



ORTA DOĞU TEKNİK ÜNİVERSİTESİ  
MIDDLE EAST TECHNICAL UNIVERSITY

**CNG 495 FALL -2024**

**TERM PROJECT  
PROGRESS REPORT I**

**SAFEBOOK**

**MEMBERS**

**Wendy Susan Ondola – 2413490**

**Aaron Bandado - 2476984**

## CLOUD BASED SERVICE BOOKING MANAGEMENT SYSTEM : SAFEBOOK

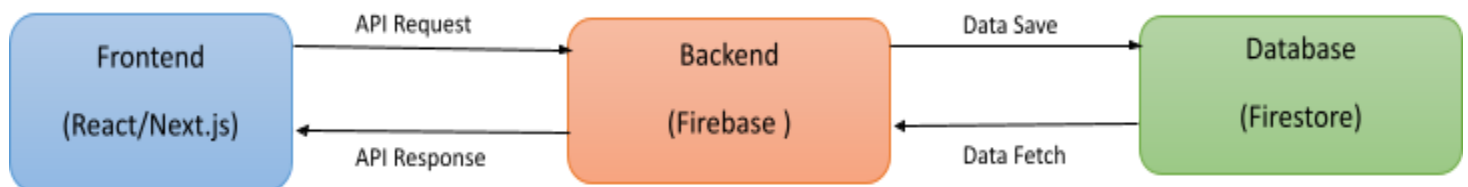
SafeBook is an all-inclusive cloud-based booking platform that enables small businesses to easily manage their appointments and schedules. Its main purpose is to enable small businesses to allow their customers to create appointments based on their availability on a publicly shared schedule and equally manage them seamlessly and efficiently. For example, a barber has many clients and would like to manage all their appointments. This software would allow the barber to specify the time slots when they are available and the services they offer along with any other vital information and their customers would be able to book an appointment with them.

Key features will include user authentication, role-based access (either the businesses or the clients), calendar integration (for efficient booking functionality), and automated reminders to reduce no-shows.

There are three different cloud delivery models in cloud computing namely Platform As A Service(PAAS), Software As A Service(SAAS), and Infrastructure As A Service (IAAS). In this project, we will implement Software As A Service (SAAS) as the cloud delivery model and later deploy our web application on Vercel.

## IMPLEMENTATION

We plan on implementing a full-stack application that supports seamless working between the frontend, backend, and database systems. For example, an action performed by the user on the front end (UI) sends requests to the backend which processes the information and then interacts with the database to either store/retrieve data, and a response is reflected on the front end.



## Frontend (React/Next.js)

This is the part where clients/users directly interact and it holds all the visuals required by the users to interact with the system. For the implementation of this part, we will use React within the Next.js framework. This provides us with a wide range of advantages including but not limited to faster page load, file-based routing, the ability to create APIs within the same project, easy deployment, etc.

## Backend (Next.js and Firebase)

Using the NextJS react framework will enable us to have routing for our web application using NextJS page routing. Also, for the backend, we will be using Firebase (BAAS-Backend As a Service) to enable us to handle our requests. The code for the development will be written using Javascript/Typescript.

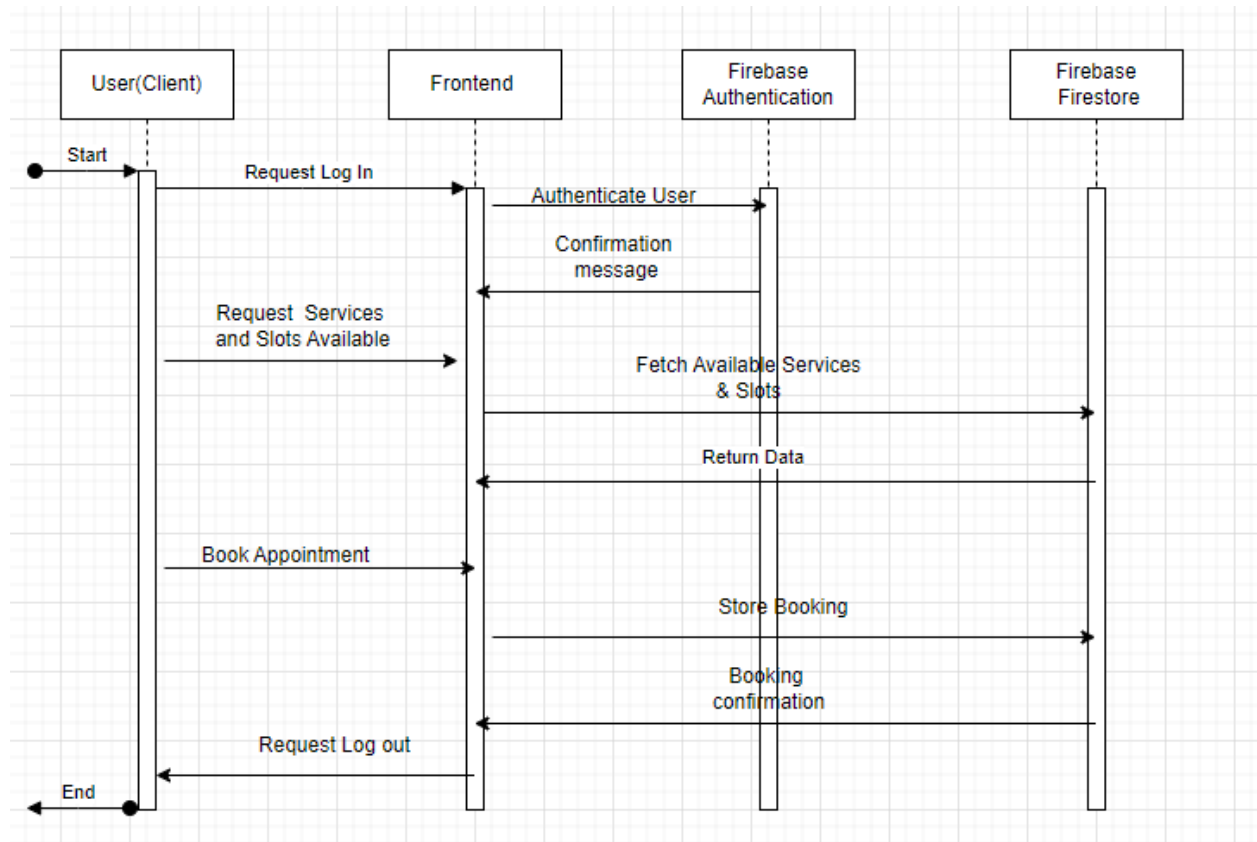
Firebase is a comprehensive app development platform owned by Google, designed to help developers build, improve, and grow applications across various platforms. Firebase is deeply integrated with Google Cloud Platform (GCP), enabling it to leverage Google's scalable infrastructure and powerful cloud services.

