Deployment Manual

MED-X AI

Med-X
May 2024
Team 1B

Table of Contents

Introduction

Welcome to the deployment manual for the Med-X AI application. This document is designed to provide a complete step-by-step solution for successful deployment of our software on your system. It is meant for use by developers, users, & system administrators.

In this manual we will be covering the following topics:

1. **System Requirements** – an outline of the hardware and software requirements
2. **Installation procedures** – step by step guide on how to install the software in your system
3. **Testing & Troubleshooting** – ensures the front end and back end are functioning appropriately and provides commands in the event the application is not working accordingly

System Requirements

Our software will run on major system operating systems without having a big impact on hardware components. Before attempting to install the software components please make sure your system meets the following hardware requirements.

Hardware requirements (Desktop)

Processor (CPU):

Minimum: Core i5/i7-U 1.5 Ghz or AMD equivalent

Recommended: Core i7-H 2.3 Ghz or Ryzen 5 3600

Apple M1 (8-core)

Memory (RAM):

Minimum: 8 Gb

Recommended: 16+ GB

Available Storage on Disk:

Minimum: 10 Gb of SSD

Recommended: 20+ Gb of SSD of space or equivalent for optimal performance

GPU:

Minimum: Intel Iris Xe with 1.30 Ghz

Recommended for AI development: GeForce RTX 3050 Ti 4Gb or higher

Software requirements (Desktop)

Operating System:

Windows 10/11

MacOS Big Sur or newer

Ubuntu v 18+

Installation Procedures

Installing Visual Studio Code (VS Code):

1. **Download VS Code:**
   - Visit the official Visual Studio Code website at https://code.visualstudio.com/.
   - Click on the *Download* to download the installer for your operating system (e.g., Windows, macOS, or Linux).
2. **Install VS Code:**
   - Once the installer is downloaded, run the installer executable.
   - Follow the on-screen instructions to install VS Code
   - Choose your preferred installation settings and location (you can usually leave the default settings as they are).
3. **Launch VS Code:**
   - After installation, you can launch VS Code from your system's application menu or desktop shortcut.

Installing Node.js and npm:

1. **Download Node.js:**
   - Open your web browser and visit the official Node.js website at https://nodejs.org/en/download
   - On the website, we recommend installing version 18.20.2 (LTS). The LTS version is for stability.
2. **Install Node.js:**
   - Once the installer is downloaded, run the installer executable.
   - Follow the installation wizard's instructions. You can typically use the default settings.
   - During the installation process, you may be asked to accept the terms and conditions.
3. **Verify Node.js and npm Installation:**
   - Open a command prompt or terminal window.
   - To verify that Node.js and npm have been successfully installed, type the following commands and press *Enter*:
     ```
     node -v
     npm -v
     ```
   - You should see the installed Node.js and npm versions displayed in the terminal.

You have now successfully installed Visual Studio Code and Node.js with npm. You are ready to start using these tools for your development projects.

Cloning Git Repository

1. **Create a folder on your desktop**
   - Right click anywhere on your screen
   - Select *New* and then *Folder*

2. **Open VSCode:**
   - Click on *File*
   - Open the Folder you created in step 1
   - On the top left corner find the three dots or find Terminal
   - Select *New Terminal*

3. **Clone the Repository:**
   - Use the command below in the terminal:

   git clone https://github.com/htmw/2024S-Med-X.git .

