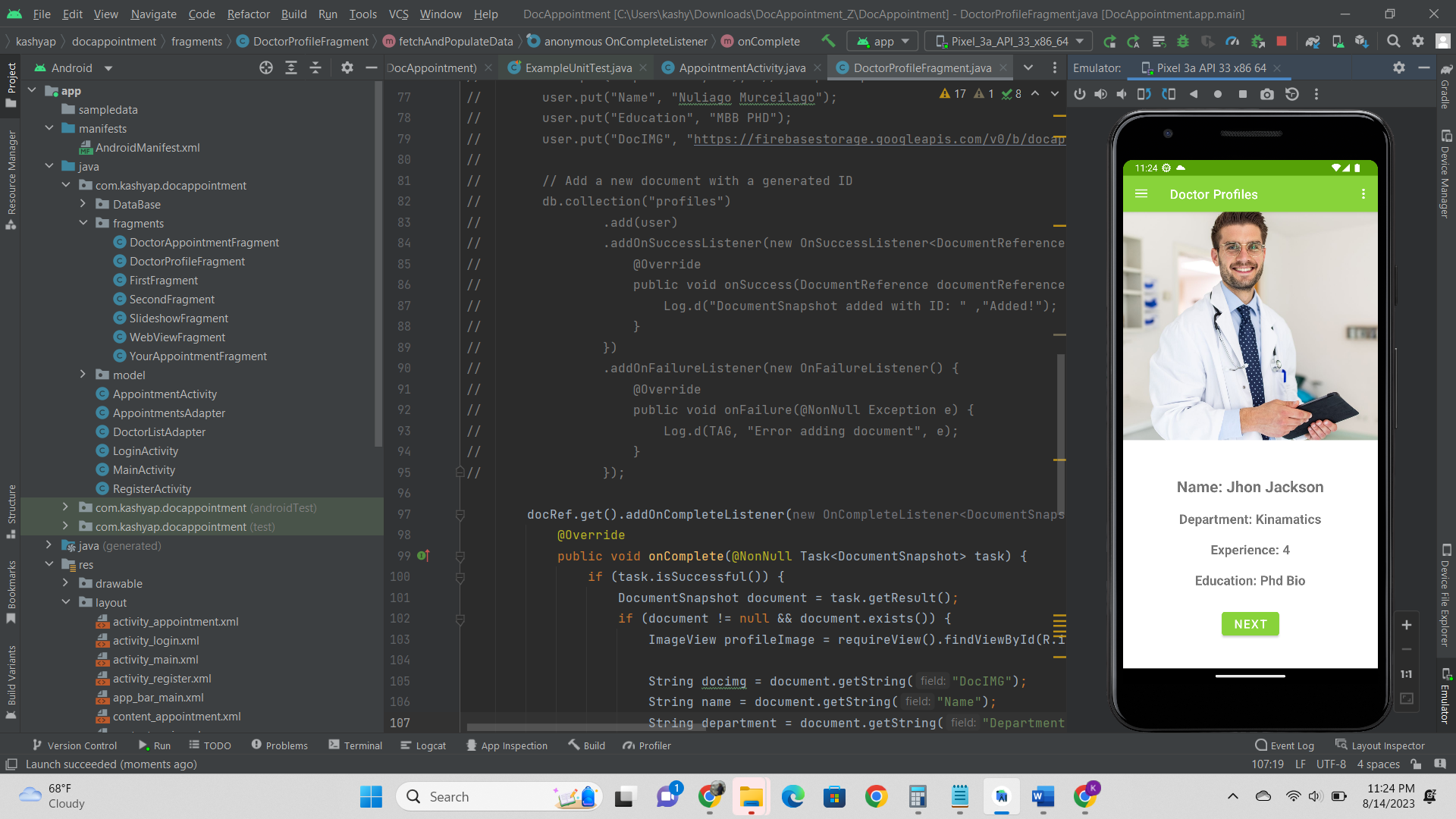










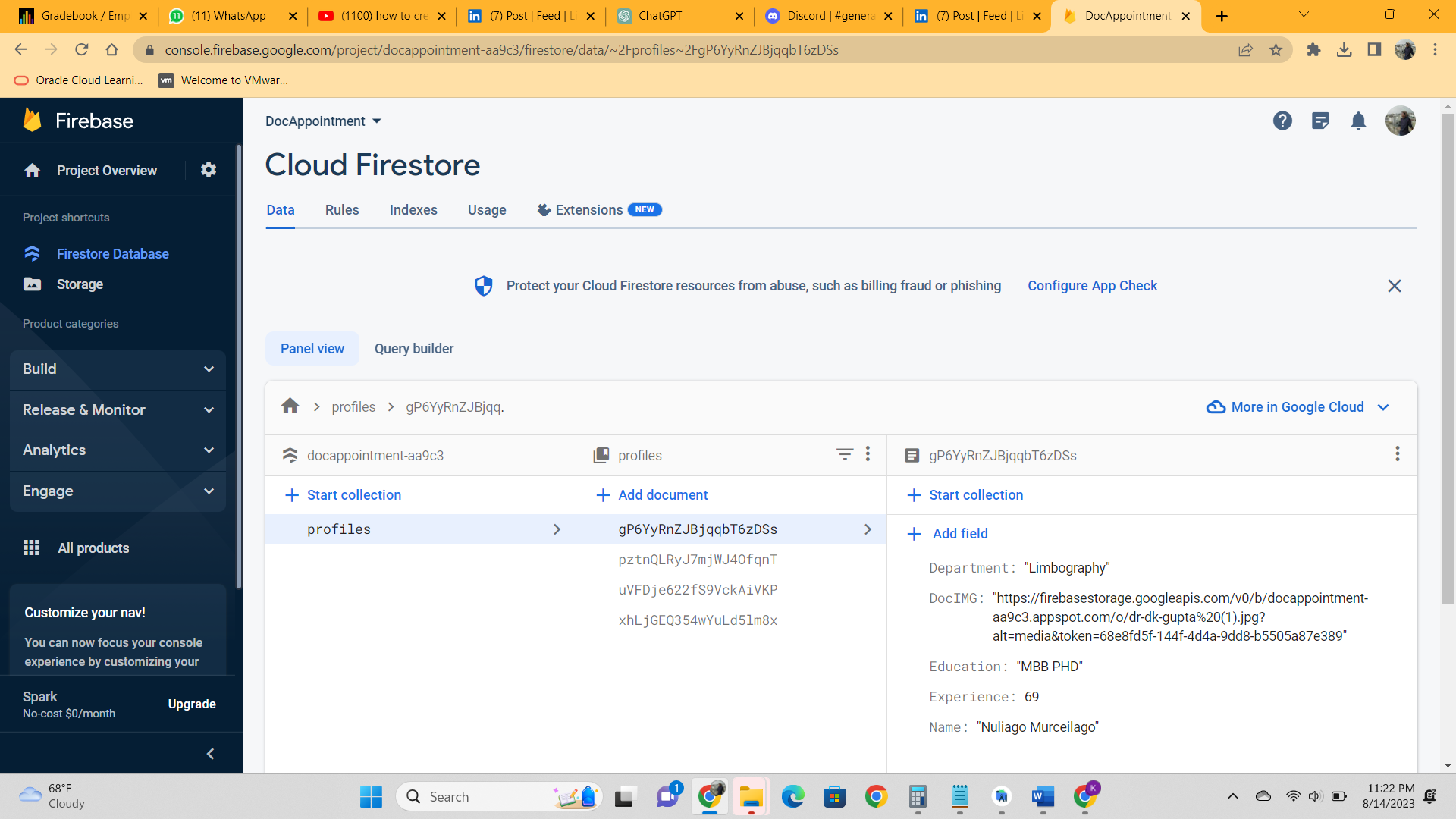


A cell phone with a map

Description automatically generated

A screenshot of a cell phone

Description automatically generated



# 6. Conclusion:

The "Doctor's Appointment App" is a significant step toward making healthcare more accessible and convenient.

By providing users with the ability to schedule appointments with nearby doctors instantly, the app addresses the common challenges of traditional appointment systems.

The project successfully integrates user authentication, doctor availability tracking, and appointment scheduling, culminating in an efficient and user-friendly application. As healthcare continues to evolve, such innovative solutions play a crucial role in enhancing the overall experience for both patients and healthcare providers.