## Database (Firebase/External Database)

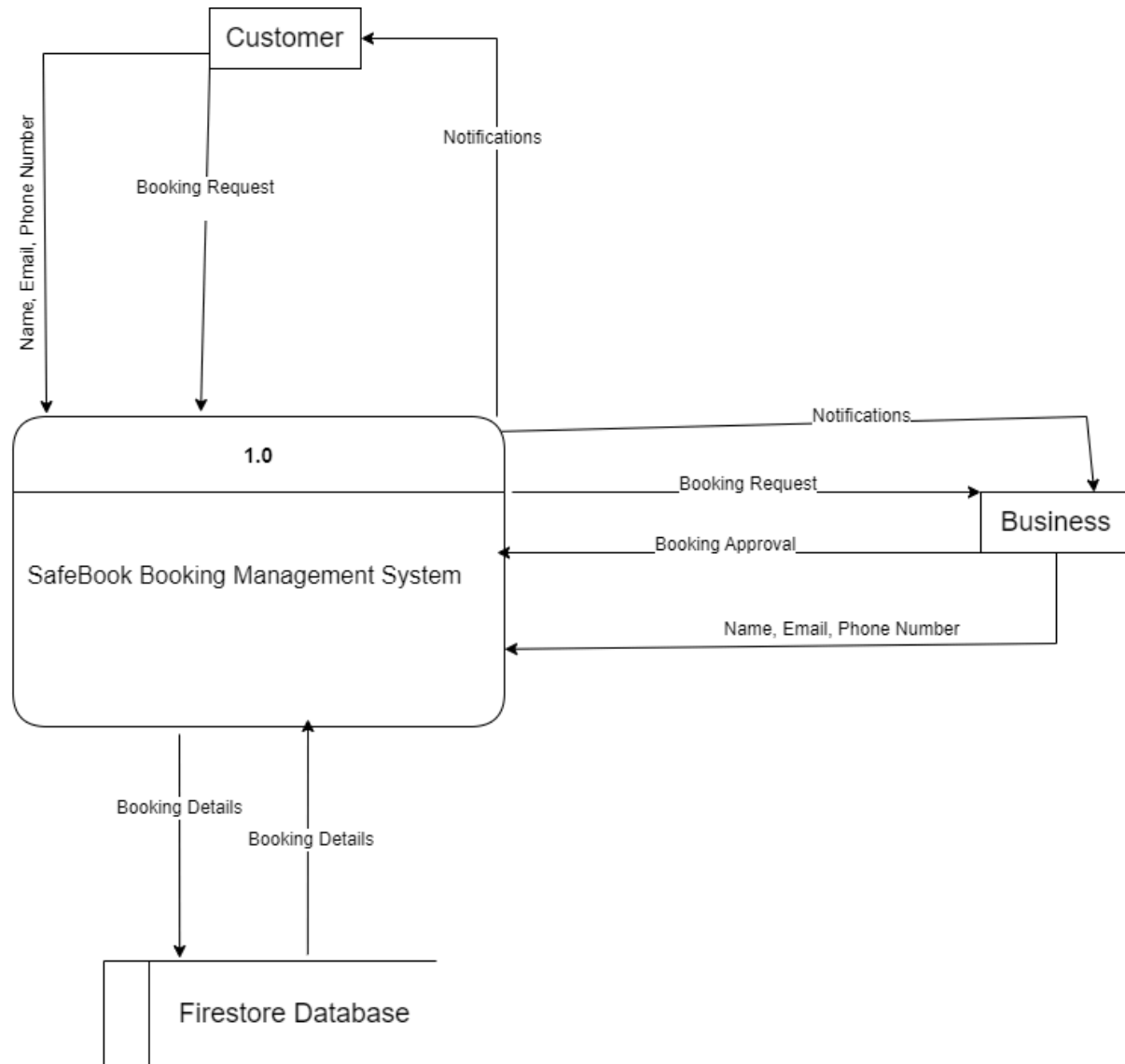
For data handling (storage and retrieval), we will use Firestore, a NoSQL database in the Firebase ecosystem. Communication with Firestore will be managed through Firebase's SDK, allowing for seamless real-time data handling. We may also use Firebase Cloud Functions for additional server-side logic as needed.