   *Do not forget to include the period

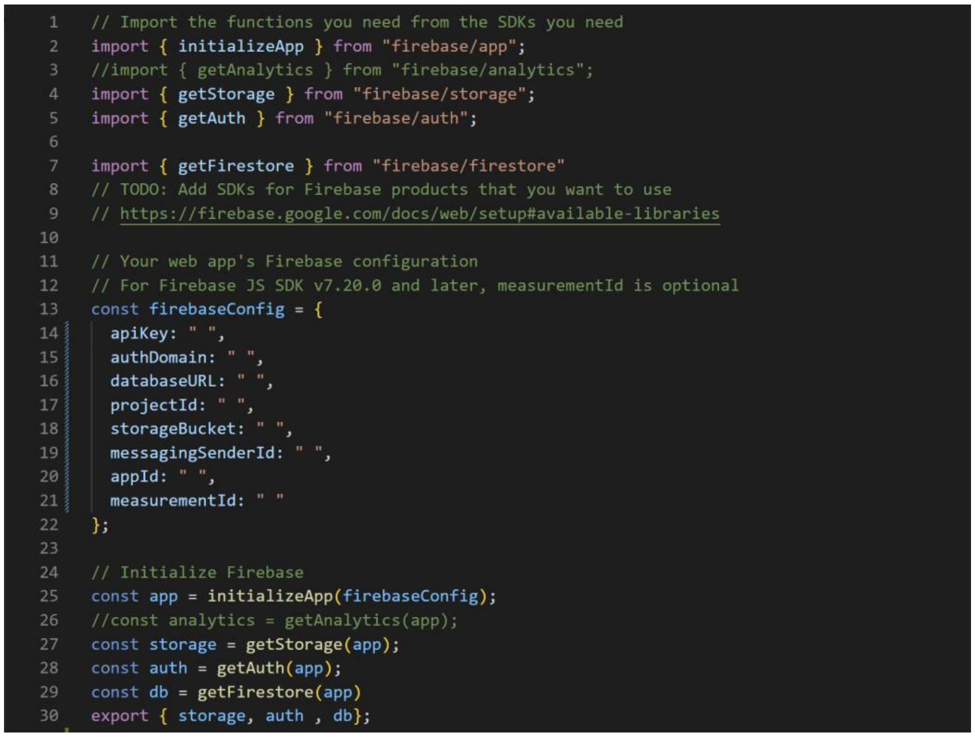
You have successfully cloned the Git repository.

Firebase

1. **Login to Firebase and use your credentials or create an account:**
   - Visit the Firebase website at https://console.firebase.google.com/

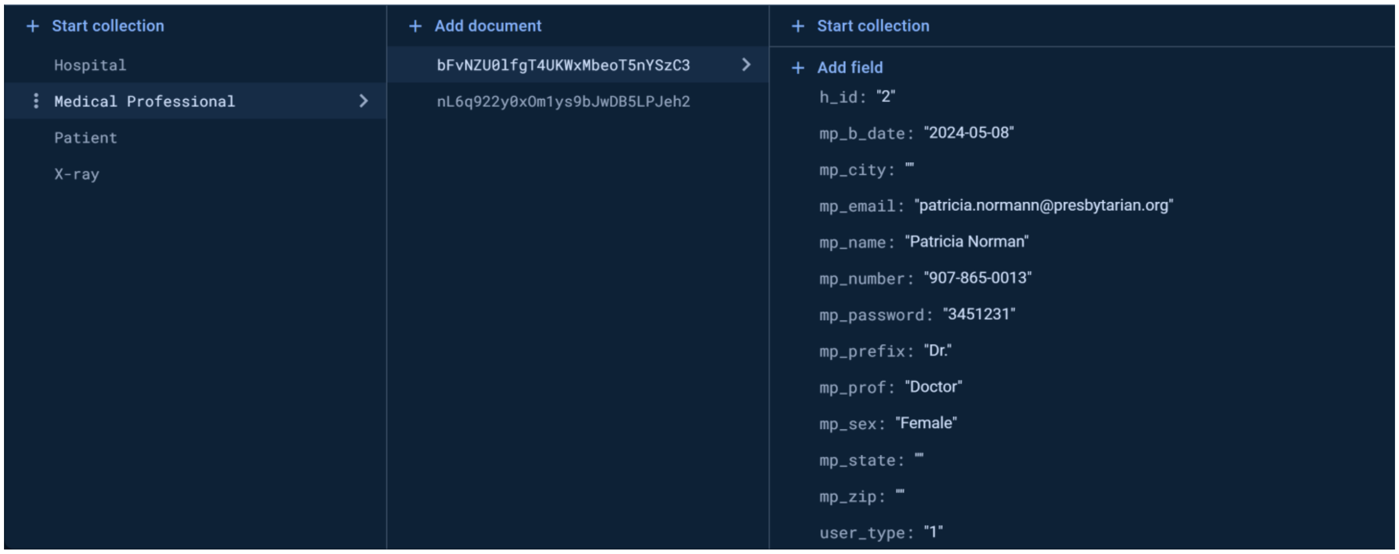
2. **Create your own Firebase project and locate your web app's Firebase configuration:**
   - Select *Add Project*
   - Enter your Project Name
   - Follow the Firebase instructions
   - Locate your firebase Config file by:
     - ◊ Click on *Web Icon*
     - ◊ Create an App nickname
     - ◊ Click on *Register app*

