1/1/2021

Pandemic Watch System

Mobile Application

CCR Lab Korea Aerospace University
South Korea

Contents

INTRODUCTION 3

1.1 Pandemic Watch System Overview 3

1.2 Pandemic Watch System (PWS) Architecture 4

1.3 Mobile Application 4

1.4 Access Point 5

1.5 Hospital 6

1.6 Server 7

1.7 Tree and Network Construction 7

Mobile Application 9

2.1 Visual Studio Code Installation 9

2.2 Installation of Flutter in visual studio 13

2.3 Installation of Flutter SDK 14

2.4 Installation of Flutter in visual studio 15

2.5 Download PWS App code 21

2.6 Importing PWS App in Visual Studio 21

2.7 Running and debugging PWS App 24

Access Point (AP) 25

3.1 Raspberry-PI Configuration 25

3.2 Downloading and Installing Raspberry Pi OS 25

3.3 Run python code 27

Hospital 28

4.1 GUI development for the Hospital Webpage 28

Server 30

5.1 Server for the Pandemic Watch System (PWS) 30

5.2 Installation of Node.js 30

5.3 PostgreSQL Installation 33

5.4 Nodejs-PostgreSQL Connection 36

5.5 Operational Flow 37

5.6 Registration Sequence diagram 39

5.7 Sign in Sequence diagram 39

5.8 CSV file reception from client 40

Tree Construction 41
6.1 Website Development 41
6.1.1 ISM Tree Construction on the basis of Bluetooth 41
6.1.2 ISM Tree Construction on the basis of GPS 44
6.1.3 ISM Network Construction on the basis of Bluetooth IDs 45
Geo Fencing 46
7.1 Website Development 46

INTRODUCTION

1.1 Pandemic Watch System Overview

Nowadays development of smartphones has determined the abundant and prevalent computation period. Smartphone is remarkable to users in everyday life, it also contains built-in sensors that proficiently sense relevant information related to its users while being supported with an extensive range-ability of network contacts. For the minimization of pandemic disease (COVID-19), we develop the Smartphone Sensors-based Platform called Pandemic Watch System (PWS). Although mobile applications are used effectively in the management of chronic diseases [1,2], the ongoing epidemic of COVID-19 has further enhanced the need for mobile application solutions to reduce the risk of infection caused by exposure [3- 4]. Mobile technology has been used in many ways to control the spread of COVID-19. Mobile applications are accessible, acceptable, and easily accepted, and have the potential to support social segregation efforts. Therefore, they have been extensively developed and used over the past few months to "curb" the growing number of cases of COVID-19, providing information and information to ordinary people while trying to relieve stress from health care systems.

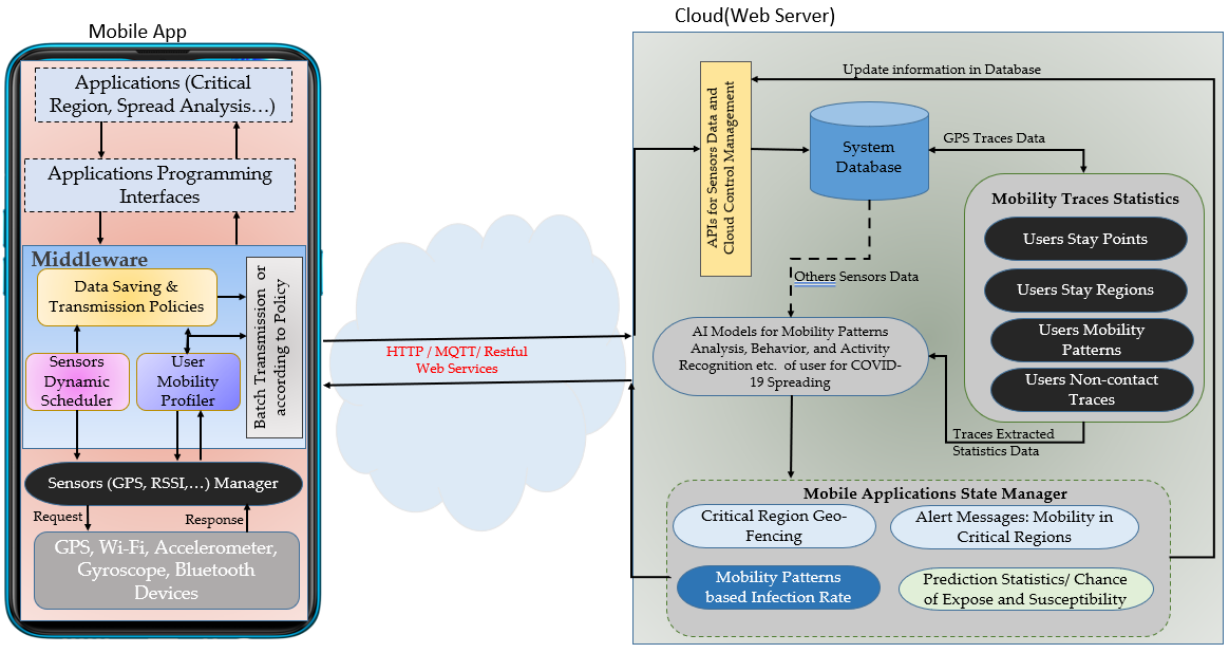


Figure. 1: Architecture of the epidemic watch system (EWS) for the minimizing of COVID-19 spread

Pandemic watch system (PWS) includes mobile application (contact tracing), access point, Geo-fencing, GPS traces, AI models for mobility patterns analysis, behavior, and activity recognition of user for COVID-19 spreading. In the figure 2 shows, the architecture of the epidemic watch system (EWS) for the minimizing of COVID-19 spread. Pandemic watch system (PWS) for the minimization of COVID-19 spread is shown in figure 1.

1.2 Pandemic Watch System (PWS) Architecture

Pandemic watch system (PWS) is developed for the minimization of covid-19 spread. The framework architecture for the collection of user trace's information including mobile app (for the contact tracing), access point, hospital portal, geo-fencing and tree construction is shown in figure 2.

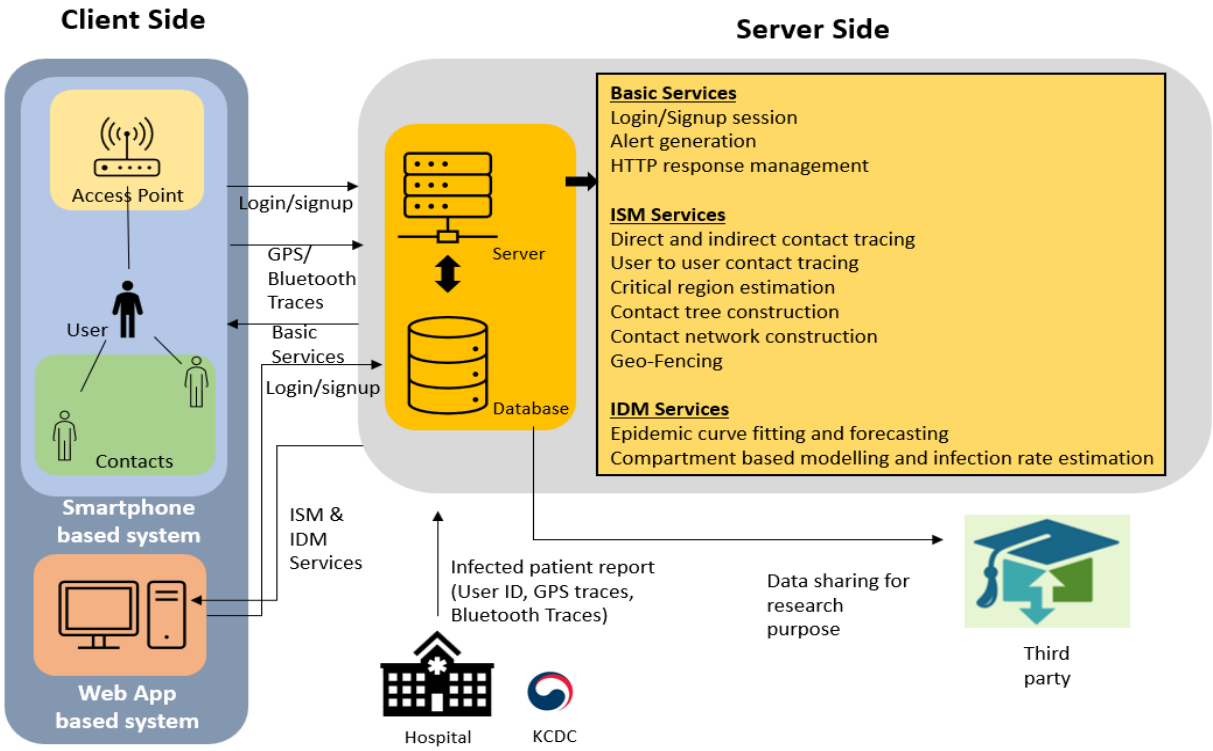


Figure. 2: overview of the PWS including mobile app, hospital, access point, geo-fencing, tree constructions

1.3 Mobile Application

Smartphone is remarkable to users in everyday life, it also contains built-in sensors that proficiently sense relevant information related to its users while being supported with an extensive

range-ability of network contacts. For the contact tracing, developed the mobile app that collect the user information, user contact traces, user location, and some other sensors information including accelerometer, gyroscope, speed sensor. Bluetooth and GPS based contact tracing is done with the help of GPS and Bluetooth data.

Mobile app contains, language selection, user registration, user login, email verification, and main screen. Functionalities including, sensors data collection, beacon transmission, beacon receiving, data storing locally in a mobile device, and data upload on the server is done automatically. The figure 4 shows the overview of the mobile app GUI.

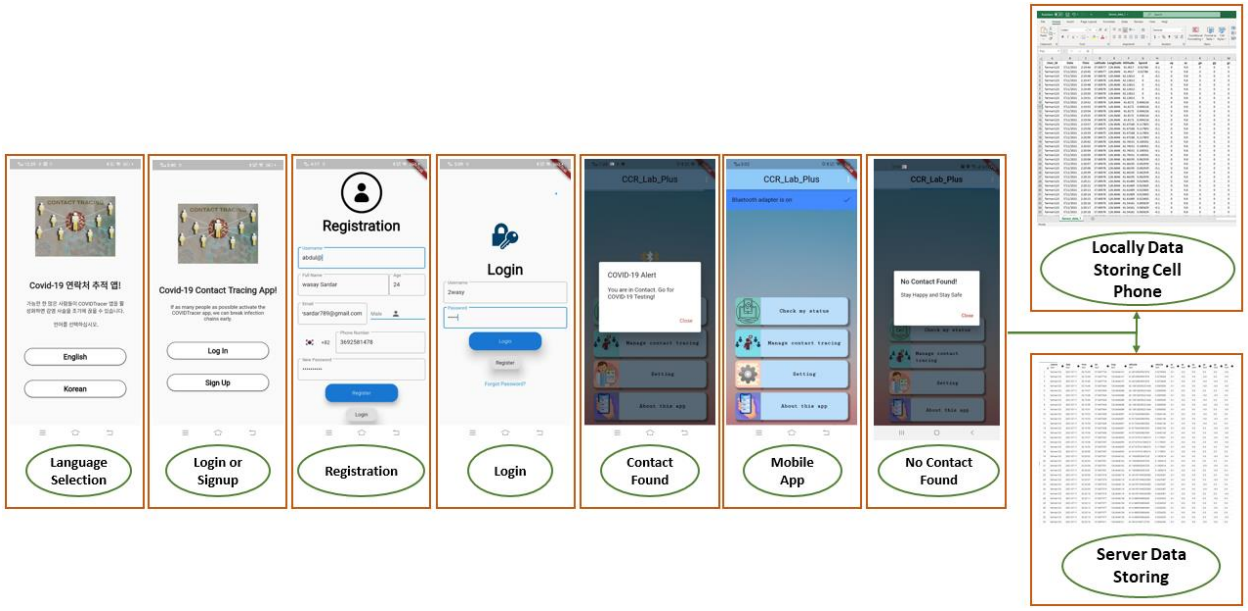


Figure. 3: Mobile APP GUI of the epidemic watch system (EWS) for the minimizing of COVID-19 spread

1.4 Access Point

Access point contains the contact traces of all the people who visit this place. Access point helpfu l for the collecting of contacts of those people who visit when some COVID-19 infected person v isit that place. COVID-19 spread by the infected person in that place where he/she visit, after som e time when other person came, he/she may not be found in the contacts of that person (contacts t hat collect with the help of mobile app) because of their no physical interaction, but may be he/sh e infected because of COVID-19 spread is there at that place where infected person is visit. With the help of access point, we found the contact traces of that person who visit same place where in

fected person visited. The figure 6 shows the overview of the access point (AP) functionalities.

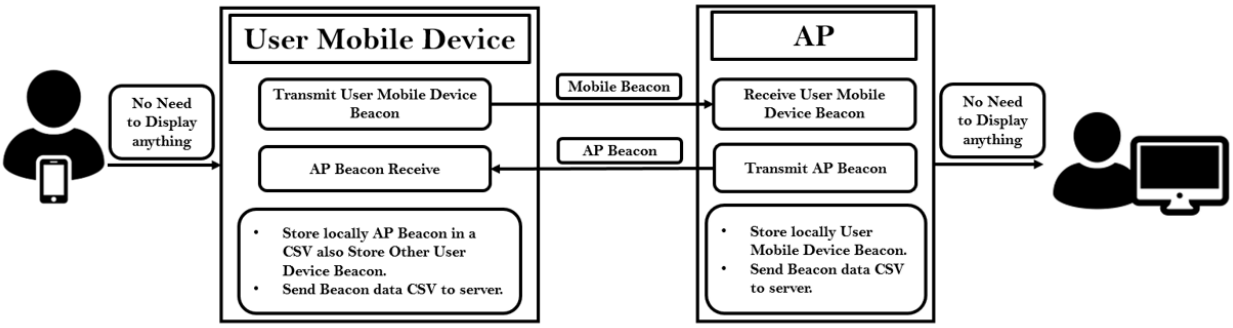


Figure. 4: Overview of the access point (AP)

1.5 Hospital

Hospital portal is needed for the storing record of the COVID-19 infected and recover persons. Hospital portal contains admin portal (For the authorizing of each hospital), hospital registration, hospital login, and user status updating. The figure 7 shows the GUI of the hospital portal.

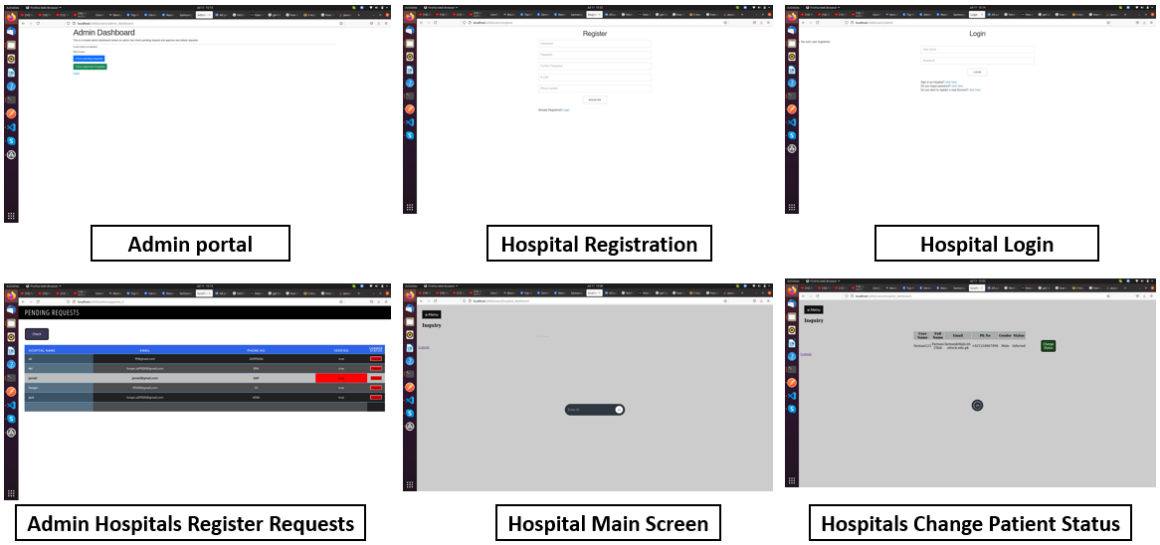


Figure. 5: Patients report collection hospital portal

1.6 Server

On the server, we implement some quires including data collection, user registration, email verification and contact tracing. In the figure 10 shows the how many persons are contacted with infected person.