## DIAGRAMS

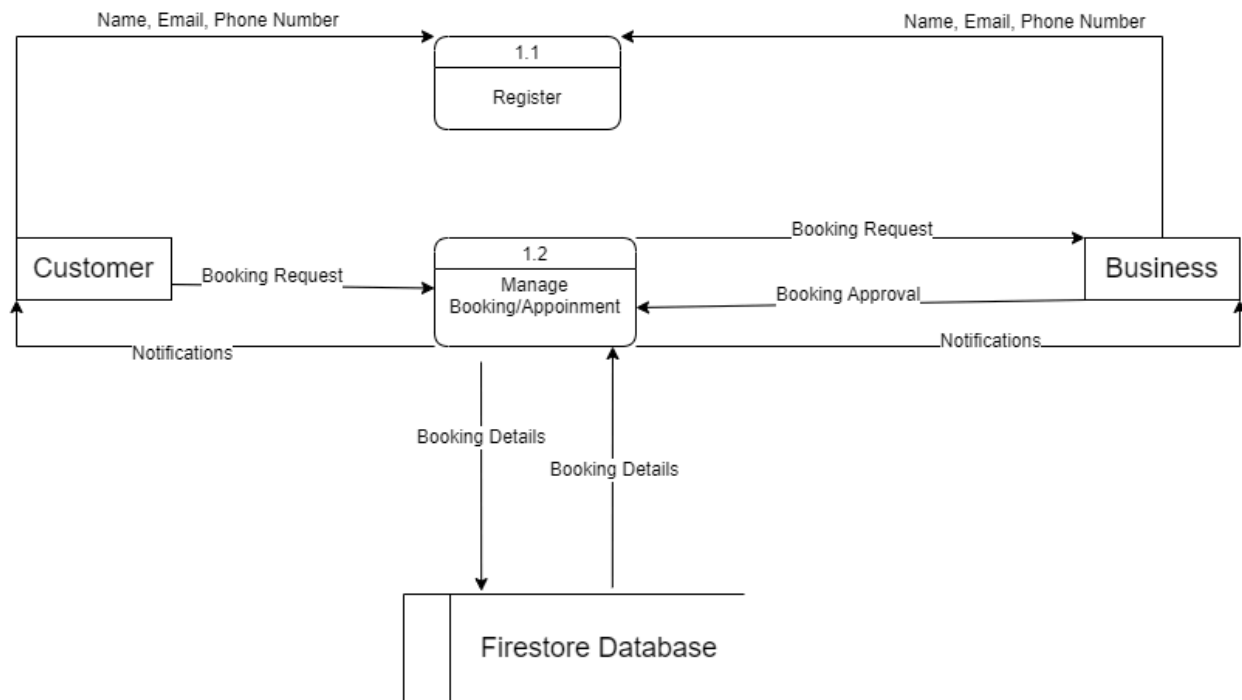
### Client Service Interaction Diagram



## Data Flow Diagrams



Level-0 Dataflow Diagram



Level-1 Dataflow Diagram

## DATA TYPES

- Text - The names, emails and passwords will be stored in text format(String)
- Numbers - IDs, prices, duration, service amount will be stored in form of numbers
- Date-time/Timestamps - Since its a booking platform we have to store the appointment dates in date-time format
- Metadata

## COMPUTATION

- Timeslot calculations/ Overlap checking: Show customers available time slots for a business and ensure they can only book free time slots.
- Notifications: Notify users of successful bookings and businesses of booking requests.
- Authentication: Ensure user accounts are secured, and users can only login to their own accounts.

## EXPECTED CONTRIBUTION

Frontend Development - Aaron

Backend Development - Both members

Database Design - Wendy

Cloud Setup - Both members

## MILESTONES

### Week 7

Setting up and firebase basics

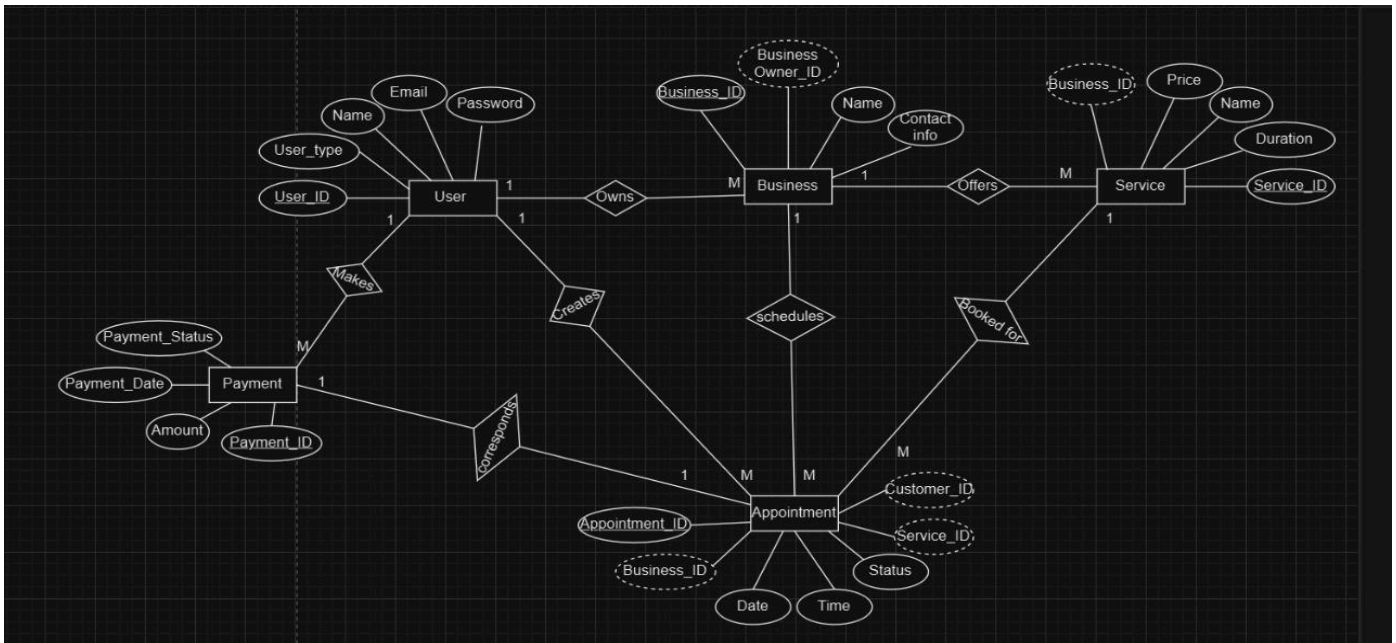
- We familiarized ourselves with the firebase and set up the project infrastructure
- Firebase set-up, learning basics and environment set-up
- GitHub set up and getting familiarized with it.