3. **Return to VSCode and locate the firebase.js:**
   - Copy and paste your own firebase Config APIs

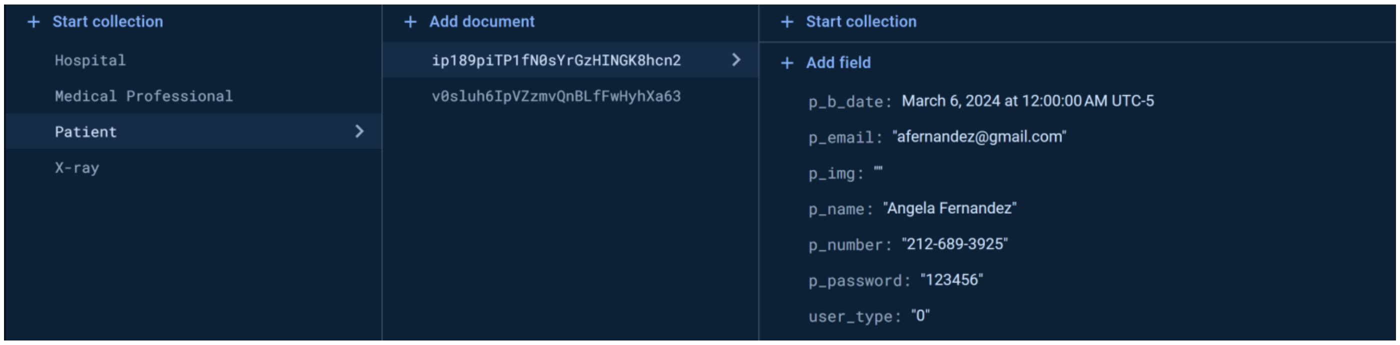


```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
//import { getAnalytics } from "firebase/analytics";
import { getStorage } from "firebase/storage";
import { getAuth } from "firebase/auth";

import { getFirestore } from "firebase/firestore"
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: " ",
  authDomain: " ",
  databaseURL: " ",
  projectId: " ",
  storageBucket: " ",
  messagingSenderId: " ",
  appId: " ",
  measurementId: " "
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
//const analytics = getAnalytics(app);
const storage = getStorage(app);
const auth = getAuth(app);
const db = getFirestore(app)
export { storage, auth , db};
```

4. **Create the tables in Firebase:**
    - Click on *Build*
    - Select *Firestore Database*
    - Click on *Create database*
    - Click *Next*
    - Select *Start* in production mode
    - Click *Create*
    - Click on *Rules*
        - ◊ Find allow read, write: if false;
        - ◊ Replace false with true
        - ◊ If done correctly, it should now be read, write: if true;
    - Select *Start collection*
        - ◊ Create each collection: Hospital, Medical Professional, Patient, X-ray
        - ◊ Follow the screenshots below and only copy the fields leave the value blank. As you use the web application these values will be populated.