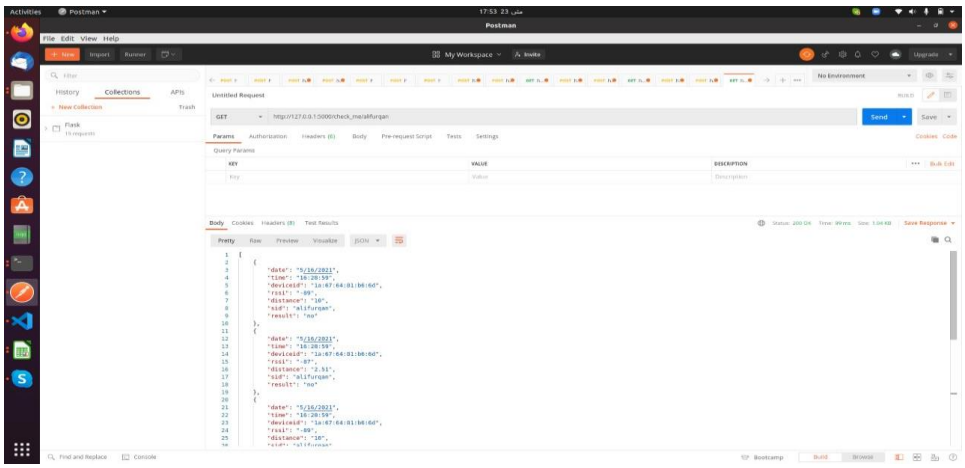


Figure. 6: Implementation of contact tracing quire on server

1.7 Tree and Network Construction

From the database we took the data of the patient and checked all those users who were close to the patient and according to that we constructed a tree. Where we checked the distance of the user, time, direction using latitude and longitude from the patients. According to those we calculate the probability of infection for the suspected person.

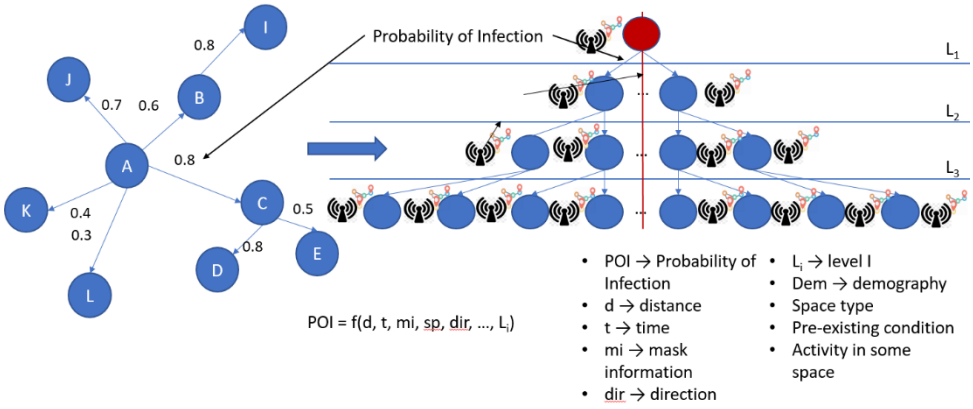


Figure. 7: Overview of the tree construction

Another way to find the contacts of contact is the Network formation. The network is constructed

based on GPS and Bluetooth both. The algorithm for Network construction is shown in Fig. 8 (a) The red nodes represent the patient and the yellow represent those suspected user who were reall y close to the patient. The green are suspected nodes but not that much serious which are show in Fig. 8 (b).

(a) (b)

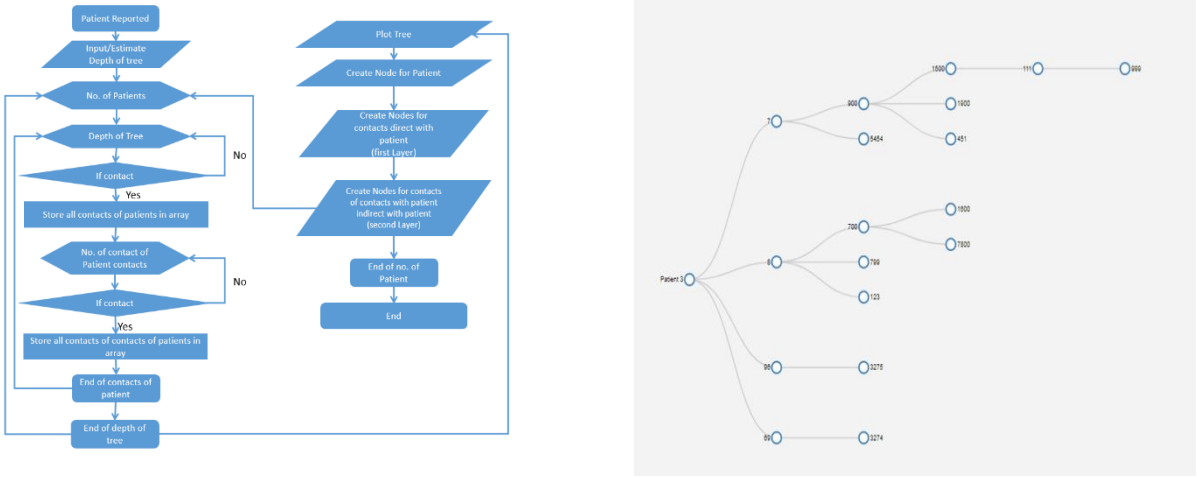


Figure 8: Algorithm and result for Network Construction after applying the algorithm on the data

Mobile Application

2.1 Mobile Application Development

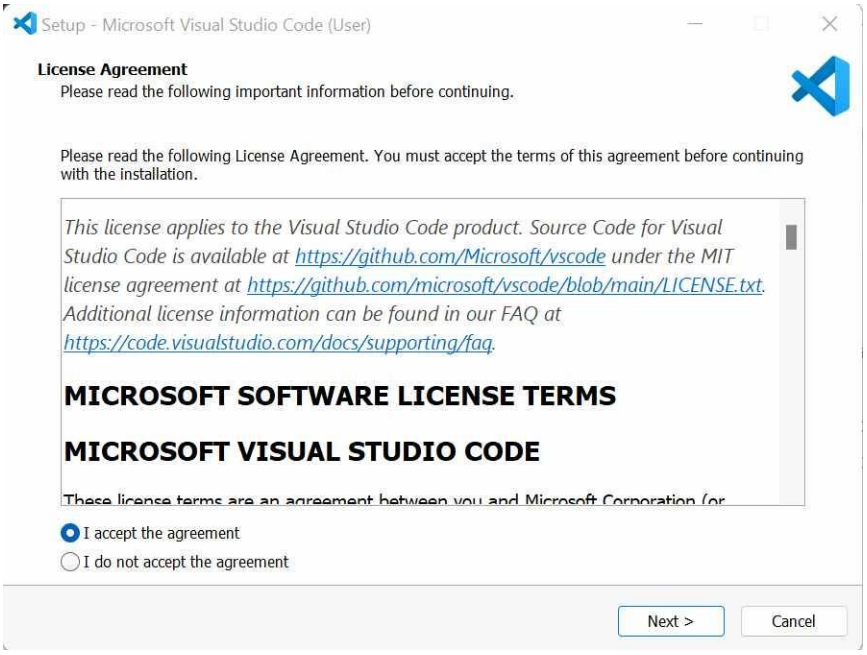
For the contact tracing, we design the mobile application in Flutter mobile application platform. Flutter is an open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase. Flutter is one of the best solutions to develop apps for Android and iOS, without having to write in a different codebase for each platform. The smartphone versions of these apps function as true, native apps on Apple and Android devices and are compiled for the respective platform before publication.

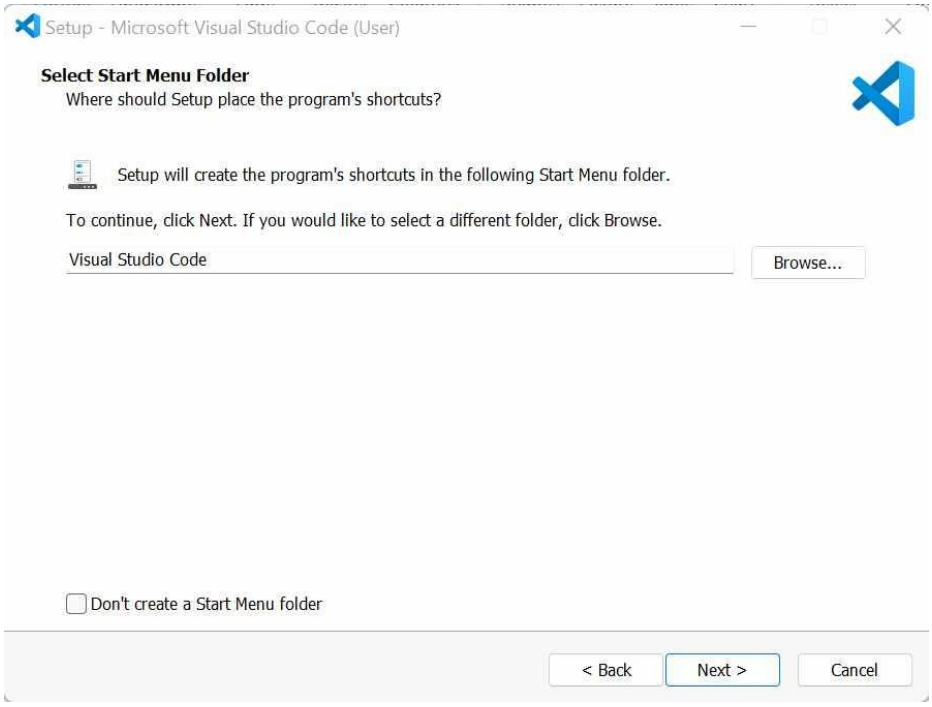
2.1 Visual Studio Code Installation

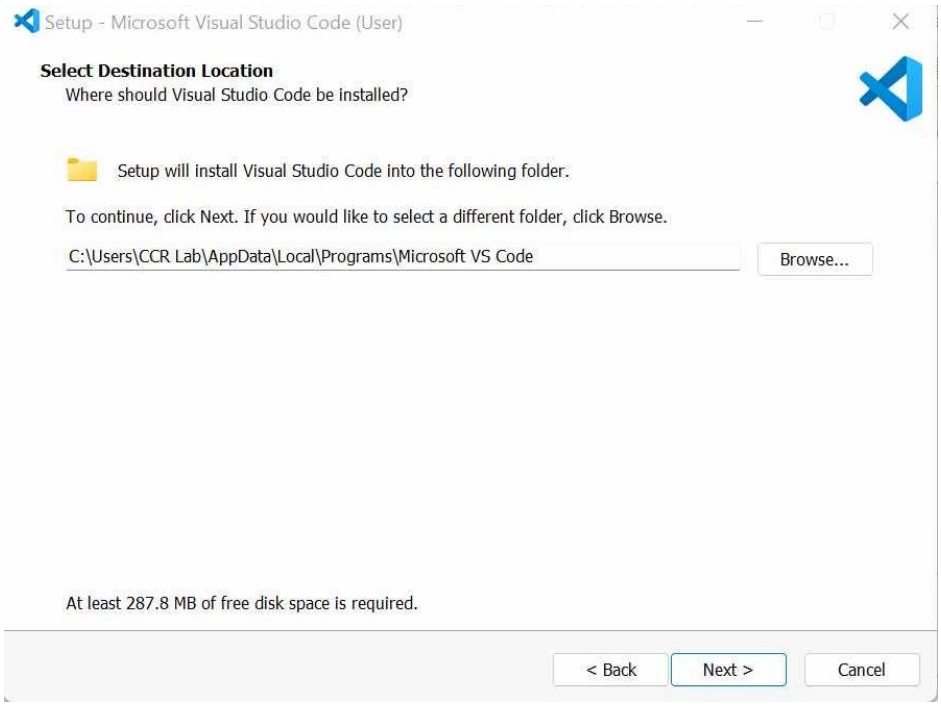
There are the following steps for the installation of Visual Studio code

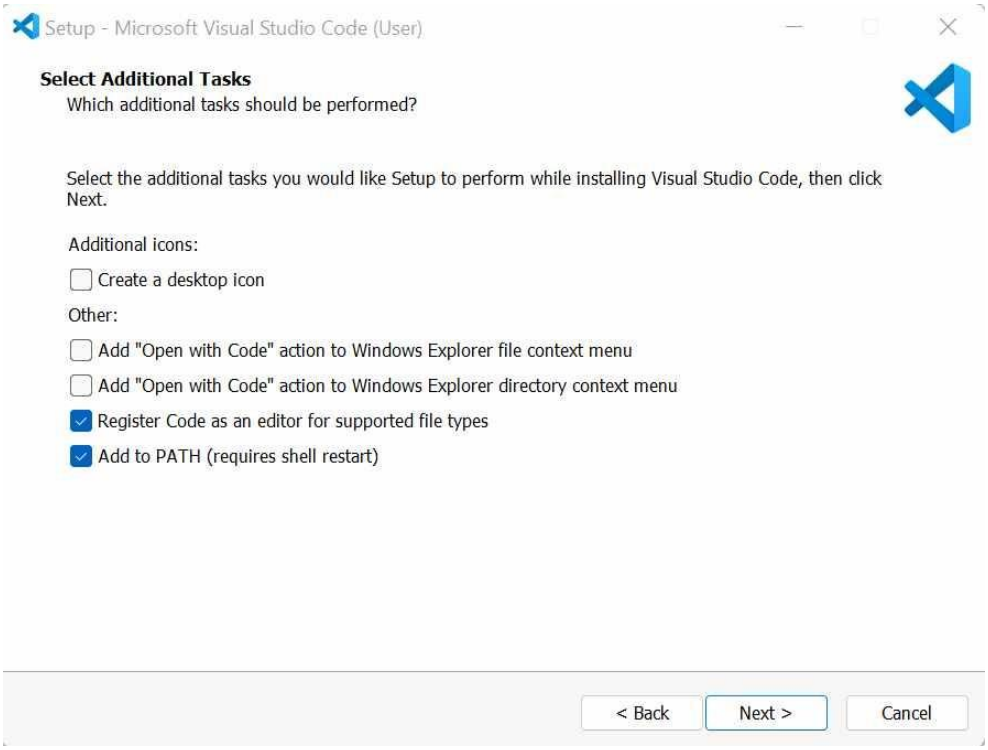
Download the Visual Studio Code installer for Windows/OS/Linux.
https://code.visualstudio.com/download

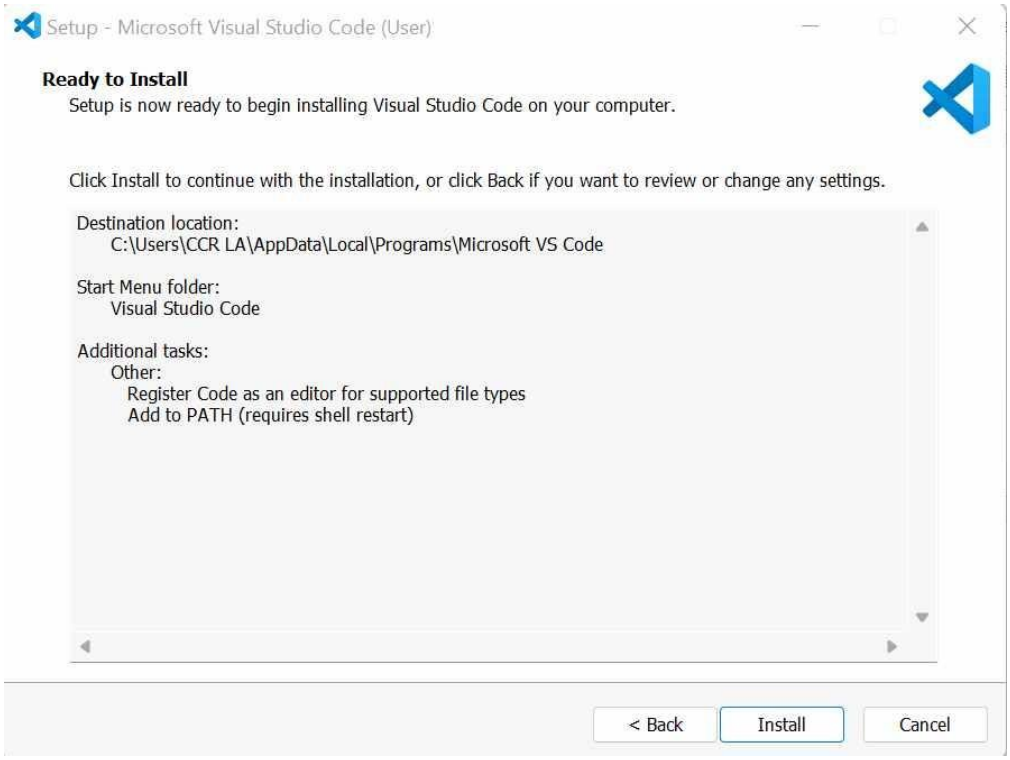
- Once it is downloaded, run the installer (VS Code User Setup-{version}.exe). This will only take a minute.
- By default, VS Code is installed under C:\Users\{Username}\AppData\Local\Programs\Microsoft VS Code.