### Week 8

- Login page design - Aaron Bandado
- Landing page design - Aaron Bandado
- Firestore database Setup - Wendy

## Week 9

- Database design - Wendy Ondola



- Above is a skeleton design /schema of the database structure we would like to have. It shows the attributes that each document is expected to have and the relationships between these entities for example a business can schedule multiple appointments and a user can own a business(business owner ) or create an appointment (Customer)
- In the schema we have a collection of documents including users, business,appointment, service and payment.

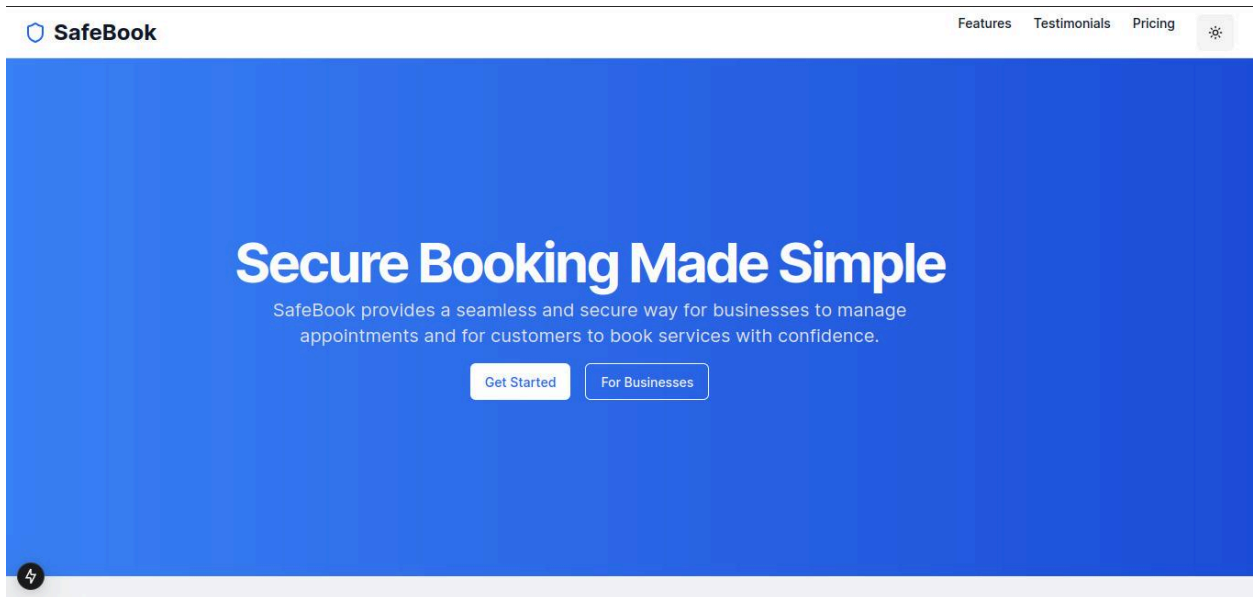
## Week 10

- Implemented the UI functionalities eg, login
- Set up/implemented the database in Firestore and did the configuration.
- Synchronized the UI with the database in firestore i.e when user enters their login in information it is stored and reflected on the database in Firestore
- Report writing and compiling the final results.



## SCREENSHOT OF USER INTERFACE


The below image shows the landing page of the SafeBook application, designed to give information to visitors about the application.



Safe Book Landing Page

The Image below shows the signup page for customers of the SafeBook application.

---



## Create your SafeBook account

Sign up to start booking appointments with ease

Full Name


Email address

Password

Confirm Password

Sign up

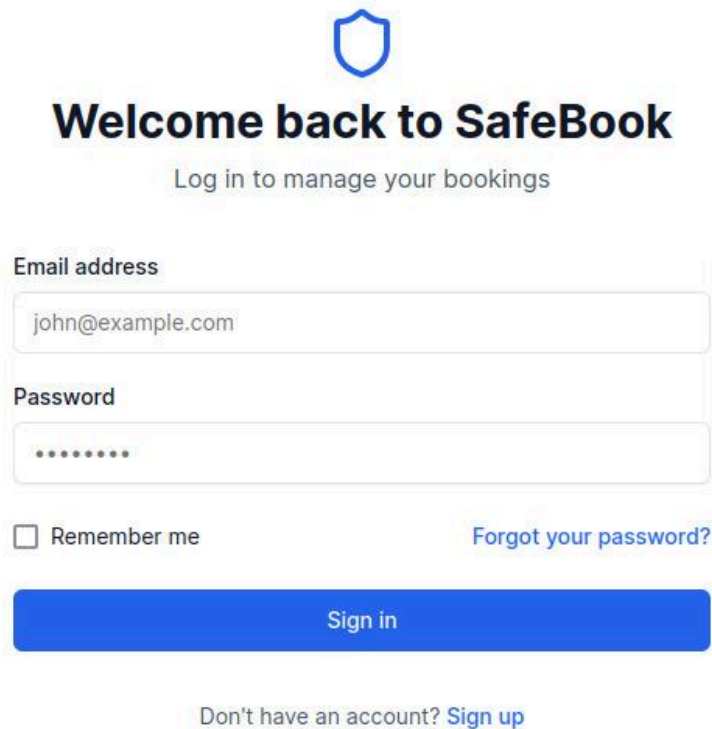
Already have an account? [Log in](#)




SafeBook User Signup Page

The below image shows the login page of the SafeBook application designed to allow users to login to the application.

---



The image shows the login page of the SafeBook application. At the top center is a blue shield icon. Below it, the text "Welcome back to SafeBook" is displayed in a bold, black font. Underneath this, in a smaller, lighter font, is the text "Log in to manage your bookings". The login form consists of two input fields: "Email address" and "Password". The "Email address" field contains the text "john@example.com". The "Password" field is masked with dots. Below the password field, there is a checkbox labeled "Remember me" and a link "Forgot your password?". At the bottom of the form is a large blue button labeled "Sign in". Below the button, there is a link "Don't have an account? Sign up".



