

• • •

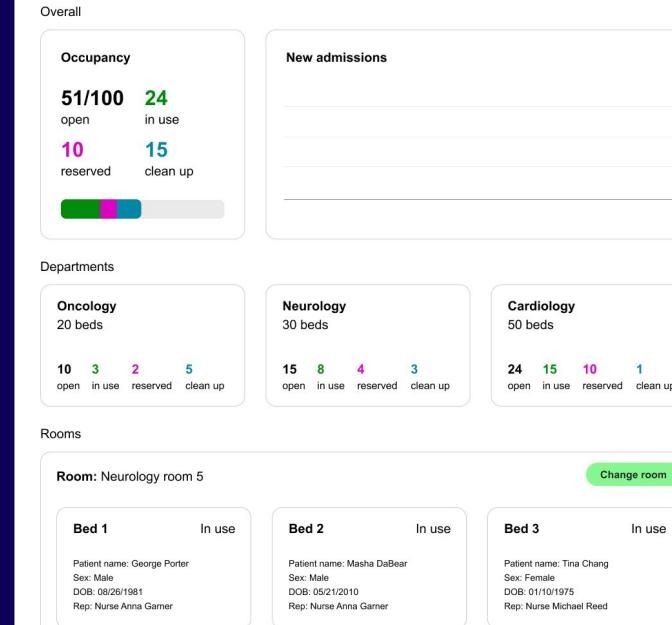
Progress 2 Presentation



Home Dashboard + Predictive Analytics

Admin
John Smith

Dashboard
Patients
Messages
Insights
Manage beds
Manage rooms
Users

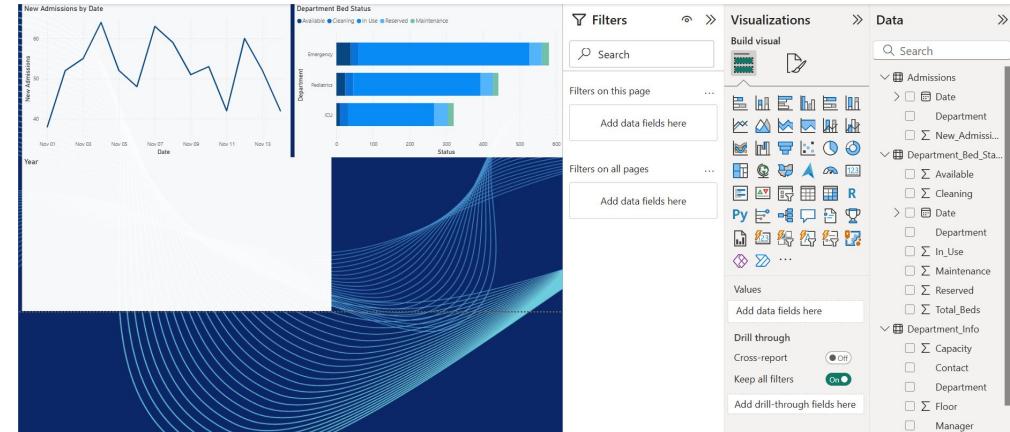


Predictive Analytics on PowerBI

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

Home Dashboard

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





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 'sqldatabasetest-473020'

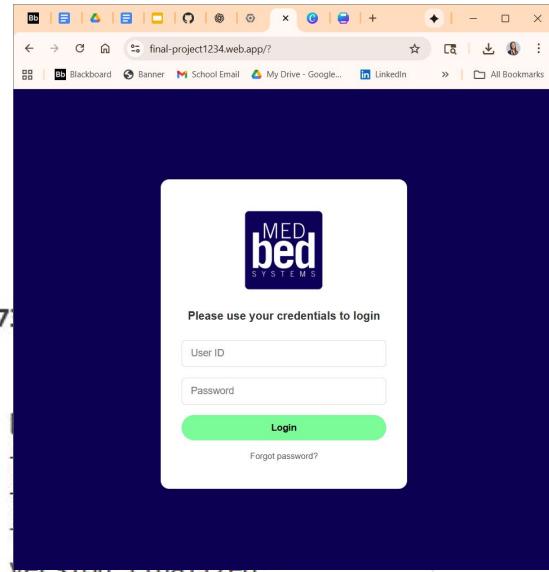
i  deploying hosting
i  hosting[sqldatabasetest-473020]: 
i  hosting[sqldatabasetest-473020]: 
+ hosting[sqldatabasetest-473020]: 
i  hosting[sqldatabasetest-473020]: 
+ hosting[sqldatabasetest-473020]: version finalized
i  hosting[sqldatabasetest-473020]: releasing new version...
+ hosting[sqldatabasetest-473020]: release complete

+ Deploy complete!
```

Project Console: <https://console.firebaseio.google.com/project/sqldatabasetest-473020/overview>

Hosting URL: <https://sqldatabasetest-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 for generating mock data. On the left, a table lists fields with their types and options. Fields include Patient_id (Row Number), First_name, Last_name, Date_of_birth (Datetime), Gender, Medical record number (NHS Number), Admission_status (Number), Admission_date (Datetime), Discharge_date (Datetime), Priority_level (Number), and Admit_reason (ICD9 Dx Desc (Long)). Each field has dropdown menus for type, options like blank or sum, and delete buttons. At the bottom are buttons for '+ ADD ANOTHER FIELD' and 'GENERATE FIELDS USING AI...'.

The screenshot shows the 'Choose a Type' section of the Mockaroo.com interface. It lists various categories of data types: All (211), Advanced (10), Basic (30), Car (4), Commerce (18), Construction (6), Crypto (7), Finance (9), Health (26), IT (21), Location (14), and Nature (5). To the right, specific examples are provided for each category, such as Drug Company (Eli Lilly and Company, Novartis Pharmaceuticals Corporation, Teva Pharmaceuticals USA Inc), Drug Name (Brand) (Cialis, Nexium, Lipitor), Drug Name (Generic) (Naproxen Sodium, Selenium Sulfide, Acetaminophen), FDA NDC Code (58443-0022, 58517-001, 44924-011), HCPCS Code (An HCPCS code), HCPCS Name (An HCPCS procedure name), Hospital City (The city of a US-based hospital), Hospital Name (The name of a US-based hospital), Hospital NPI (The NPI of a US-based hospital), Hospital Postal Code (The postal code of a US-based hospital), Hospital State (The state of a US-based hospital), Hospital Street Address (The street address of a US-based hospital), ICD10 Diagnosis Code (ICD10 diagnosis code. Source: cms.gov), ICD10 Dx Desc (Long) (Long description of diagnosis from ICD10. Source: cms.gov), and ICD10 Dx Desc (Short) (Short description of diagnosis from ICD10. Source: cms.gov). A search bar at the top right says 'Find Type...'. At the bottom, there's a note about generating data using AI and buttons for 'GENERATE DATA', 'PREVIEW', 'SAVE AS...', 'DERIVE FROM EXAMPLE...', 'MORE ▾', and 'GENERAL'.

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 <input checked="" type="checkbox"/> (you already deployed this)
Cloud Run (Node.js)	Backend API to talk to SQL
Cloud SQL	Stores hospital bed & patient data