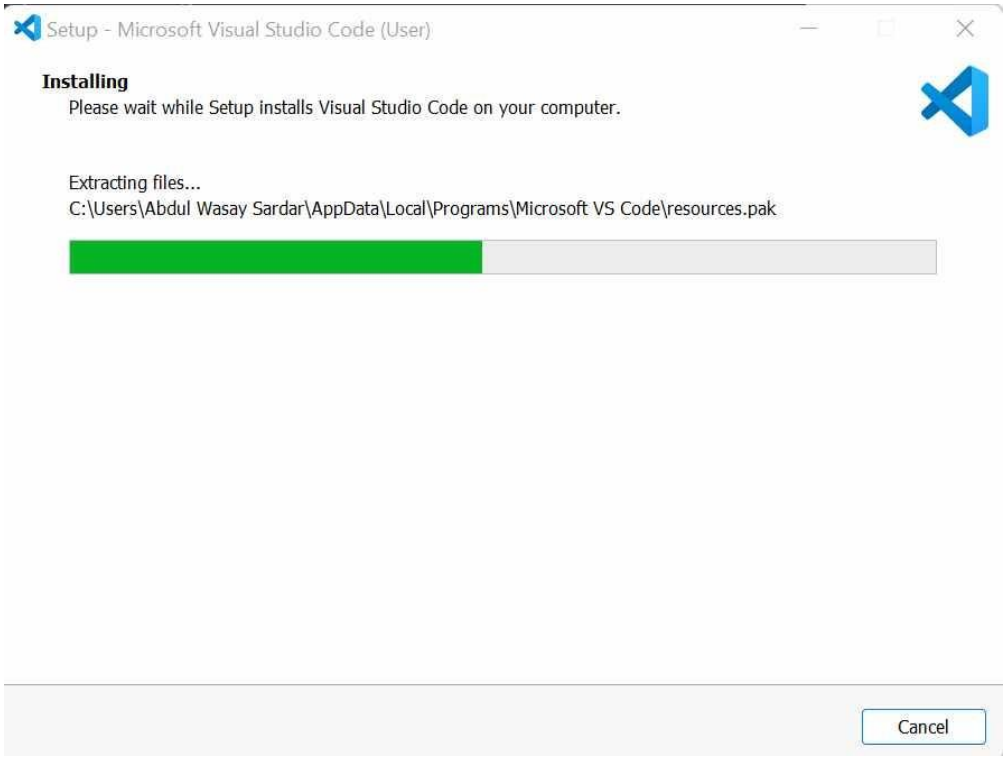


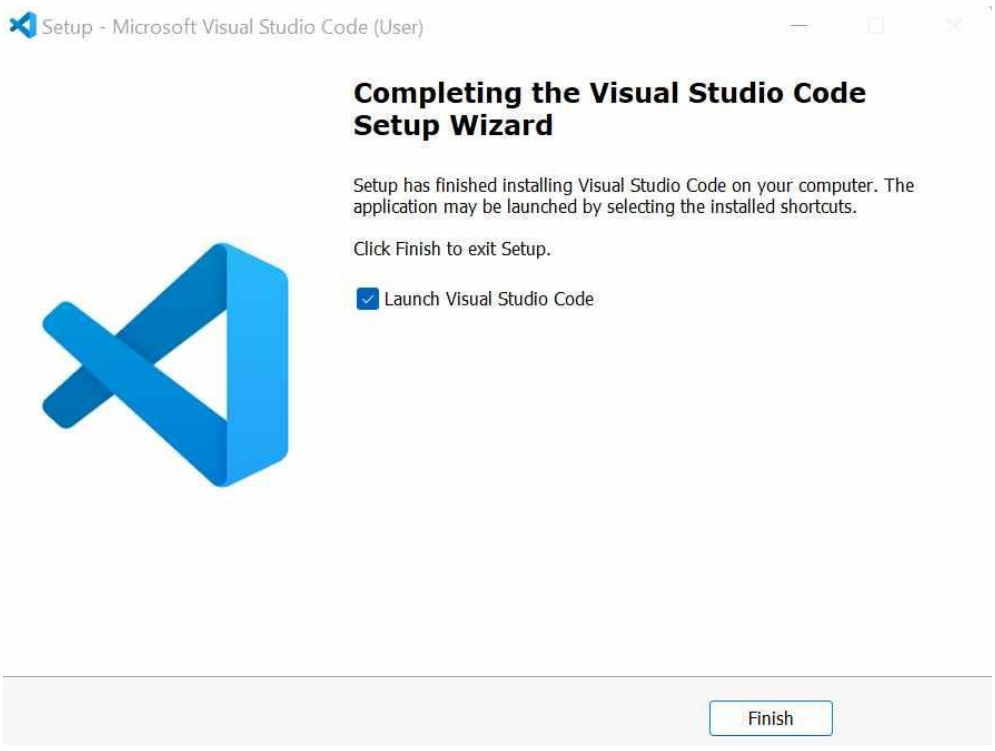

Setup - Microsoft Visual Studio Code (User)
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
Visual Studio Code
Browse...
Don't create a Start Menu folder
< Back
Next >
Cancel


Setup - Microsoft Visual Studio Code (User)
Select Destination Location
Where should Visual Studio Code be installed?
Setup will install Visual Studio Code into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Users\CCR Lab\AppData\Local\Programs\Microsoft VS Code
Browse...
At least 287.8 MB of free disk space is required.
< Back
Next >
Cancel


Setup - Microsoft Visual Studio Code (User)
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing Visual Studio Code, then click Next.
Additional icons:
Create a desktop icon
Other:
Add "Open with Code" action to Windows Explorer file context menu
Add "Open with Code" action to Windows Explorer directory context menu
Register Code as an editor for supported file types
Add to PATH (requires shell restart)
< Back
Next >
Cancel

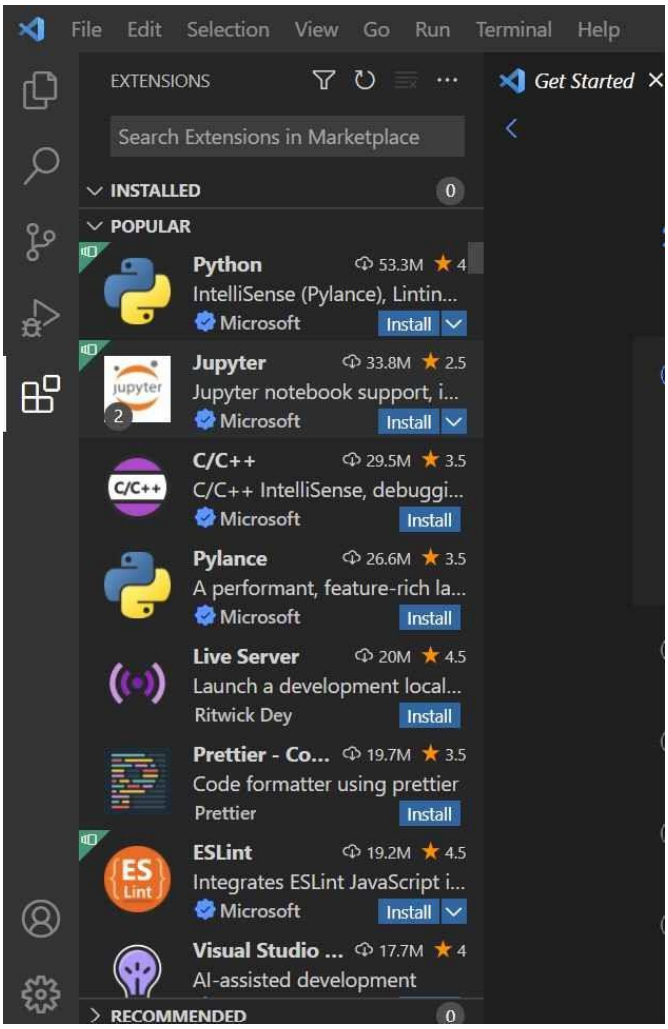

Setup - Microsoft Visual Studio Code (User)
Ready to Install
Setup is now ready to begin installing Visual Studio Code on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\Users\CCR LA\AppData\Local\Programs\Microsoft VS Code
Start Menu folder:
Visual Studio Code
Additional tasks:
Other:
Register Code as an editor for supported file types
Add to PATH (requires shell restart)
< Back
Install
Cancel


Setup - Microsoft Visual Studio Code (User)
Installing
Please wait while Setup installs Visual Studio Code on your computer.
Extracting files...
C:\Users\Abdul Wasay Sardar\AppData\Local\Programs\Microsoft VS Code\resources.pak
Cancel

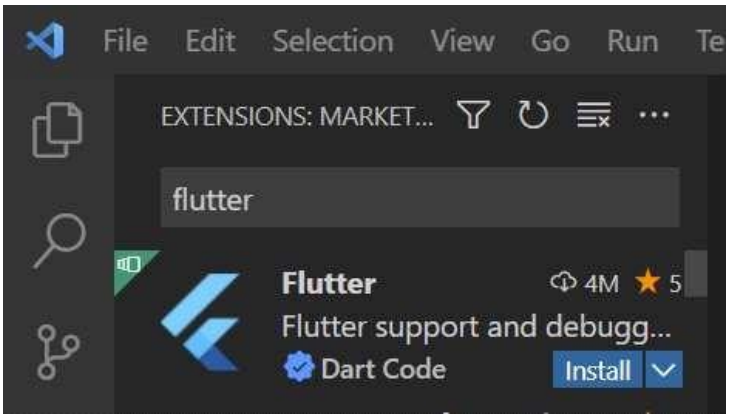

Setup - Microsoft Visual Studio Code (User)
Completing the Visual Studio Code Setup Wizard
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Visual Studio Code
Finish

2.2 Installation of Flutter in visual studio

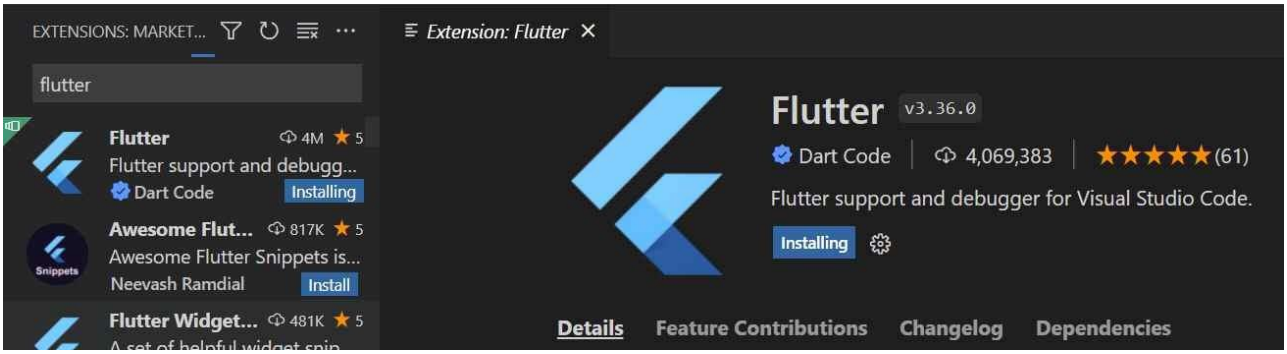
Open visual studio: Press control + shift + x and open the extensions



Search flutter in the extensions



Press install flutter



2.3 Installation of Flutter SDK

Flutter SDK: https://docs.flutter.dev/get-started/install/windows

- Download the following installation bundle to get the latest stable release of the Flutter SDK:
- For other release channels, and older builds, see the SDK releases page.
- Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK (for example, C:\Users\<your-user-name>\Documents).

Update your path

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

- From the Start search bar, enter 'env' and select Edit environment variables for your account.
- Under User variables check if there is an entry called Path:
- If the entry exists, append the full path to flutter\bin using; as a separator from existing values.
- If the entry doesn't exist, create a new user variable named Path with the full path to flutter\bin as its value.
- You must close and reopen any existing console windows for these changes to take effect. Run flutter doctor

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

C:\src\flutter>flutter doctor

This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully for other software you might need to install or further tasks to perform (shown in bold text).

For Example

```
[-] Android toolchain - develop for Android devices
    • Android SDK at D:\Android\sdk
    ✗ Android SDK is missing command line tools; download from https://goo.gl/XxQghQ
    • Try re-installing or updating your Android SDK,
      visit https://docs.flutter.dev/setup/#android-setup for detailed instructions.
```

The following sections describe how to perform these tasks and finish the setup process. Once you have installed any missing dependencies, you can run the flutter doctor command again to verify that you've set everything up correctly.

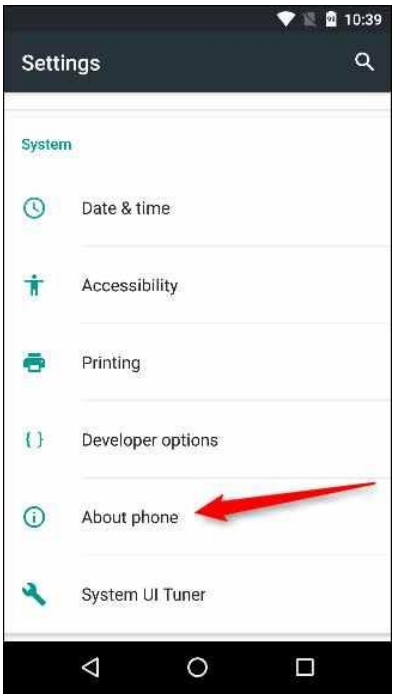
2.4 Installation of Flutter in visual studio

After the setup visual studio, extract EWS app, and import EWS app code in visual studio now we need to run that code. For the running code first, we need to connect mobile phone. For the connecting mobile device with visual studio, we need to set mobile phone in debugger mode. There are the following steps for the mobile setting in debugging mode.