## Welcome back to SafeBook

Log in to manage your bookings

Email address

john@example.com

Password

.....

☐ Remember me [Forgot your password?](#)

[Sign in](#)

Don't have an account? [Sign up](#)

SafeBook Login Page

The image below shows the user interface for the appointment bookings table on the business dashboard. It shows the businesses the bookings made by the customers and the service they selected.

ScheduleBusiness InfoAppointments

Appointment Bookings

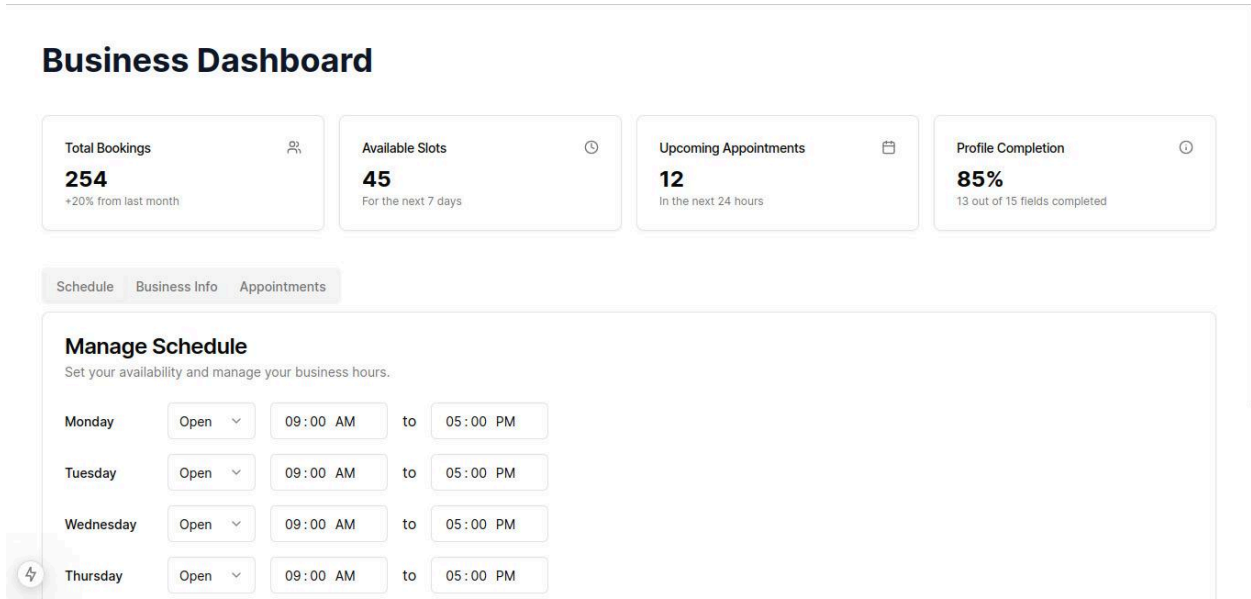
View and manage your upcoming appointments.

Pick a date

Customer	Service	Date & Time
John Doe	Haircut	June 15th, 2023 10:00 AM
Jane Smith	Manicure	June 15th, 2023 11:30 AM
Bob Johnson	Massage	June 16th, 2023 2:00 PM


SafeBook Appointment Bookings Table

The image below shows an image of the dashboard for the business, which contains some metrics about the businesses bookings and appointments.



The image below shows the signup page for the Business accounts. It is separate from the first signup page, because it requires more information from the businesses for them to be able to use the application.

---



## Create your Business Account

Sign up to start managing appointments with SafeBook

Business Name

Acme Inc.