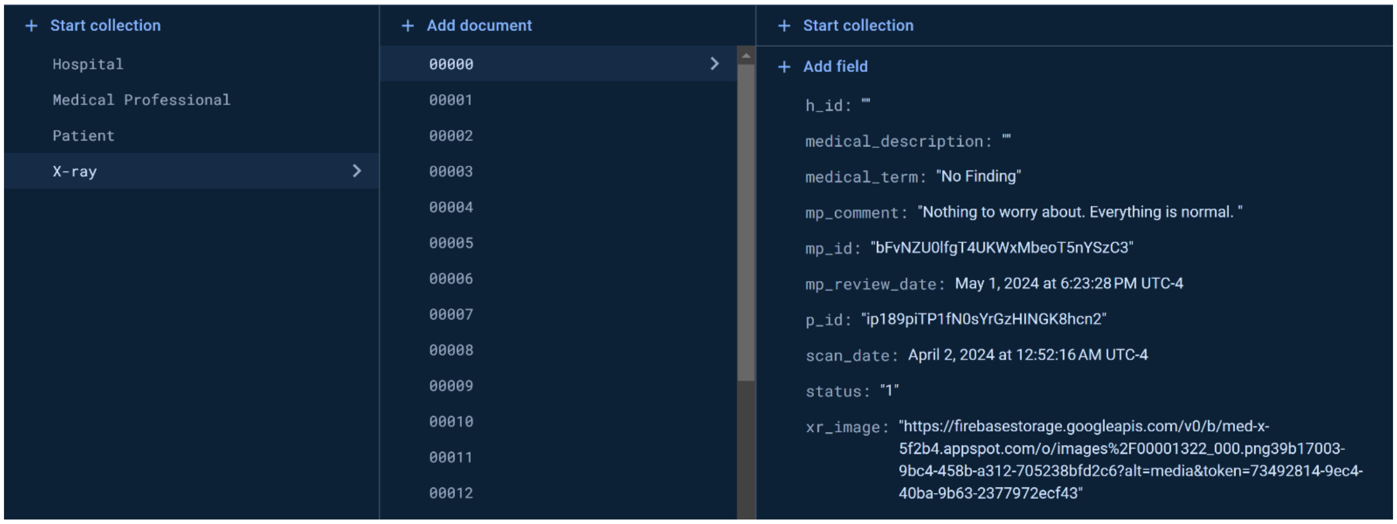
```
h_id: "2"
mp_b_date: "2024-05-08"
mp_city: ""
mp_email: "patricia.normann@presbytarian.org"
mp_name: "Patricia Norman"
mp_number: "907-865-0013"
mp_password: "3451231"
mp_prefix: "Dr."
mp_prof: "Doctor"
mp_sex: "Female"
mp_state: ""
mp_zip: ""
user_type: "1"
```



Hospital
Medical Professional
Patient
X-ray

ip189piTP1fN0sYrGzHINGK8hcn2
v0sluh6IpVZzmvQnBLfFwHyhXa63

```
p_b_date: March 6, 2024 at 12:00:00 AM UTC-5
p_email: "afernandez@gmail.com"
p_img: ""
p_name: "Angela Fernandez"
p_number: "212-689-3925"
p_password: "123456"
user_type: "0"
```



Hospital
Medical Professional
Patient
X-ray

00000
00001
00002
00003
00004
00005
00006
00007
00008
00009
00010
00011
00012

```
h_id: ""
medical_description: ""
medical_term: "No Finding"
mp_comment: "Nothing to worry about. Everything is normal. "
mp_id: "bFvNZU0lfgT4UKWxMbeoT5nYSzC3"
mp_review_date: May 1, 2024 at 6:23:28 PM UTC-4
p_id: "ip189piTP1fN0sYrGzHINGK8hcn2"
scan_date: April 2, 2024 at 12:52:16 AM UTC-4
status: "1"
xr_image: "https://firebasestorage.googleapis.com/v0/b/med-x-
          5f2b4.appspot.com/o/images%2F00001322_000.png39b17003-
          9bc4-458b-a312-705238bfd2c6?alt=media&token=73492814-9ec4-
          40ba-9b63-2377972ecf43"
```