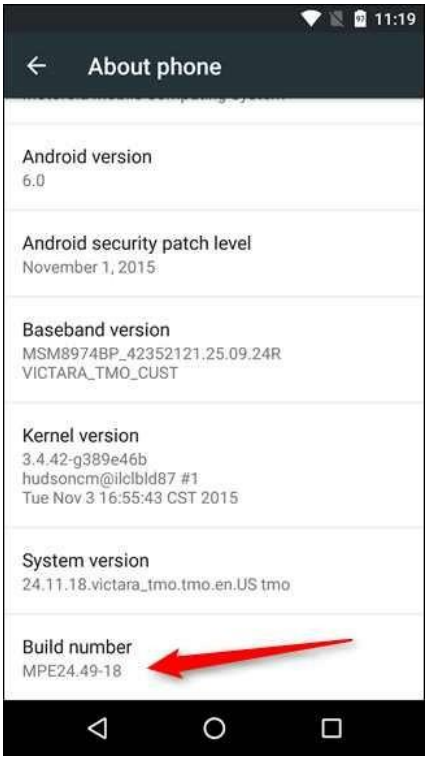
- Go to Settings > About phone and tap Build Number repeatedly until developer mode is enabled
- Go into Settings > Developer options and turn USB debugging ON
- Connect you phone to your PC via USB. We had to plug our phone into our PC's on-board USB port, not our USB hub as this did not work
- You should be prompted to trust your PC (trust it)
- Fire up Visual Studio
- Open your solution and your device should now be available.

Side note: we have noticed that sometimes it also depends on the cable you use. If you still have no luck, perhaps try another cable.

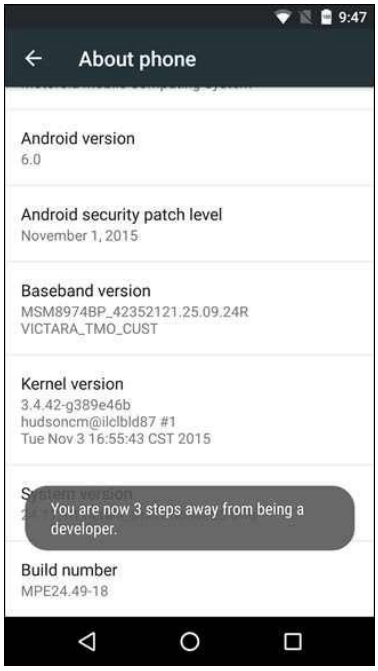
To enable Developer Options, open the Settings screen, scroll down to the bottom, and tap About phone or About tablet.



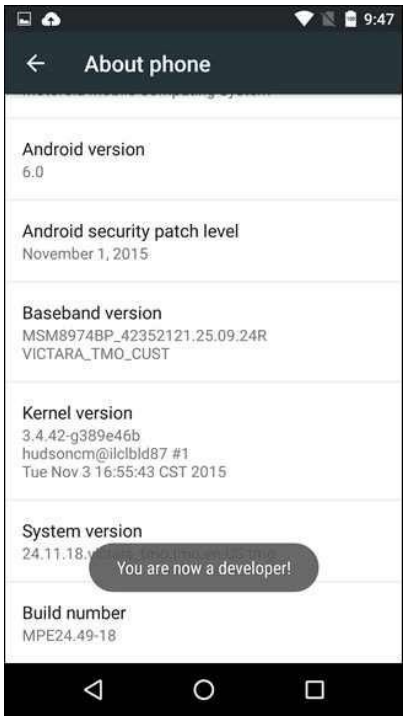
Scroll down to the bottom of the About screen and find the Build number.



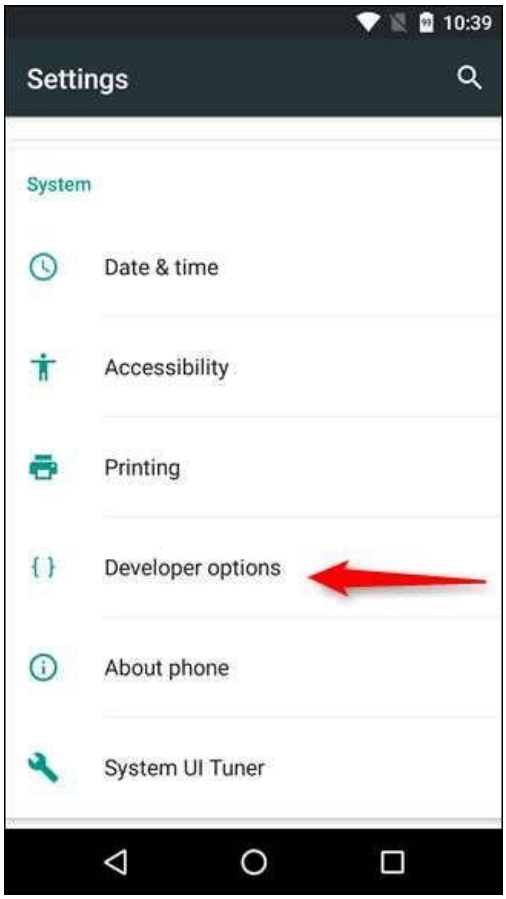
Tap the Build number field seven times to enable Developer Options. Tap a few times and you'll see a toast notification with a countdown that reads "You are now X steps way from being a developer."



When you're done, you'll see the message "You are now a developer!". Congratulations. Don't let this newfound power go to your head.

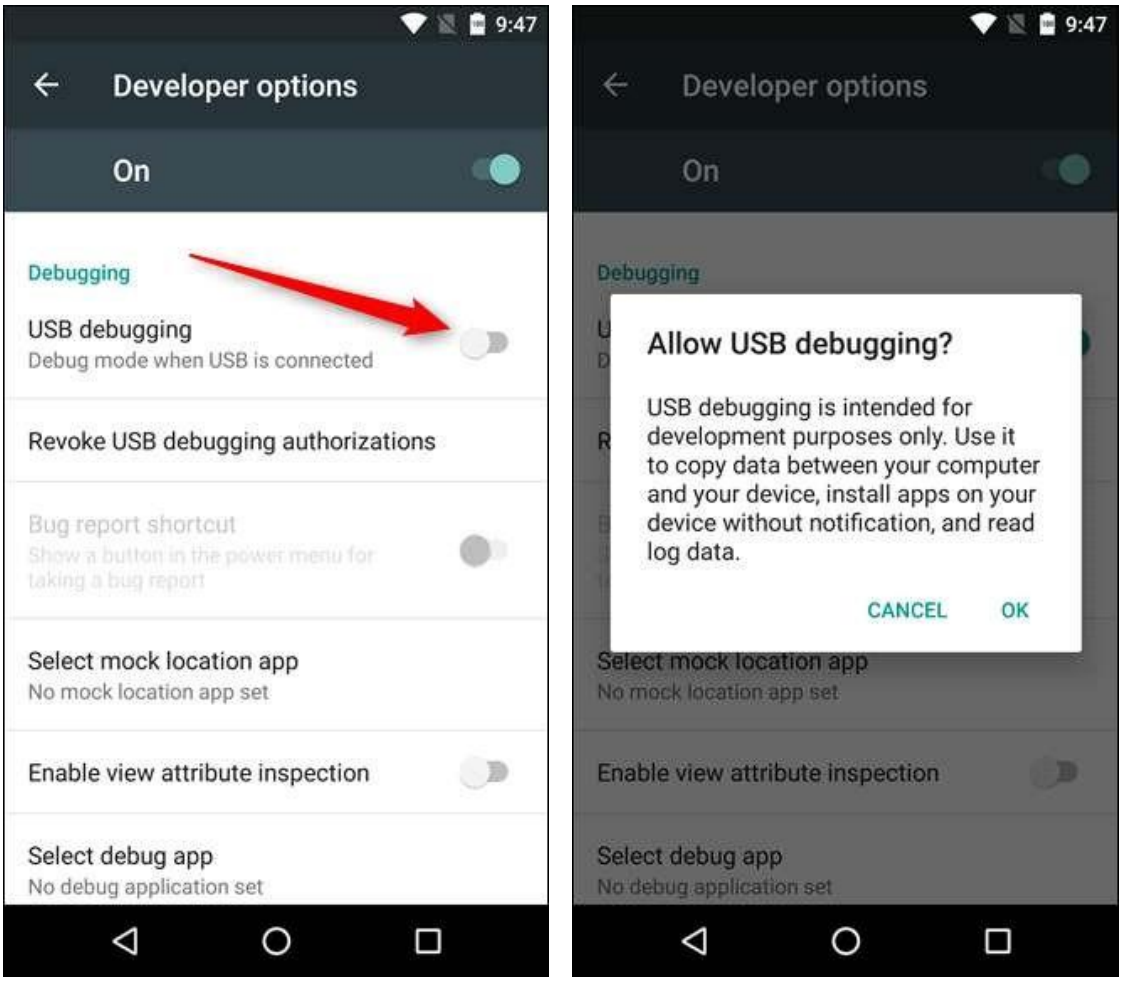


Tap the Back button and you'll see the Developer options menu just above the "About Phone" section in Settings. This menu is now enabled on your device—you won't have to repeat this process again unless you perform a factory reset.

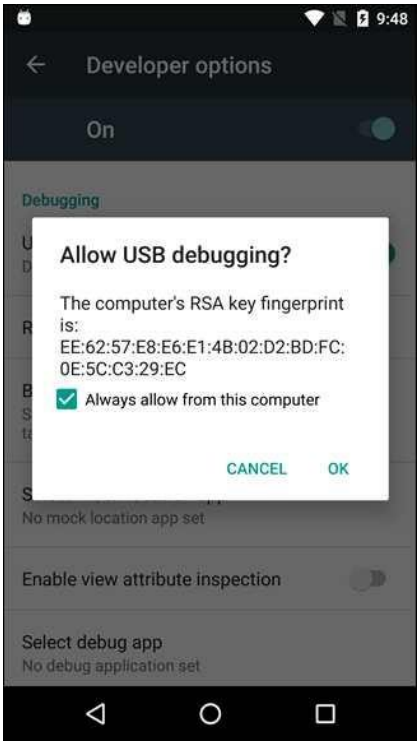


How to Enable USB Debugging

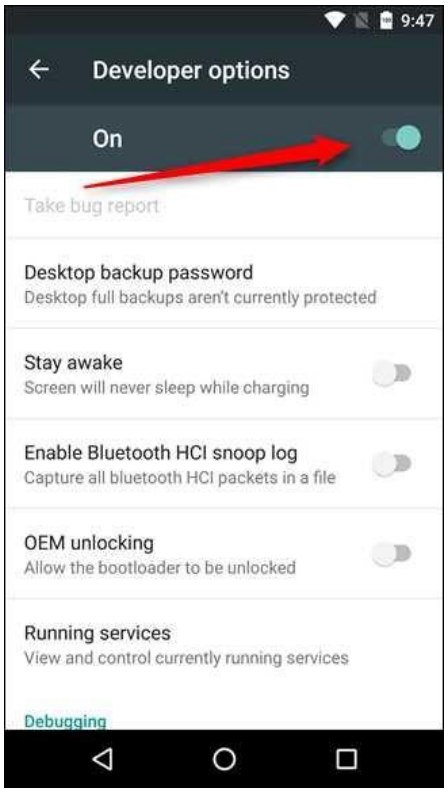
To enable USB Debugging, you'll need to jump into the Developer options menu, croll down to the Debugging section, and toggle the "USB Debugging" slider.



Once upon a time, USB Debugging was thought to be a security risk if left on all the time. Google has done a few things that make that less of an issue now, because debugging requests have to be granted on the phone—when you plug the device into an unfamiliar PC, it will prompt you to allow USB debugging (as seen in the screenshot below).

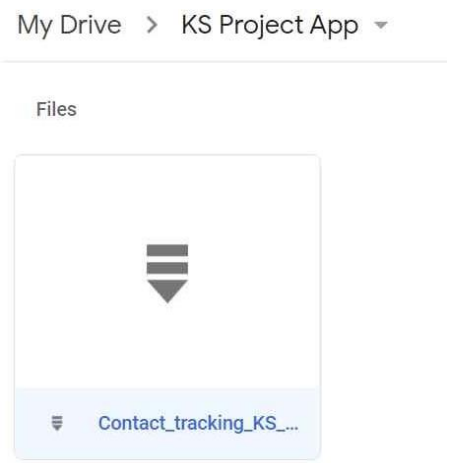


If you still want to disable USB debugging and other developer options when you don't need them, slide the switch at the top of the screen. Easy peasy.

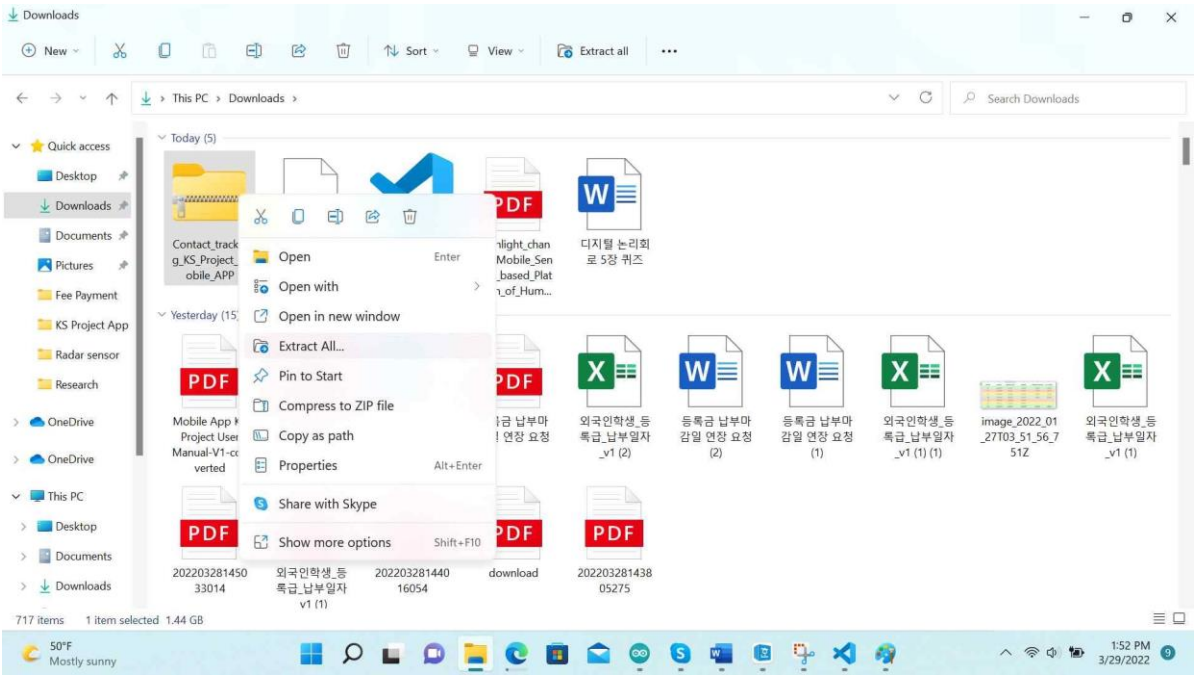


2.5 Download PWS App code

https://drive.google.com/drive/u/1/folders/1ZUmKBbXWvtk7xm57niwS8O4r1EfTPFsN



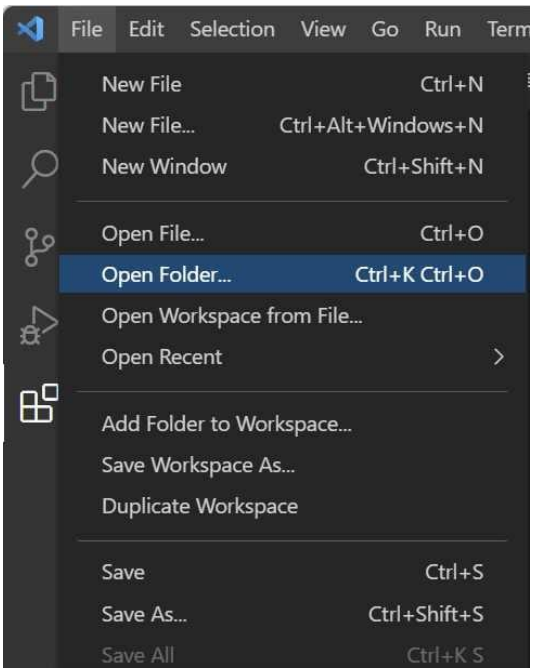
Extract file Contact_tracking_KS_Project_Mobile_APP