Business Type

e.g., Salon, Dental Clinic, Consulting

Contact Person

John Doe

Business Email

contact@acmeinc.com

Business Phone

+1 (555) 123-4567

Password

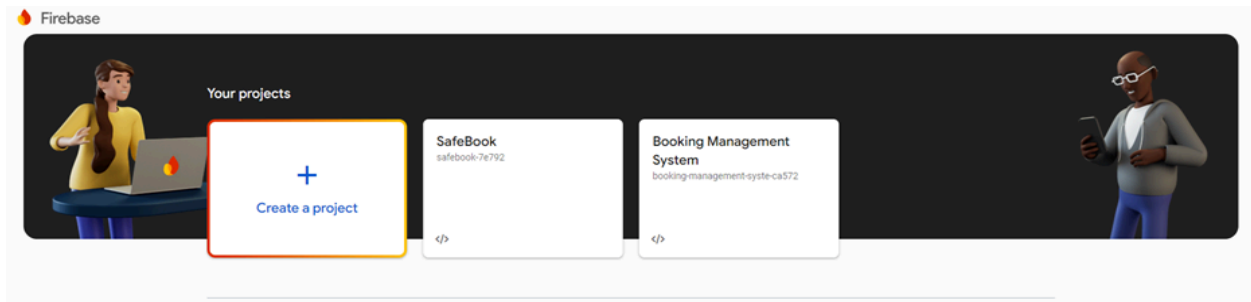
••••••••

SafeBook Business Signup Page

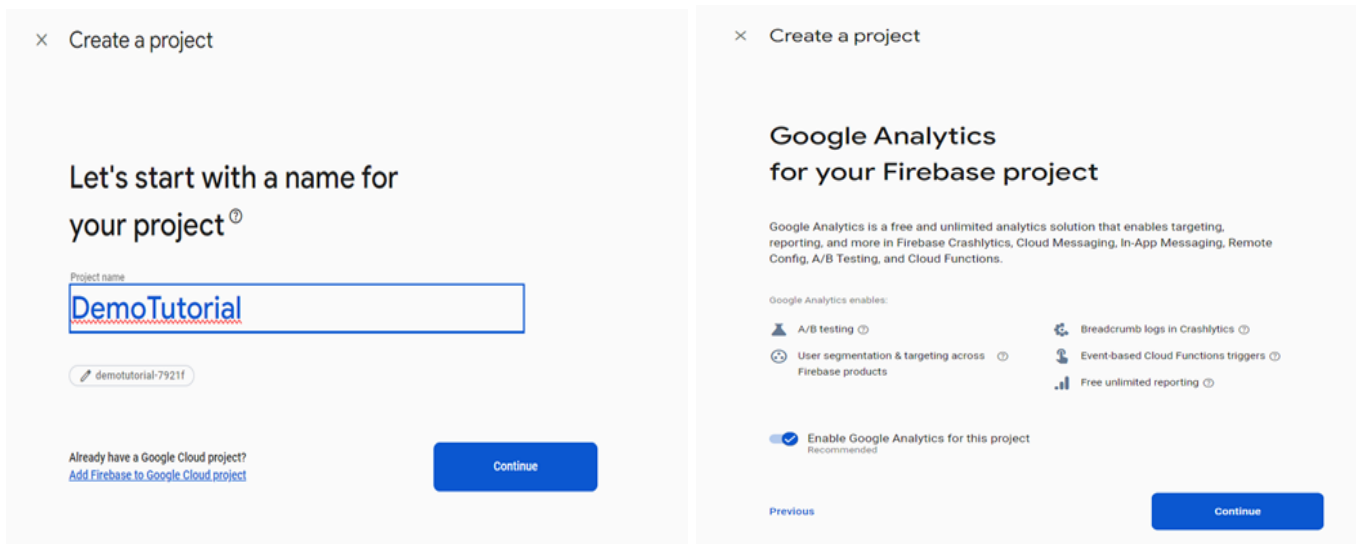
# TUTORIAL FOR CLOUD TECHNOLOGY USED

## 1. Setting Up a Firebase Project

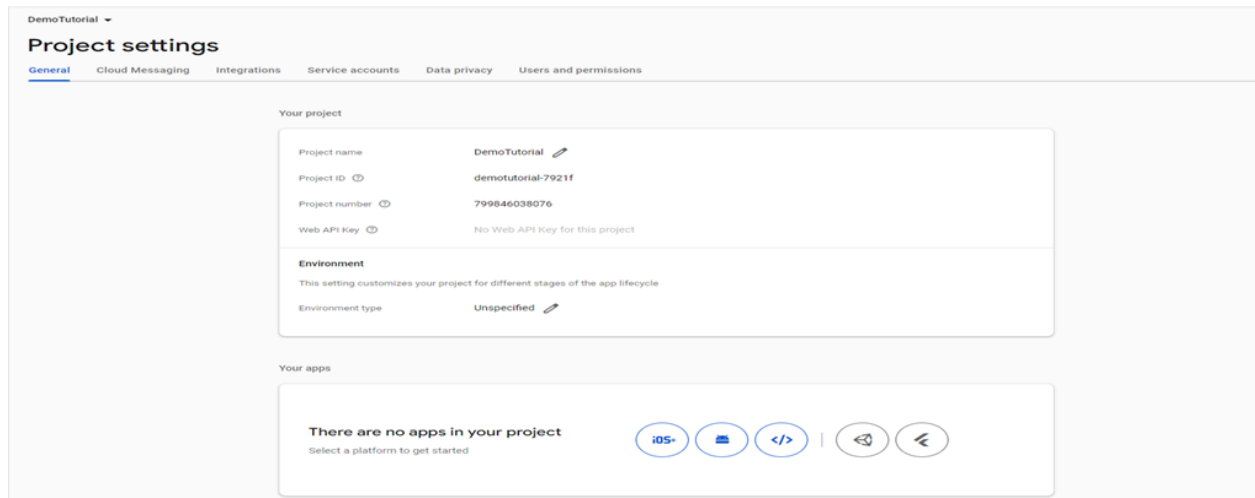
- Go to the Firebase Console.
- Click "**Create Project**" and follow the steps to name your project.



- Name your project and enable **Google Analytics** (optional).



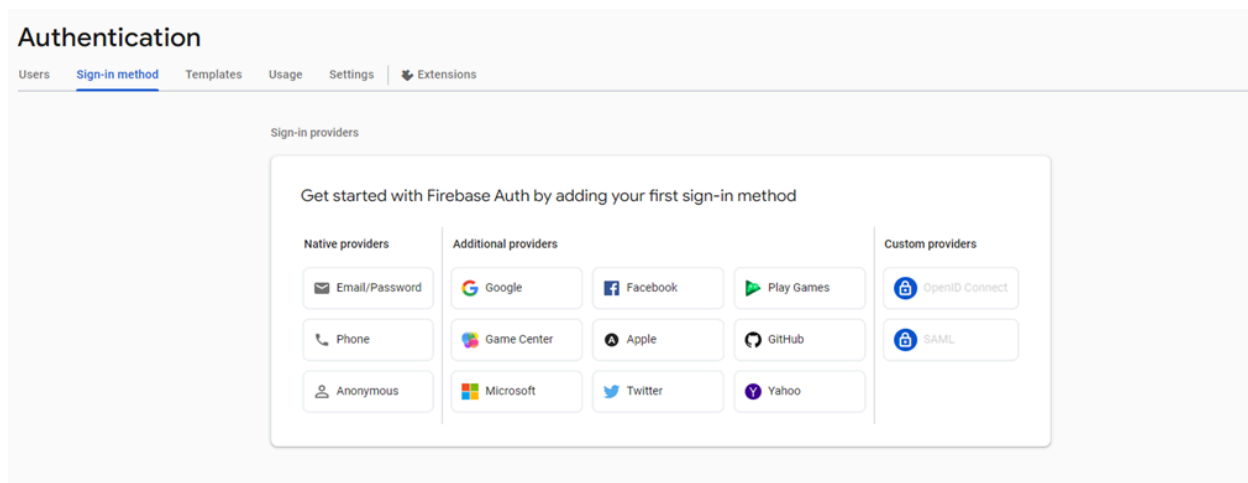
- Once the project is created, navigate to the **Project Settings** to find your project credential



## 2. Firebase Authentication

- In Firebase Console:

Go to **Build > Authentication > Sign-in Methods**.