Starting the server (Docker)

Provided that you have WSL installed in your Windows machine; If you are using Mac or Linux proceed to Step 1

1. **Download Docker**
   - Visit the official Docker website at https://www.docker.com/products/docker-desktop/

2. **Open Docker and let it run in the background**

3. **Open VSCode**
   - In the terminal window click on the plus sign (+) to create an additional terminal

4. **Change your directory to flask server**

   ```
   cd flask-server
   ```

5. **Create the back-end image that runs the AI**
   - Run the command to create an image of the back-end

     ```
     docker build -t flask-app  .
     ```

     *Do not forget to include the period

   - Wait until the image is created

6. **Create the front-end image that runs the UI**
   - Open a new terminal window by clicking on the plus sign (+)
   - Navigate to the Project folder

     ```
     cd Project
     ```

   - Navigate to the React app

     ```
     cd my-react-app
     ```

   - Run the command to create an image of the front-end

docker build -t med-x:dev .

*Do not forget to include the period

- Wait until the image is created

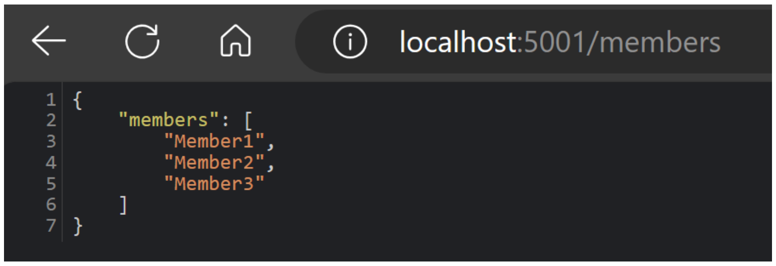
7. Return to the flask-server terminal

- Run the command

docker run -p 5001:5000 flask-app

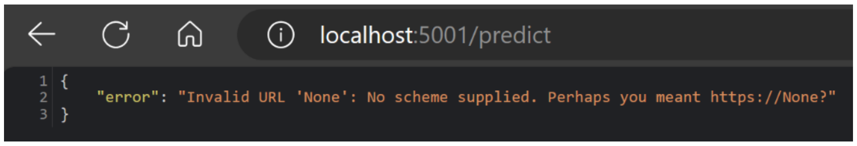
8. Test the back-end API

- Return to Docker Desktop
- Click on *Containers*
- Click on the ports *5001:5000* shown in flask-app container
- A browser window will open:
  - ◊ Enter the following URL to test the members API
    1. http://localhost:5001/members
    2. A key value pair table will be displayed showing Member1, Member2, Member3. If you are viewing this, it means the API is working



```
{
    "members": [
        "Member1",
        "Member2",
        "Member3"
    ]
}
```

  - ◊ Enter the following URL to test the back-end API
    1. http://localhost:5001/predict
    2. If you can access this URL it means that the back-end application is working



```
{
    "error": "Invalid URL 'None': No scheme supplied. Perhaps you meant https://None?"
}
```

9. **Test the front-end UI**
   - Return to VSCode
   - Click on the my-react-app terminal
     - ◊ Run the command

```
docker run -p 3000:3000 med-x:dev
```

     - ◊ Return to Docker Desktop
   - Click on *Containers*
     - ◊ Click on the ports *3000:3000* shown in my-react-app container
   - A browser window will open
     - ◊ The Med-X app will load
     - ◊ If you can see the Med-X app congratulations, you have followed the steps successfully

Troubleshooting:

In case you want to develop the application further and you need to build new images without past data interference run the following commands respectively for the back-end and front-end

```
docker build -t flask-app . --no-cache

docker build -t med-x:dev . --no-cache
```

Contact information

Date: May 8, 2024

Feedback and Updates

We value your feedback and are committed to continuously improving the deployment process and the software itself. Your input is invaluable to us, and we encourage you to share your thoughts, suggestions, or report any issues you encounter during the deployment.

Email: **medxteam2024@gmail.com**