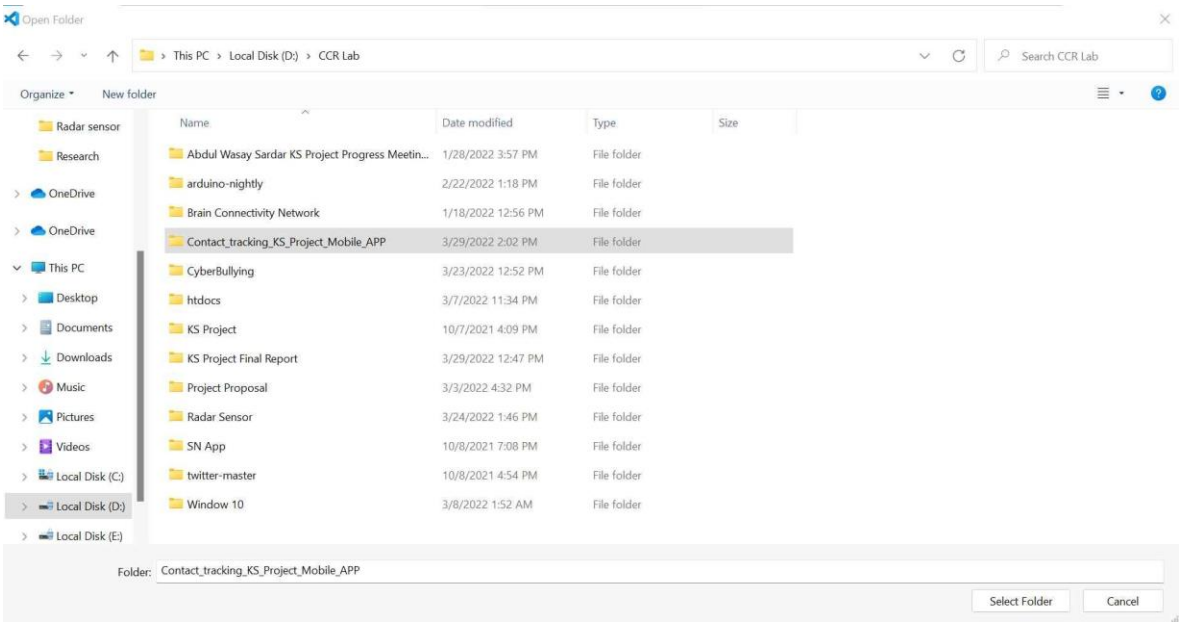
2.6 Importing PWS App in Visual Studio

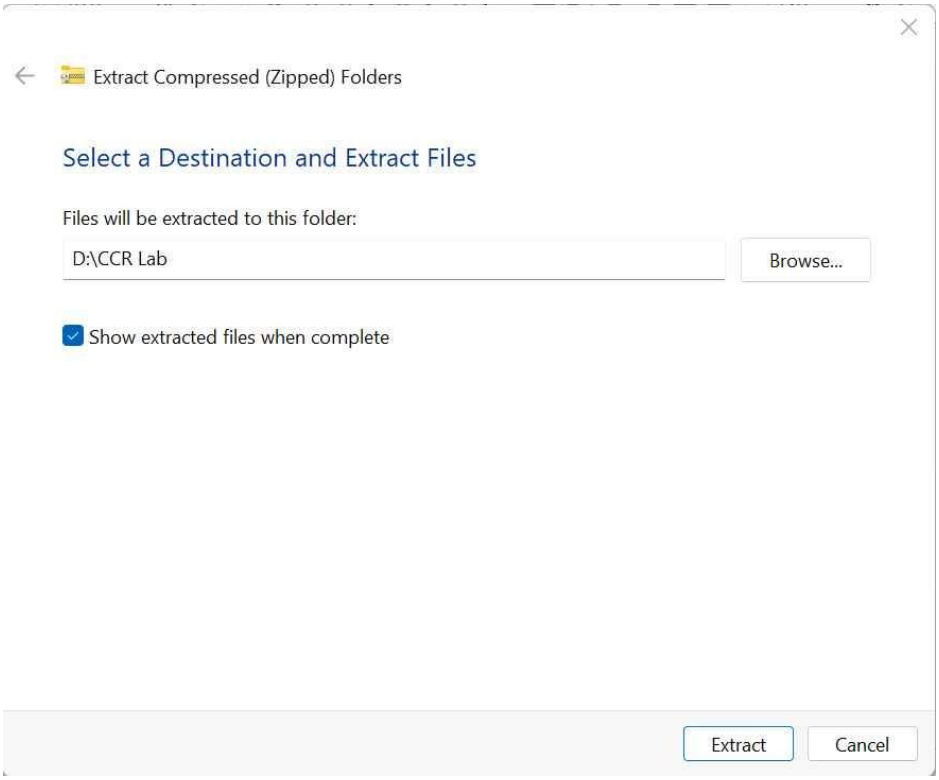
After extraction of file Contact_tracking_KS_Project_Mobile_APP, go to the visual

studio->file->open folder

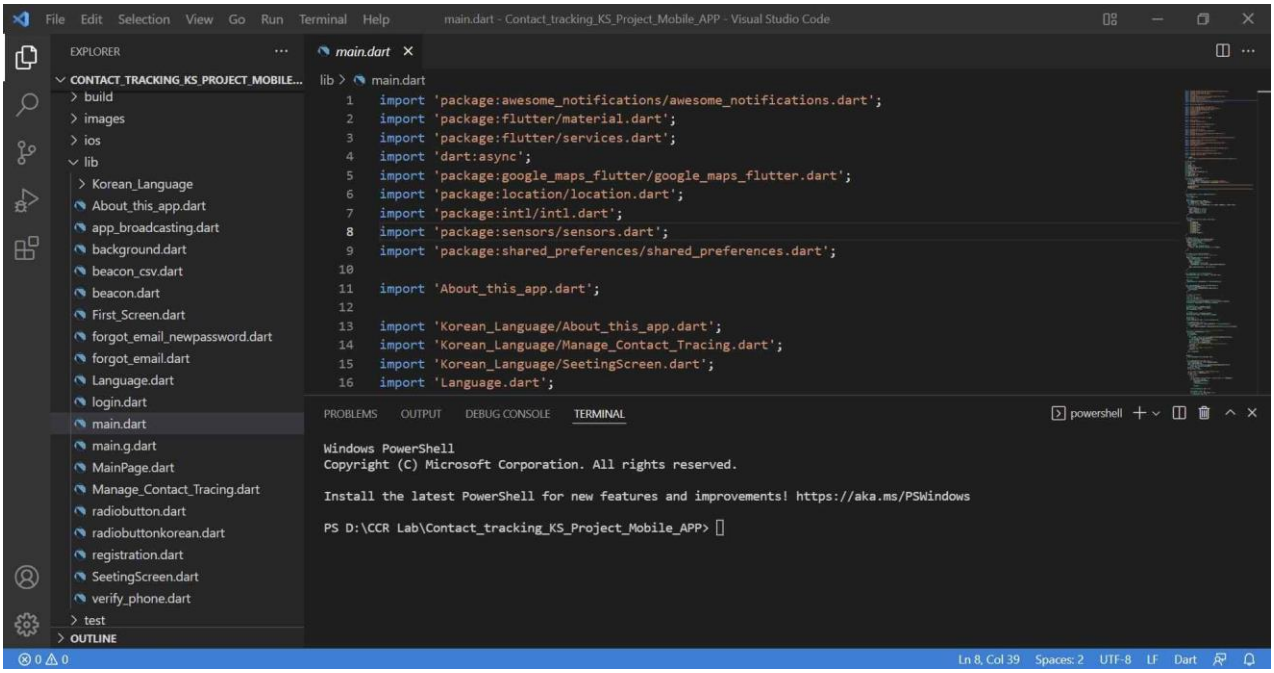


Select the folder Contact_tracking_KS_Project_Mobile_APP





After selection of folder open the lib-> main.dart file (visual studio)

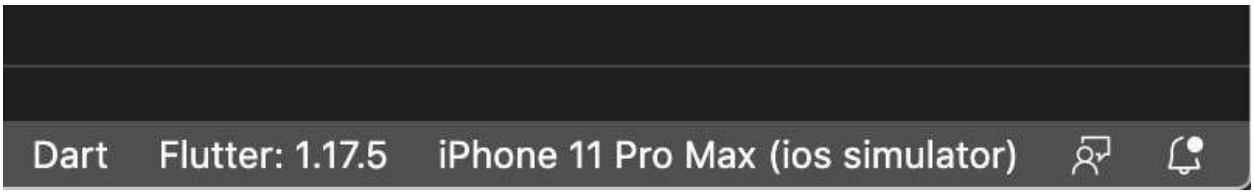


2.7 Running and debugging PWS App

Start debugging by clicking Run > Start Debugging from the main IDE window, or press F5.

Selecting a target device

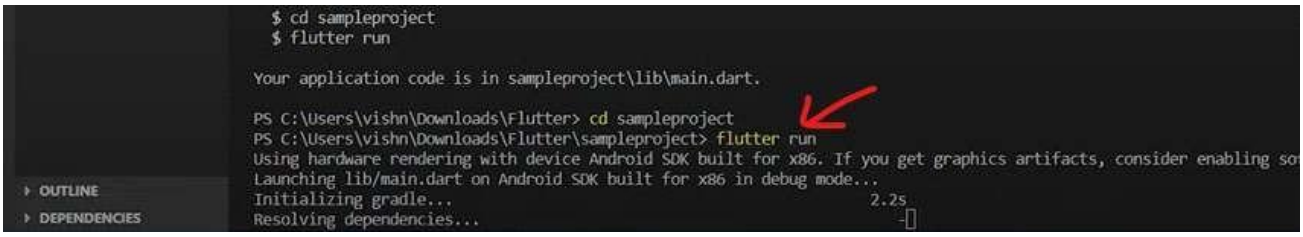
When a Flutter project is open in VS Code, you should see a set of Flutter specific entries in the status bar, including a Flutter SDK version and a device name (or the message No Devices):



The Flutter extension automatically selects the last device connected. However, if you have multiple devices/simulators connected, click device in the status bar to see a pick-list at the top of the screen. Select the device you want to use for running or debugging.

Running the project

After choosing a suitable device, and connecting that device to PC, the flutter project can be be complied by running "flutter



Access Point (AP)

3.1 Raspberry-PI Configuration

For the Access point development, we used raspberry-pi. First, we need to install the operating system in SD card. For the development of access point, we need raspberry-pi, SD card, Bluetooth module, and WIFI module. Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries.

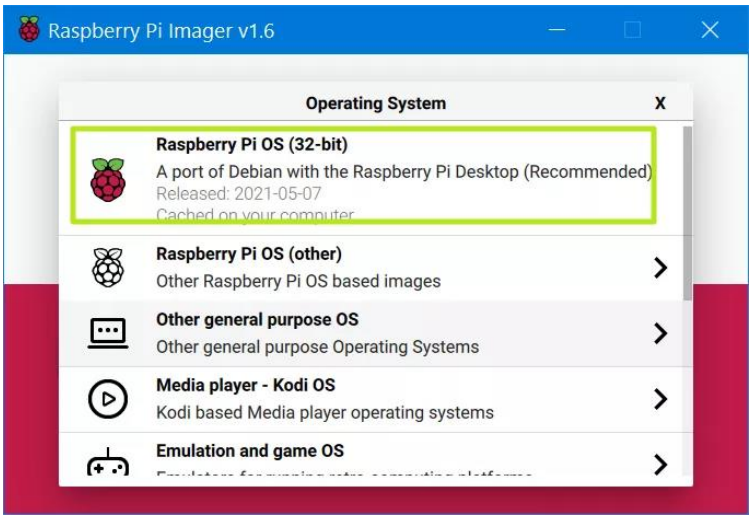
3.2 Downloading and Installing Raspberry Pi OS

Once you have all the components you need, use the following steps to set up your Raspberry Pi using a Windows, Mac, or Linux-based PC (we tried this on Windows, but it should be the same on all three).

1. Insert a microSD card / reader into your computer.

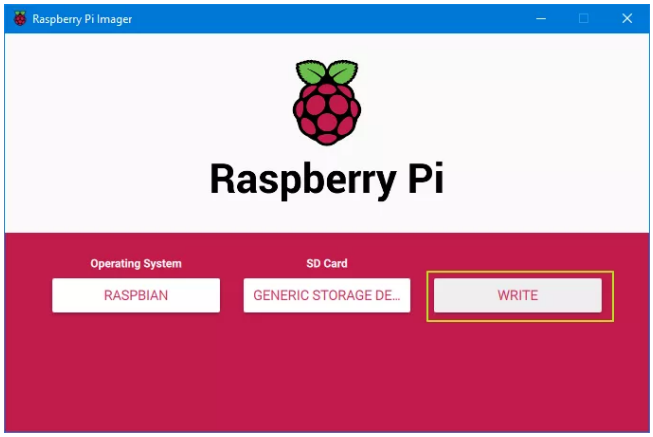
2. Download and install the official Raspberry Pi Imager. Available for Windows, macOS or Linux, this app will both download and install the latest Raspberry Pi OS. There are other ways to do this, namely by downloading a Raspberry Pi OS image file and then using a third-party app to "burn it," but the Imager makes it easier.

3. Click Choose OS and select Raspberry Pi OS (32-bit) from the OS menu (there are other choices, but for most uses, 32-bit is the best).



4. Click Choose SD card and pick the one you're using.

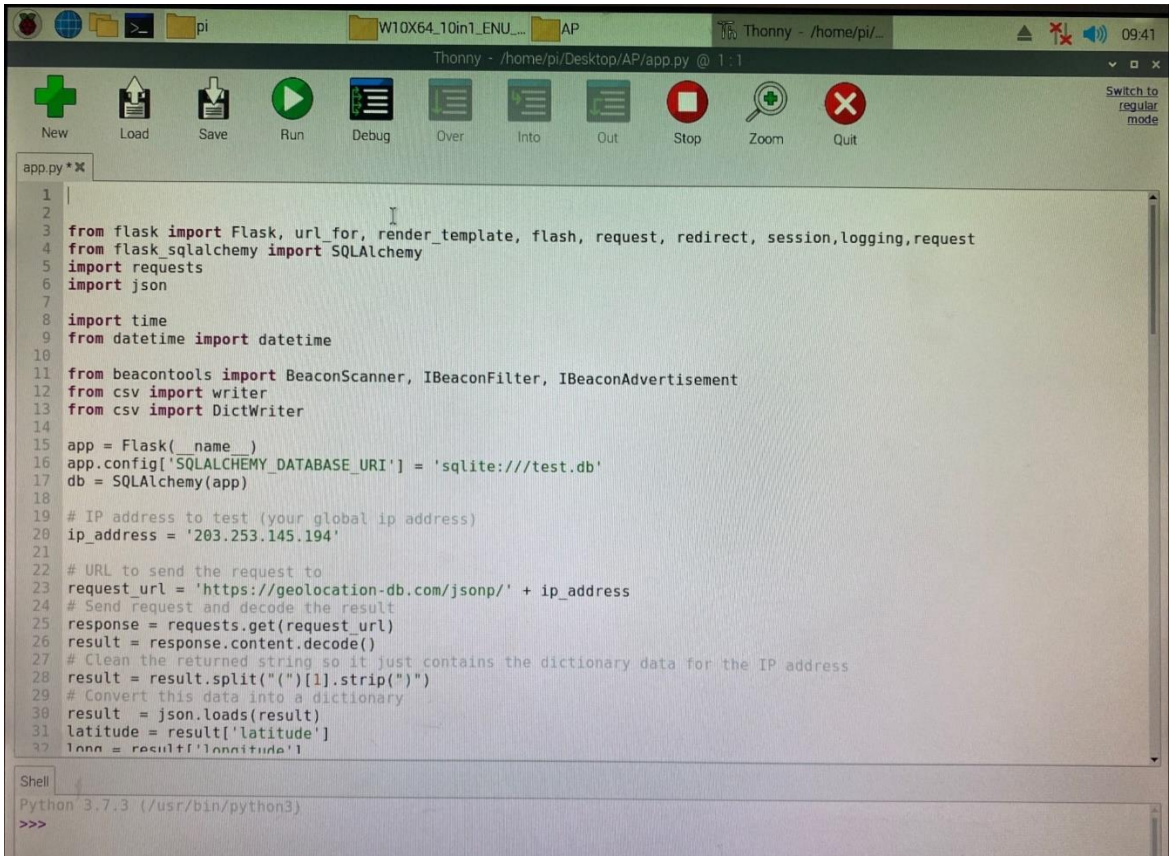
5. Click Write. The app will now take a few minutes to download the OS and write to your card.