- Enable **Email/Password** and **Google** sign-in methods.



## Authentication

Users Sign-in method Templates Usage Settings Extensions

Sign-in providers

Email/Password ☒ Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. [Learn more](#)

Email link (passwordless sign-in) ☒ Enable

Cancel

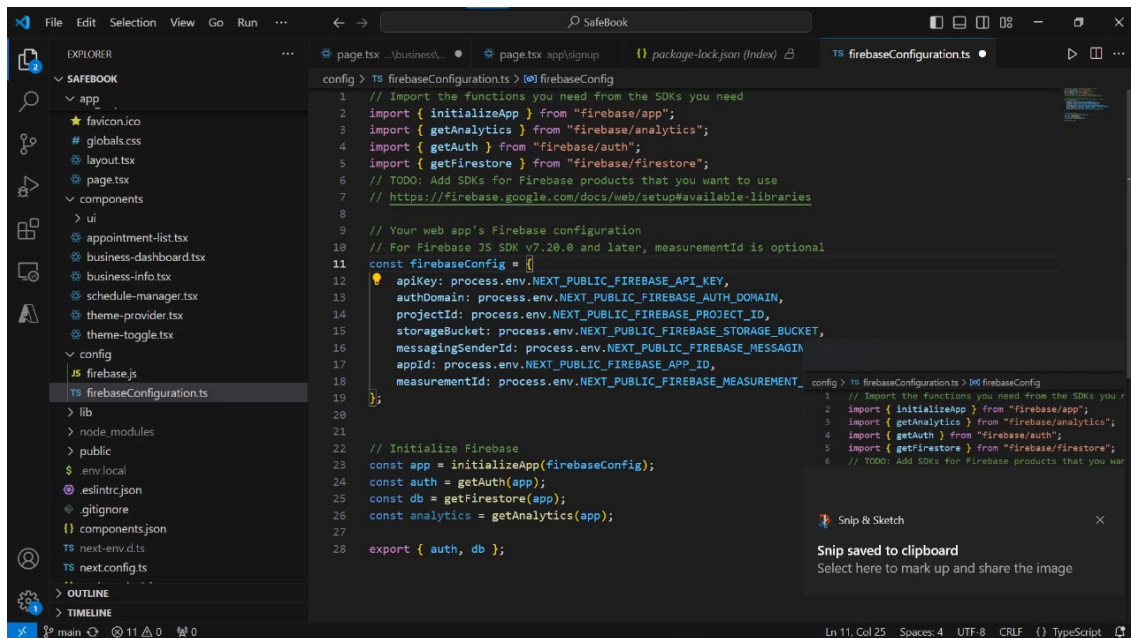
Save

Advanced

- Install Firebase in your app:

```
npm install firebase
```

- Initialize Firebase and add authentication:

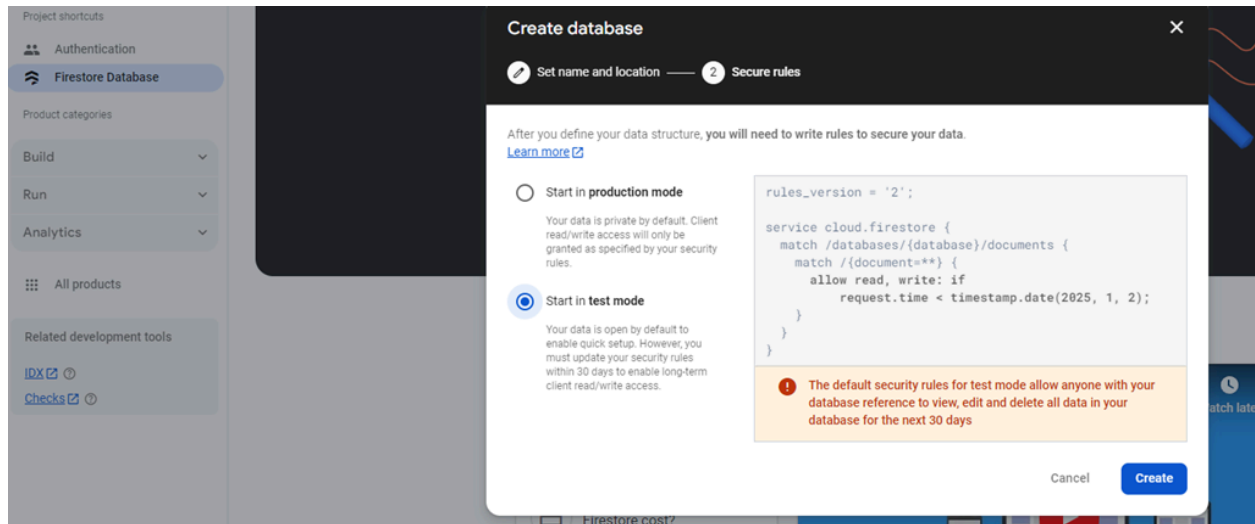


```
1 // Import the functions you need from the SDKs you need
2 import { initializeApp } from "firebase/app";
3 import { getAnalytics } from "firebase/analytics";
4 import { getAuth } from "firebase/auth";
5 import { getFirestore } from "firebase/firestore";
6 // TODO: Add SDKs for Firebase products that you want to use
7 // https://firebase.google.com/docs/web/setup#available-libraries
8
9 // Your web app's Firebase configuration
10 // For Firebase JS SDK v7.20.0 and later, measurementId is optional
11 const firebaseConfig = {
12   apiKey: process.env.NEXT_PUBLIC_FIREBASE_API_KEY,
13   authDomain: process.env.NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN,
14   projectId: process.env.NEXT_PUBLIC_FIREBASE_PROJECT_ID,
15   storageBucket: process.env.NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET,
16   messagingSenderId: process.env.NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID,
17   appId: process.env.NEXT_PUBLIC_FIREBASE_APP_ID,
18   measurementId: process.env.NEXT_PUBLIC_FIREBASE_MEASUREMENT_ID,
19 };
20
21 // Initialize Firebase
22 const app = initializeApp(firebaseConfig);
23 const auth = getAuth(app);
24 const db = getFirestore(app);
25 const analytics = getAnalytics(app);
26
27 export { auth, db };
28
```

