

Progress 2 Presentation

Home Dashboard + Predictive Analytics

Admin
John Smith

[Dashboard](#)

Patients

Messages

Insights

Manage beds

Manage rooms

Users

Overall

Occupancy

51/100 **24**
open in use
10 **15**
reserved clean up



New admissions



Departments

Oncology

20 beds

10 **3** **2** **5**
open in use reserved clean up

Neurology

30 beds

15 **8** **4** **3**
open in use reserved clean up

Cardiology

50 beds

24 **15** **10** **1**
open in use reserved clean up

Rooms

Room: Neurology room 5

Change room

Bed 1

In use

Patient name: George Porter
Sex: Male
DOB: 08/26/1981
Rep: Nurse Anna Garner

Bed 2

In use

Patient name: Masha DaBear
Sex: Male
DOB: 05/21/2010
Rep: Nurse Anna Garner

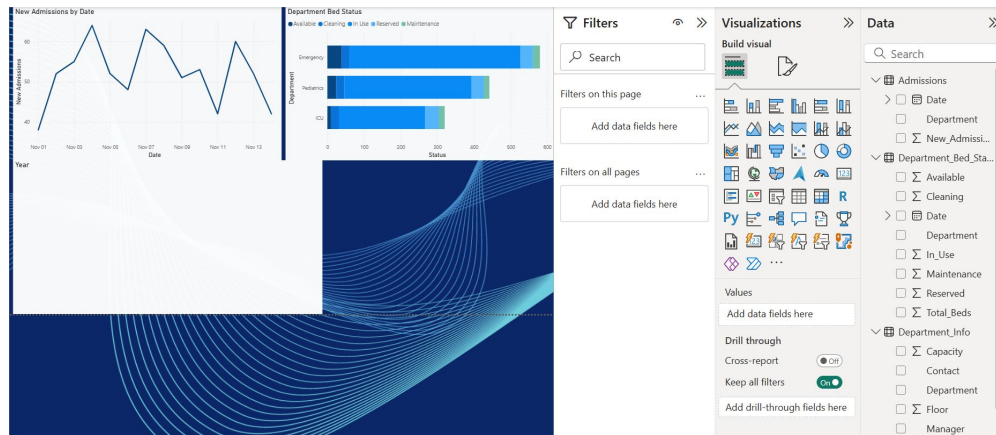
Bed 3

In use

Patient name: Tina Chang
Sex: Female
DOB: 01/10/1975
Rep: Nurse Michael Reed

Home Dashboard

- Completed wireframes
- Currently coding the front-end
- Next steps: integrate the back-end and deploy on GCP web hosting



Predictive Analytics on PowerBI

- Creating interactive charts and graphs using mock data
- Next steps: integrate these visualizations into the actual dashboard

Current fix for host issue/future

Current – Firebase Hosting (Frontend Only)

Plan:

Firebase Hosting (your website)

↓ HTTPS call (fetch/axios)

Cloud Run API (Node.js server)

↓ Secure connection

Cloud SQL (MySQL/Postgres)

```
=== Deploying to 'sql.databasetest-473020'...

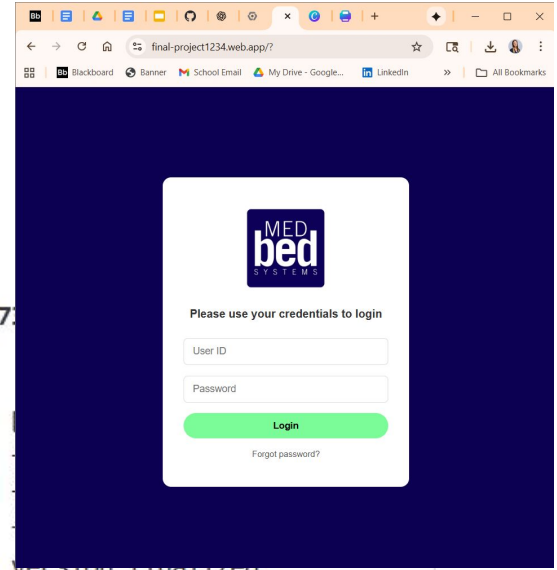
i  deploying hosting
i  hosting[sql.databasetest-473020]: v1.0.0: building...
i  hosting[sql.databasetest-473020]: v1.0.0: upload complete
+  hosting[sql.databasetest-473020]: v1.0.0: version finalized
i  hosting[sql.databasetest-473020]: v1.0.0: releasing new version...
+  hosting[sql.databasetest-473020]: v1.0.0: release complete

+  Deploy complete!
```

Project Console: <https://console.firebase.google.com/project/sql.databasetest-473020/overview>

Hosting URL: <https://sql.databasetest-473020.web.app>

PS C:\Users\stell\OneDrive\Desktop\final IT project\capstone-f25-project-repository-group-5> █



Mock Data and Backend testing

The screenshot shows the mockaroo.com interface with a table of fields for a medical record. The fields are:

Field Name	Type	Options
Patient_id	Row Number	blank: 0% [Σ] [X]
First_name	First Name	blank: 0% [Σ] [X]
Last_name	Last Name	blank: 0% [Σ] [X]
Date_of_birth	Datetime	01/01/1925 to 12/31/2024 format: m/d/yyyy blank: 0% [Σ] [X]
Gender	Gender	blank: 0% [Σ] [X]
Medical record number	NHS Number	blank: 0% [Σ] [X]
Admission_status	Number	min: 1 max: 3 decimals: 0 blank: 0% [Σ] [X]
Admission_date	Datetime	01/01/2015 to 12/31/2025 format: m/d/yyyy blank: 0% [Σ] [X]
Discharge_date	Datetime	01/01/2015 to 12/31/2025 format: m/d/yyyy blank: 0% [Σ] [X]
Priority_level	Number	min: 1 max: 5 decimals: 0 blank: 0% [Σ] [X]
Admit_reason	ICD9 Dx Desc (Long)	blank: 0% [Σ] [X]

Buttons at the bottom: + ADD ANOTHER FIELD, GENERATE FIELDS USING AI...

The screenshot shows the mockaroo.com interface with a list of field types to choose from. The types are:

- All (211)
- Advanced (10)
- Basic (30)
- Car (4)
- Commerce (18)
- Construction (6)
- Crypto (7)
- Finance (9)
- Health (26)
- IT (21)
- Location (14)
- Nature (5)

Buttons at the bottom: GENERATE DATA, PREVIEW, SAVE AS, DERIVE FROM EXAMPLE, MORE

1. Create Backend Folder

backend/

-server.js (main API code) contains the code that runs your server and handles requests from clients.

-package.json (Node.js app info)

-.env (database credentials) keeps secrets out of actual code (passwords, ect.)

-certs/ (SSL certs for secure DB connection)

2. Write API Code in server.js

Work in progress, research needed, looking into Cloud Run? Other APIs

Current fix for host issue/future

Once we begin integrating the backend, we will...

- create a Google Cloud Run service that runs our Node.js server and connects securely to our Cloud SQL database
- The Firebase-hosted frontend will communicate with this backend using HTTPS API requests
- Cloud Run will handle authentication and secure environment variables, and it will use a private connection to Cloud SQL

✅ What You'll Build When You're Ready for SQL

Component	Purpose
Firebase Hosting	Displays your UI ✅ (you already deployed this)
Cloud Run (Node.js)	Backend API to talk to SQL
Cloud SQL	Stores hospital bed & patient data