3.3 Run python code

After installation of Raspberry-pi operating system, just need to run the python code that provided in the link: [KS Project App - Google Drive]

After the running of code automatically, contacts are collected by the access and point and send to the server. Initially contacts are stored in csv file locally at access point (raspberry-pi), when hospital needs contact, they send request to the access point and automatically access point send the data into server.



Hospital

4.1 GUI development for the Hospital Webpage

We develop the GUI for the hospital portal for the storing of covid-19 patients' status, infected, recover or dead. For that we make the basic GUI and, we implement API on the server that are explain on server. Hospital portal is needed for the storing record of the COVID-19 infected and recover persons. Hospital portal contains admin portal (For the authorizing of each hospital), hospital registration, hospital login, and user status updating. The figure 7 shows the GUI of the hospital portal.

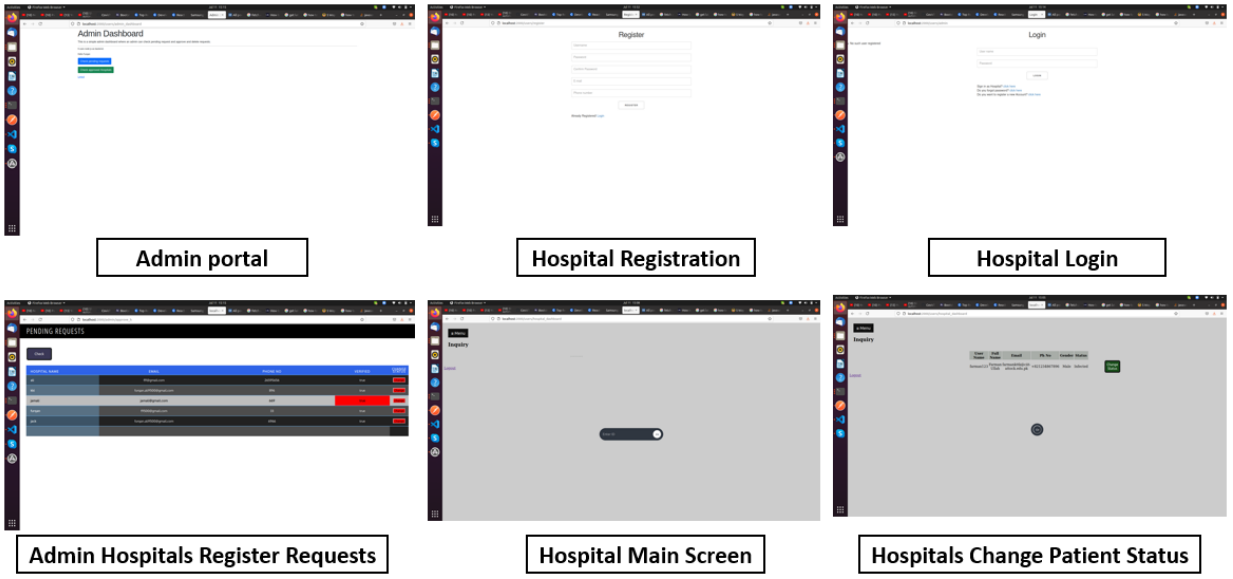


Figure.: Patients report collection hospital portal

The entity relationship diagram (ERD Diagram) for patient's status changing (infected, recover) by hospital

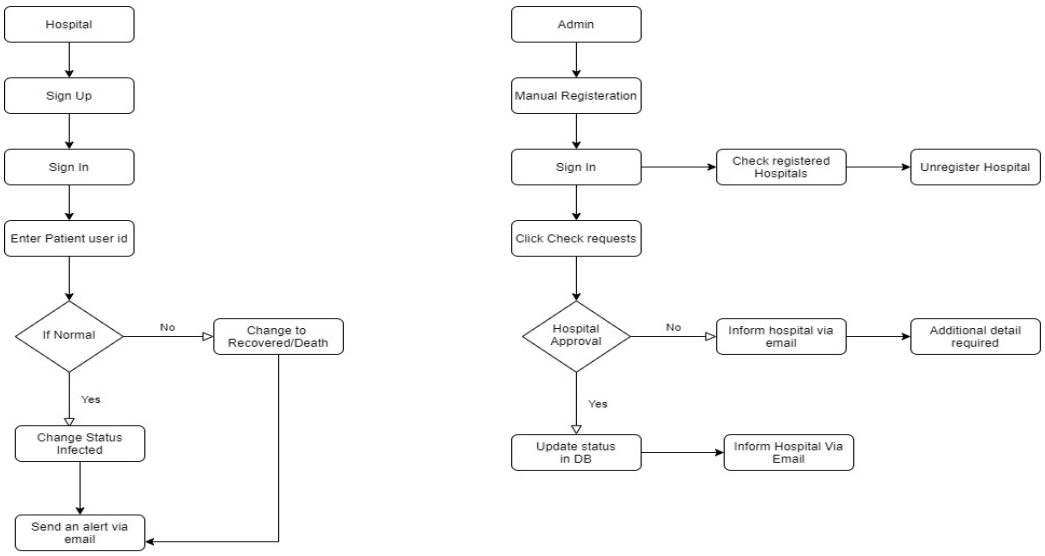


Figure 9 shows the hospital and admin portal workflow for the patients COVID-19 reports collection.

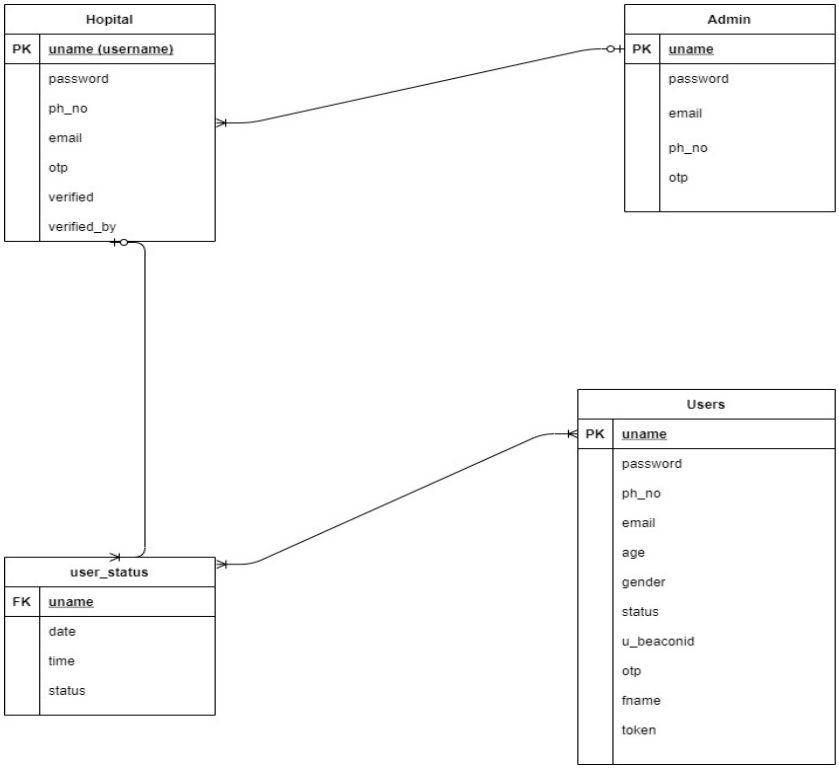


Figure: Entity relationship (ERD) diagram for patients' status changing

Server

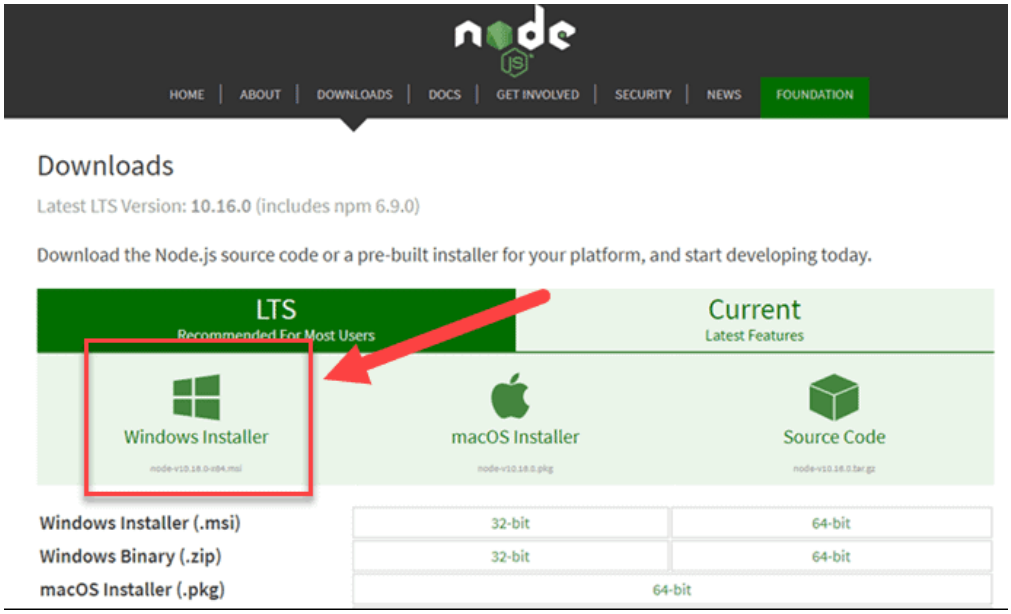
5.1 Server for the Pandemic Watch System (PWS)

Pandemic watch system (PWS) data is stored in the PostgreSQL server. PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance. It was originally named POSTGRES, referring to its origins as a successor to the Ingres database developed at the University of California, Berkeley. PostgreSQL is a powerful, open-source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

5.2 Installation of Node.js

Step 1: Download Node.js Installer

In a web browser, navigate to https://nodejs.org/en/download/. Click the **Windows Installer** button to download the latest default version. At the time this article was written, version 10.16.0-x64 was the latest version. The Node.js installer includes the NPM package manager.



Step 2: Install Node.js and NPM from Browser

1. Once the installer finishes downloading, launch it. Open the **downloads** link in your browser and click the file. Or browse to the location where you have saved the file and double-click it to launch.

2. The system will ask if you want to run the software – click **Run**.

3. You will be welcomed to the Node.js Setup Wizard – click **Next**.

4. On the next screen, review the license agreement. Click **Next** if you agree to the terms and install the software.

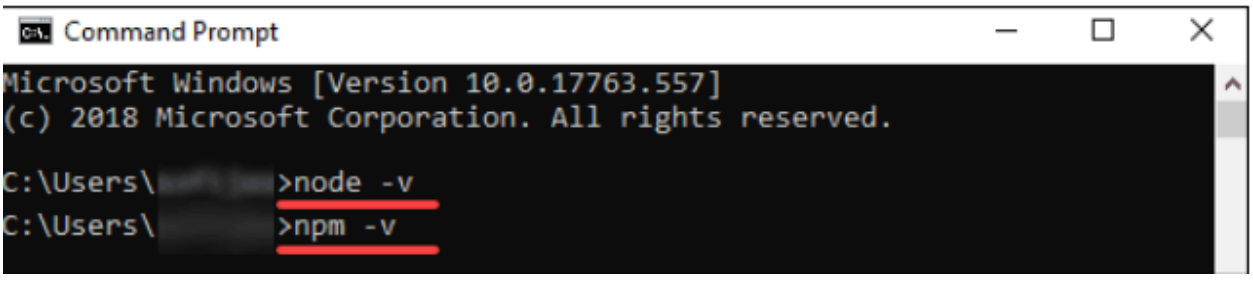
5. The installer will prompt you for the installation location. Leave the default location, unless you have a specific need to install it somewhere else – then click **Next**.

6. The wizard will let you select components to include or remove from the installation. Again, unless you have a specific need, accept the defaults by clicking **Next**.

7. Finally, click the **Install** button to run the installer. When it finishes, click **Finish**.

Step 3: Verify Installation

Open a command prompt (or PowerShell), and enter the following commands:



Step 4: Validate by a Random Project

Node.js is a framework, which means that it doesn't work as a normal application. Instead, it interprets commands that you write. To test your new Node.js installation, create a *Hello World* script.

1. Start by launching a text editor of your choice.

2. Next, copy and paste the following into the text editor you've just opened:

```
var http = require('http');
 http.createServer(function (req, res) {
   res.writeHead(200, {'Content-Type': 'text/html'});
   res.end('Hello World!');
 }).listen(8080);
```

3. Save the file, then exit. Open the PowerShell, and enter the following:

```
node \users\<your_username>\myprogram.js
```

It will look like nothing has happened. In reality, your script is running in the background. You may see a Windows Defender notice about allowing traffic – for now, click **Allow**.

4. Next, open a web browser, and enter the following into the address bar:

```
http://localhost:8080
```

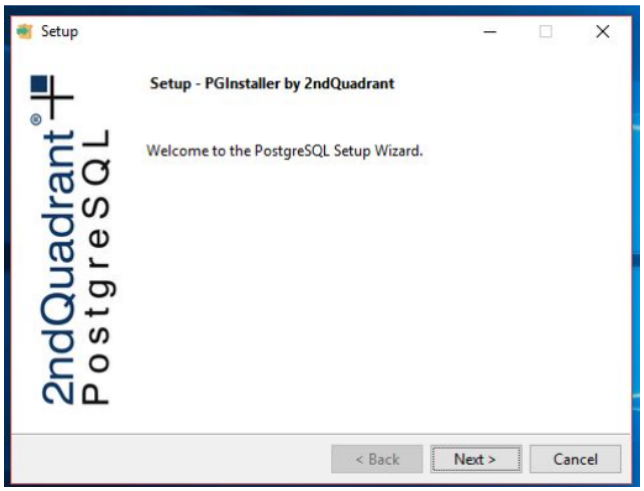
In the very upper-left corner, you should see the text ***Hello World!***

Right now, your computer is acting like a server. Any other computer that tries to access your system on port 8080 will see the *Hello World* notice. To turn off the program, switch back to PowerShell and press **Ctrl+C**. The system will switch back to a command prompt. You can close this window whenever you are ready.