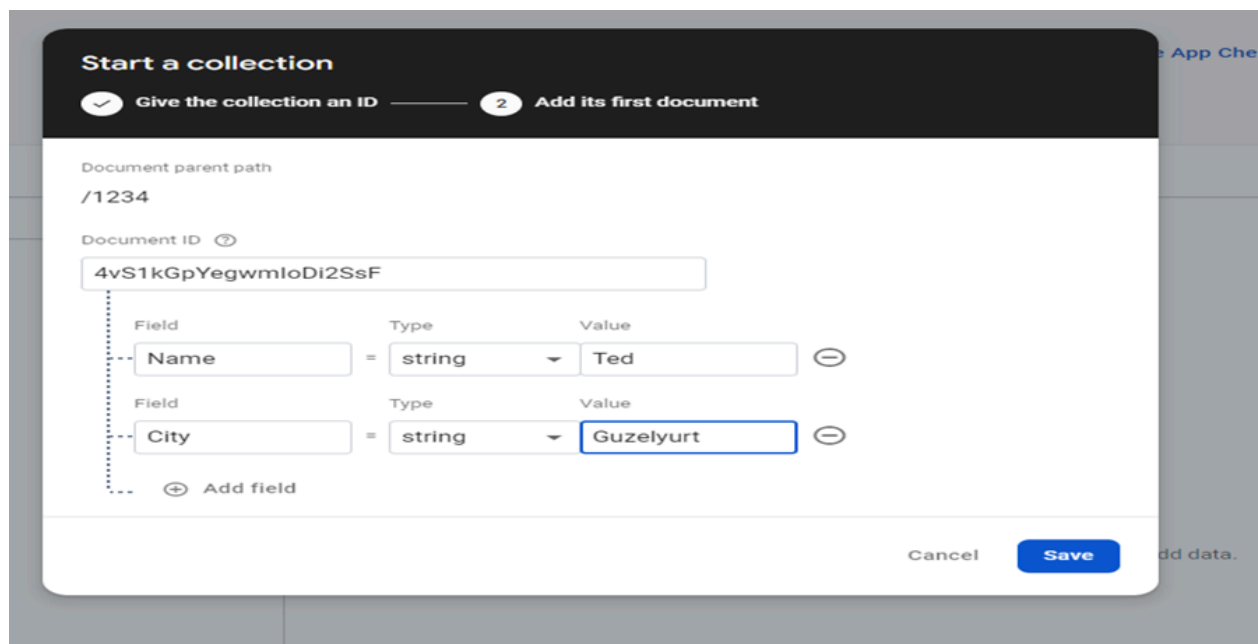
## 3. Firestore Database

1. Enable Firestore:

- In the Firebase Console, go to **Build > Firestore Database**.
- Click **Create Database** and select **Test Mode**.



## 2. Add a Document



## GITHUB CLONING URL

Below is the GitHub link containing our repository.

<https://github.com/aaronaminubandado/SafeBook.git>

## DIFFICULTIES ENCOUNTERED SO FAR

- Deciding how to structure collections and documents in Firestore was a bit tricky.
- Firestore does not support deep/multiple nested queries. Had to consider that when designing the database.
- Connecting/Integrating UI elements to the database(Firestore) was one of the major difficulties encountered.

## MILESTONES REMAINED

### Week 11

- Connecting the frontend of the application to the database.
- Creating a customer page for them to make bookings.
- Ensuring booking creation functionality is available for businesses.

### Week 12

- Implementing the logic for time slot calculation.
- Implementing notification for the users.

### Week 13

- Testing and debugging
- Application deployment
- Final UI and database improvements

### Week 14

- Report writing and application documentation

## REFERENCES

*Documentation.* (n.d.). Firebase.

[https://firebase.google.com/docs?gad\\_source=1&gclid=Cj0KCQiA\\_qG5BhDTARIsAA0UHSIG5k8NfONafuWl2SbCf2TAGr4ofmswLlegoo4ADt\\_K4dU8CEDNLkQaAiN9EALw\\_wcB&gclsrc=aw.ds](https://firebase.google.com/docs?gad_source=1&gclid=Cj0KCQiA_qG5BhDTARIsAA0UHSIG5k8NfONafuWl2SbCf2TAGr4ofmswLlegoo4ADt_K4dU8CEDNLkQaAiN9EALw_wcB&gclsrc=aw.ds)

*Next.js by Vercel - The React Framework.* (n.d.). <https://nextjs.org/>

Firebase. (n.d.). *Firebase.*

[https://firebase.google.com/?gad\\_source=1&gclid=CjwKCAiA9bq6BhAKEiwAH6bqoFNlh0ydCh3d9zDAnvqo3KjLLaVA0u1BKOGmI68yjdo\\_iDOWvogLPhoCJ\\_kQAvD\\_BwE&gclsrc=aw.ds](https://firebase.google.com/?gad_source=1&gclid=CjwKCAiA9bq6BhAKEiwAH6bqoFNlh0ydCh3d9zDAnvqo3KjLLaVA0u1BKOGmI68yjdo_iDOWvogLPhoCJ_kQAvD_BwE&gclsrc=aw.ds)