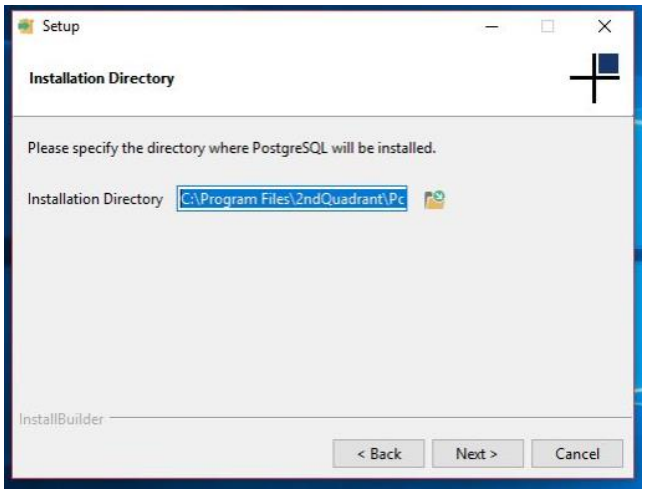
5.3 PostgreSQL Installation

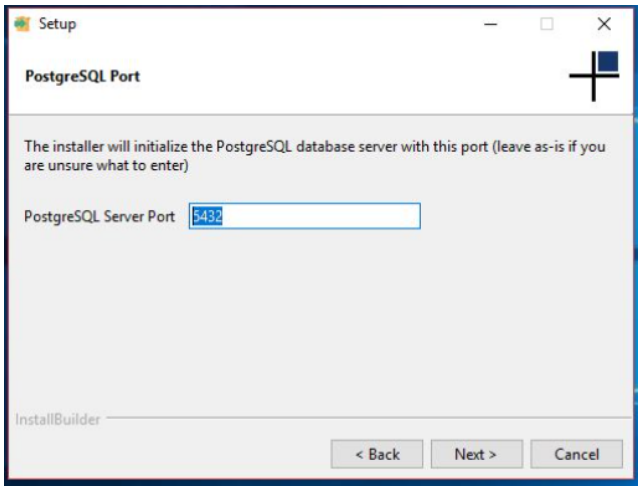
1. Download Postgres Installer from here: **https://www.2ndquadrant.com/en/**. Click on the executable file to run the installer. Select your preferred language.

2. Specify directory where you want to install PostgreSQL.

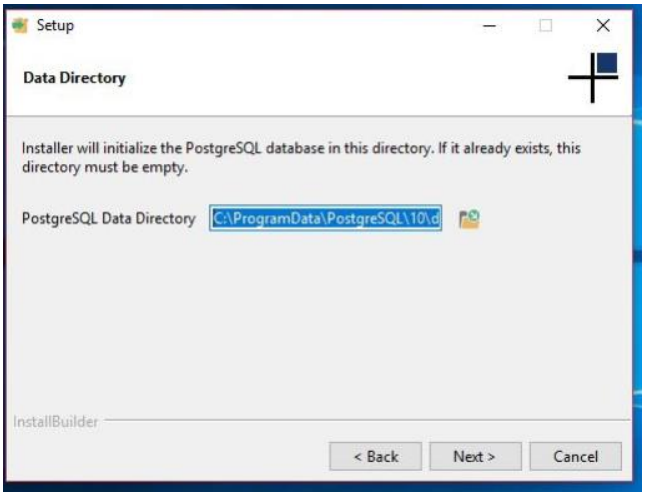


3. Specify PostgreSQL server port. You can leave this as default if you're unsure what to enter

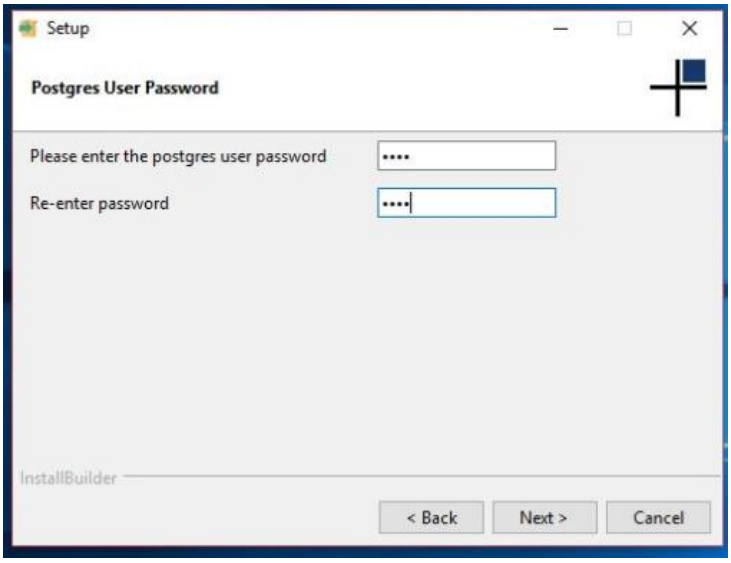




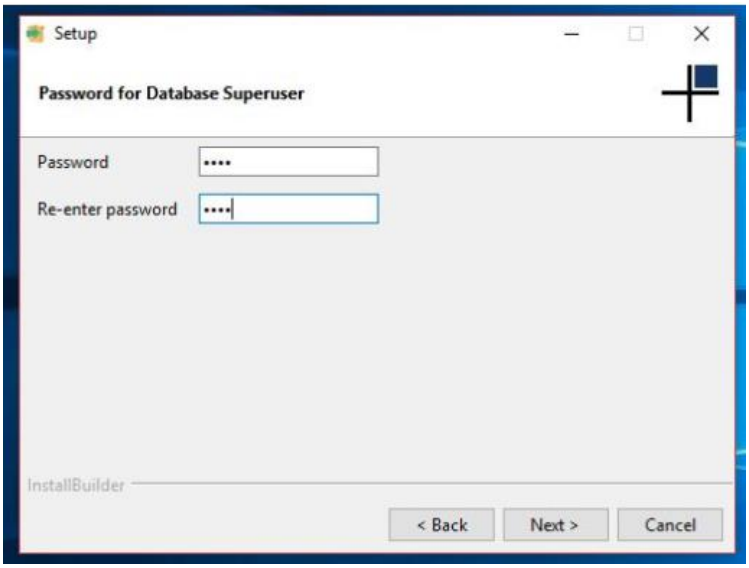
4. Specify data directory to initialize PostgreSQL database.



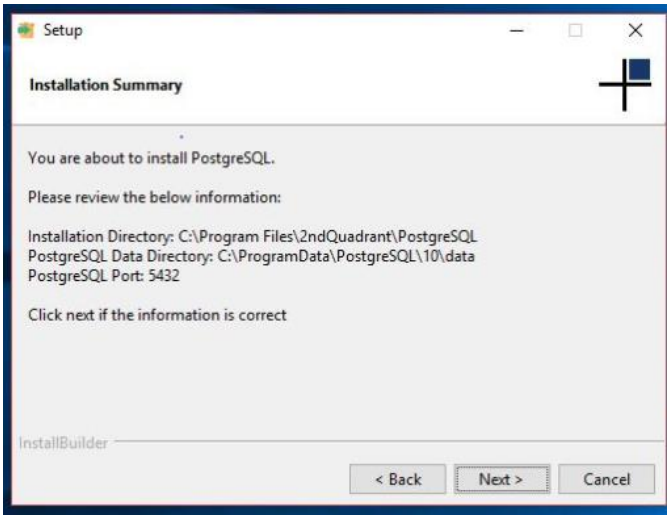
5. Create a PostgreSQL user password.

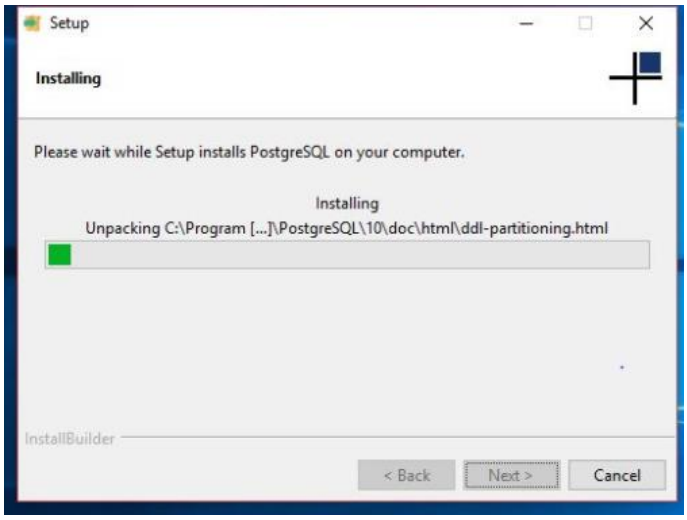


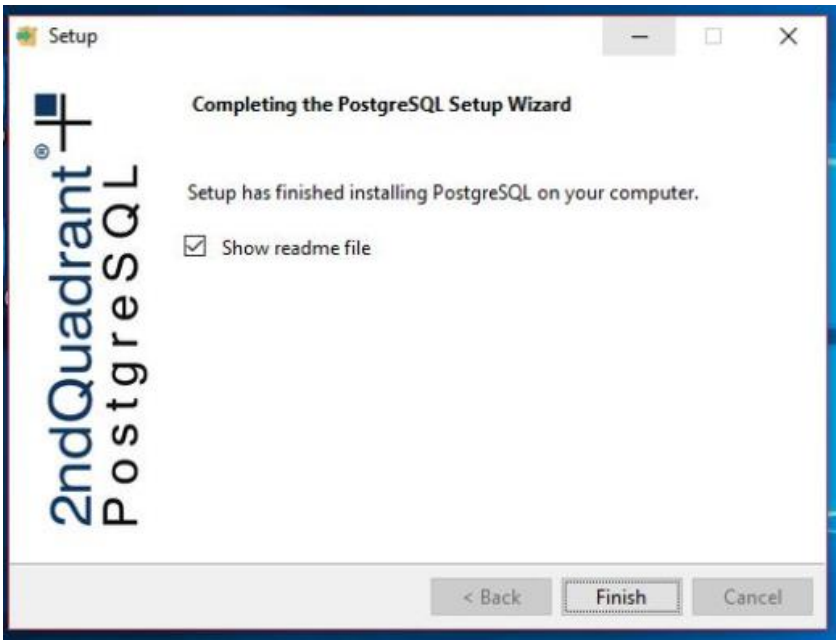
6. Create password for database Superuser.



7. Click next to begin PostgreSQL installation.





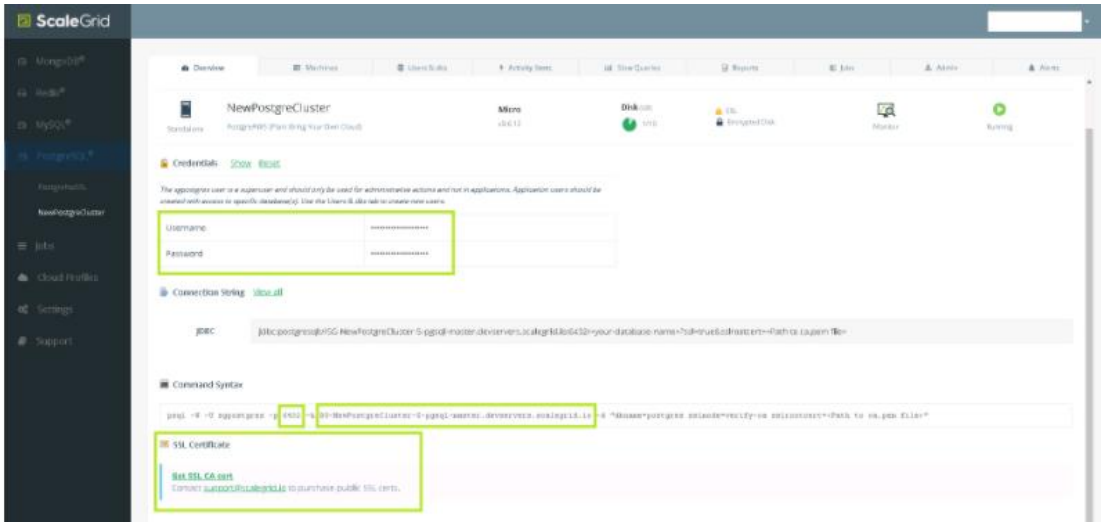


5.4 Nodejs-PostgreSQL Connection

1. After installing both Nodejs and PostgreSQL, get the following PostgreSQL deployment information from your cluster details page:
   a. Username
   b. Password
   c. Hostname

d. Port (default 5432)

e. SSL certificate (if deployment has SSL)



2. Run the following code on Nodejs to connect to you PostgreSQL deployment:

Deployment without SSL

```
const { Client } = require('pg')
const client = new Client({
  user: 'sgpostgres',
  host: 'SG-PostgreNoSSL-14-pgsql-master.devservers.scalegrid.io',
  database: 'postgres',
  password: 'password',
  port: 5432,
})
client.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});
```

Deployment with SSL

```
const { Client } = require('pg');
var fs = require('fs');

const client = new Client({
  user: 'sgpostgres',
  host: 'SG-NewPostgreCluster-5-pgsql-master.devservers.scalegrid.io',
  database: 'postgres',
  password: 'password'',
  port: 6432,
  ssl  : {
    ca : fs.readFileSync('<path to CA cert file>')
  }
})
client.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});
```

5.5 Operational Flow

Download the Nodejs project from the following link:

https://github.com/CovidTrackerApp/nodeJs_apis

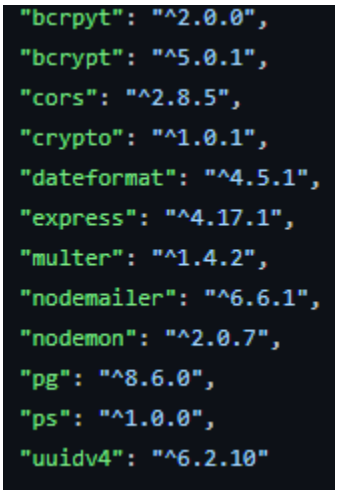
After downloading and opening the project, run "npm init --yes" command.

Module Installation:

Installing modules from npm is one of the most basic things you should learn to do when getting started with npm. As you dive deeper, you'll begin to learn some variations on installing modules, but here's the very core of what you need to know to install a standalone module into the current directory:

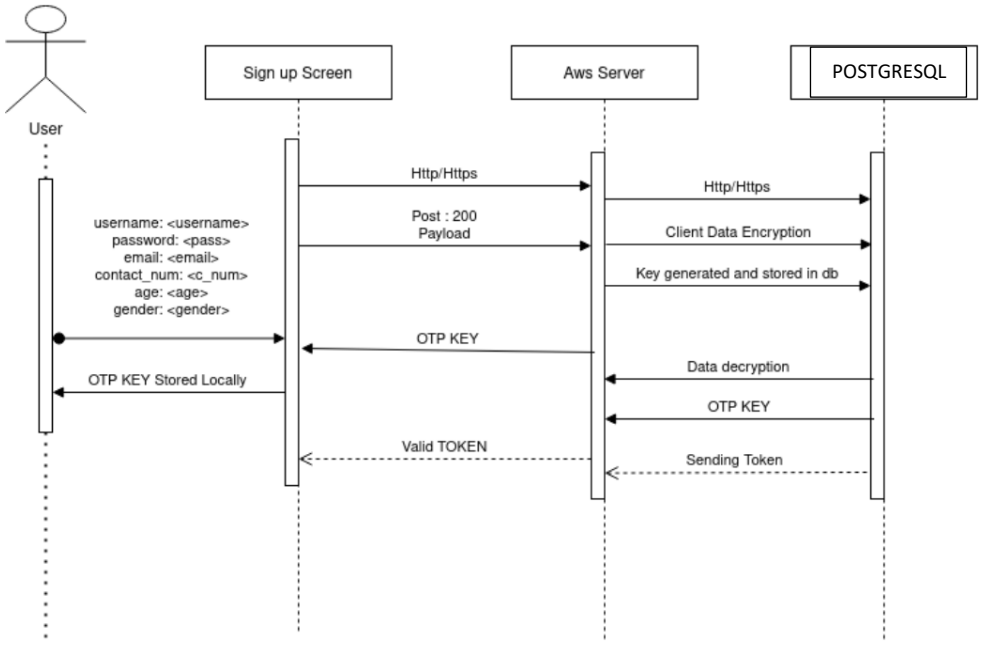
```
npm install <module>
```

Where "<module>" stands for name of the module to be installed. For PWS project, following modules are to be installed.

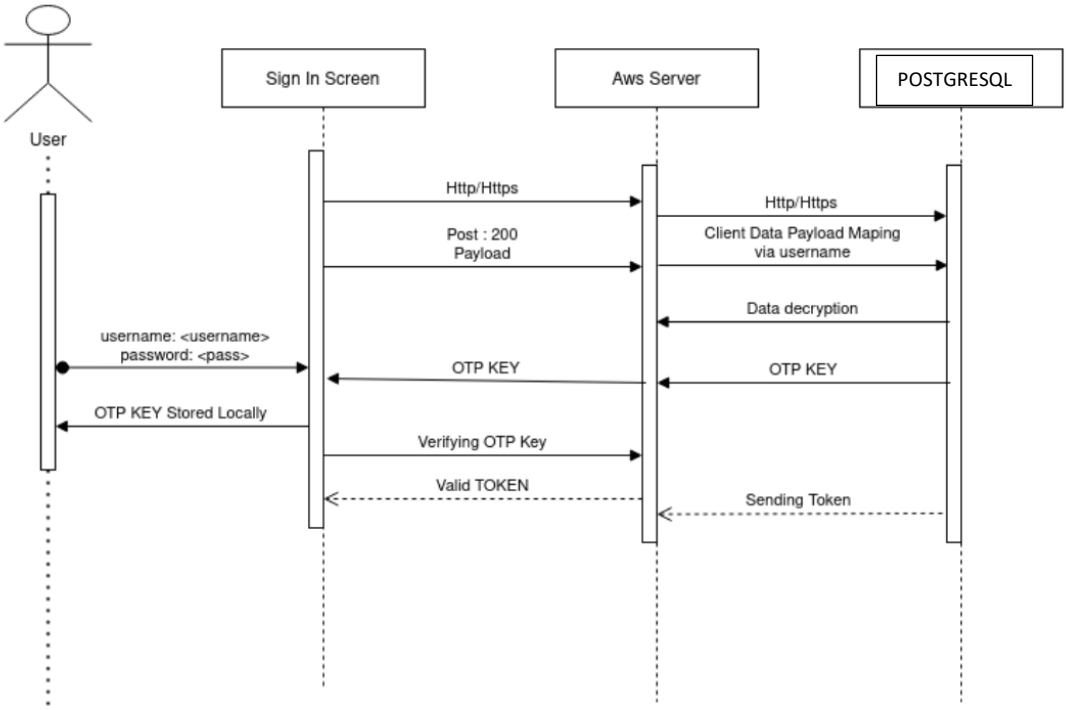


After initialization of project, following operations can be realized by the server.

5.6 Registration Sequence diagram



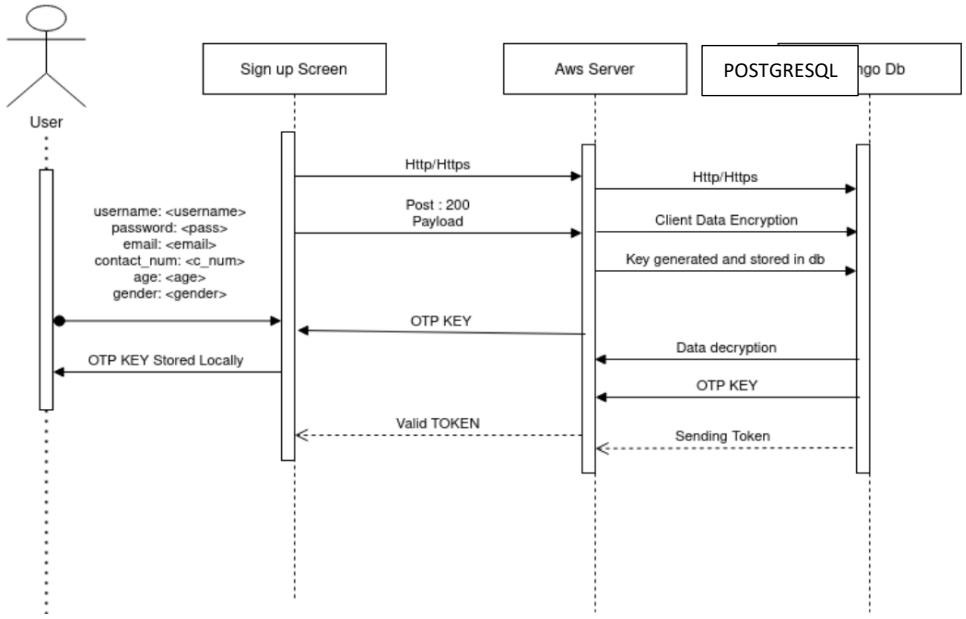
5.7 Sign in Sequence diagram



5.8 CSV file reception from client

A single CSV will be posted to server per day. In order to store the CSV file from server to POSTGRE SQL database, following steps are to be followed:

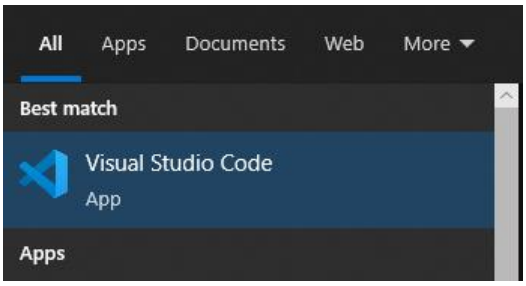
- First give write access to PostgreSQL dB table that you are accessing through pg. admin then carry out these steps
- If the file and the PostgreSQL database are on the same machine, then the path and/or name of the file are not correct.
- If the file is on your local machine and the database is on another, you cannot use the COPY command in SQL. You have two main choices to make this work:
- Use psql \copy from your local machine. The syntax is similar, but it will transfer from your local to the remote. The docs are pretty helpful: https://www.postgresql.org/docs/9.5/static/app-psql.html#APP-PSQL-META-COMMANDS-COPY
- Upload the file to the remote machine and then execute your command. Just make sure you are referencing the correct path and filename.



Tree Construction

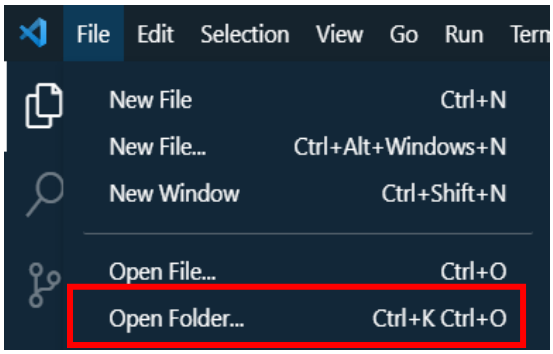
6.1 Website Development

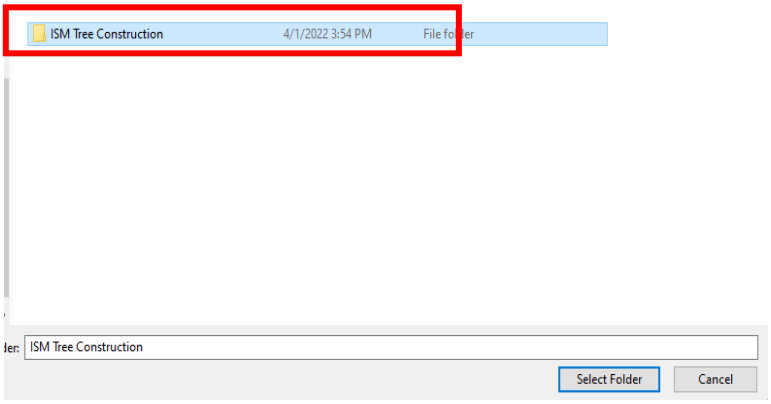
- To run this file first install node.js as explained in 5.1.1.1.
- Install Visual Studio code as explained in 2.2
- Open VS code from your PC



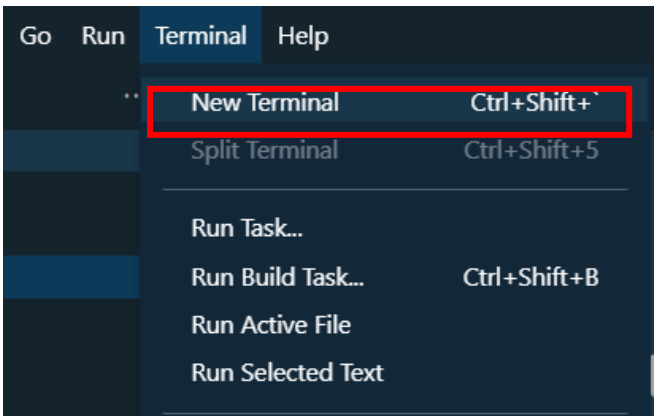
6.1.1 ISM Tree Construction on the basis of Bluetooth

- Click on file and locate to the folder named ISM Tree Construction

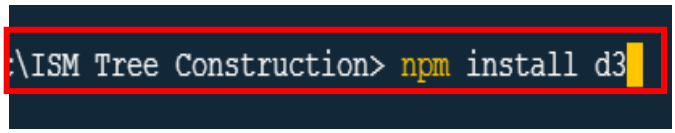




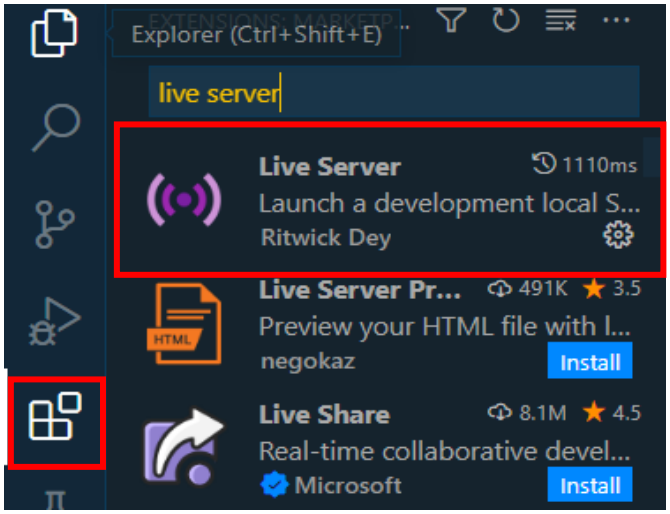
- Open the new terminal in VS code, it will open below the screen and will show the location of the project



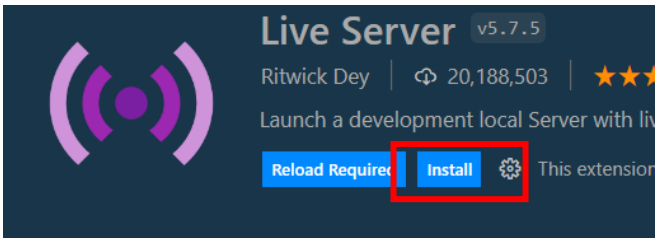
- Need to install the library named d3.js by command: ***npm install d3*** and hit enter



- Click on the Extensions icon and write live server in the search bar and click on it



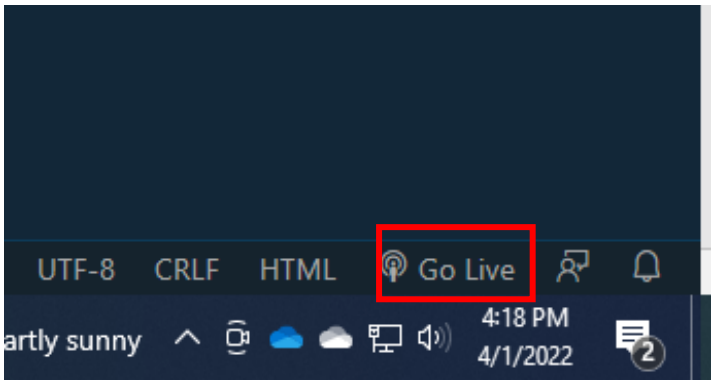
- Now click on the install button and install it



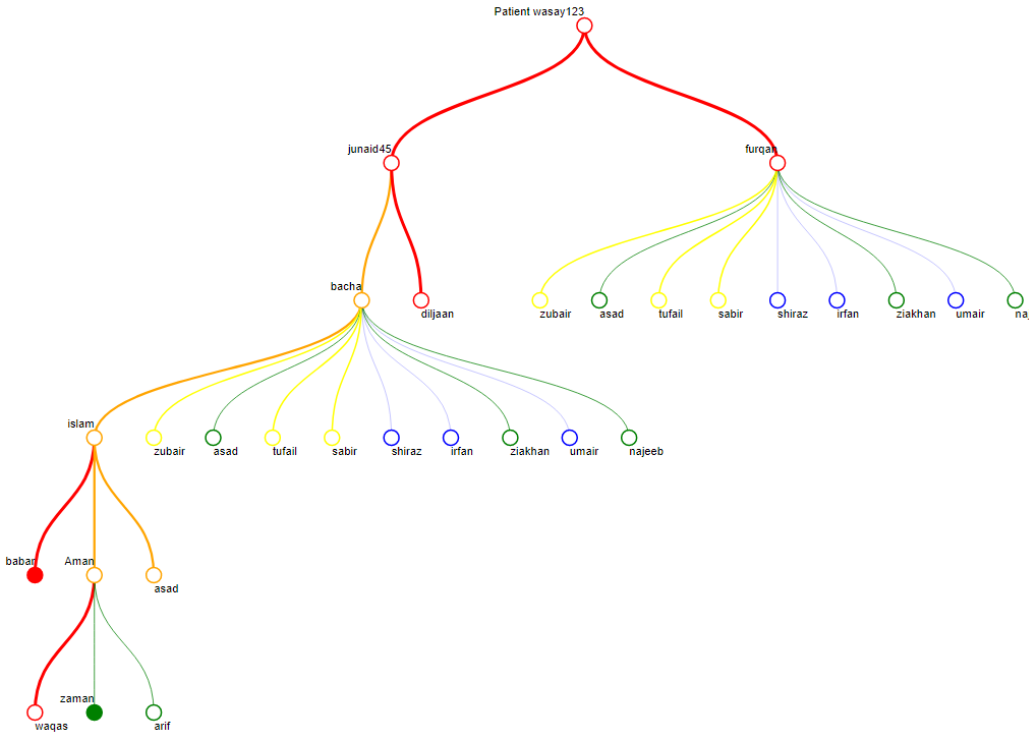
- Double click on treeobtheBasisofBLE.html file from left bar and it will open in a new window as show in below figure

```
treeontheBasisofBLE.html
treeontheBasisofBLE.html > head > style > .node circle
1  <!DOCTYPE html>
2  <meta charset="utf-8">
3
4  <head>
5      <style>
6          .node {
7              cursor: pointer;
8          }
9
```

- Now click on the Go live icon in vs code in the right below corner



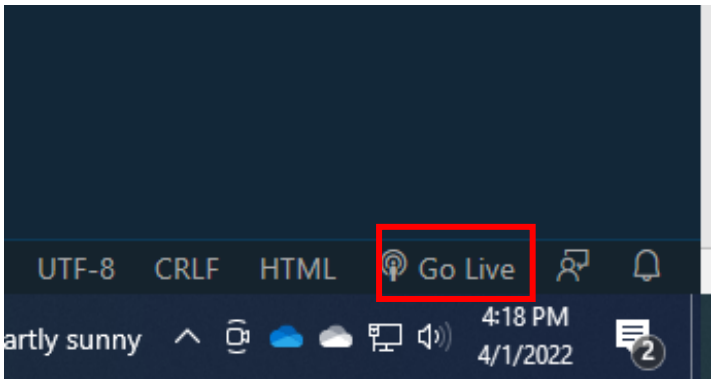
- The output will be generated on the browser page



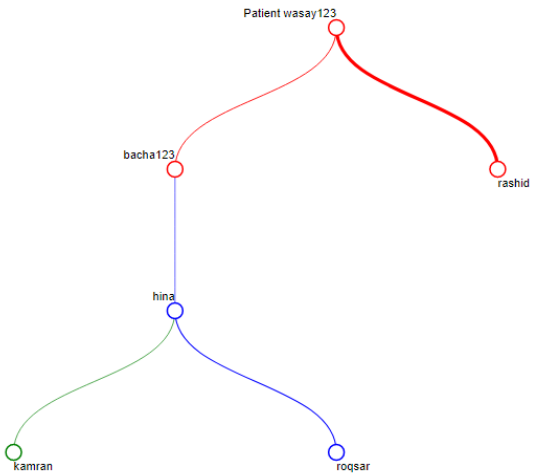
6.1.2 ISM Tree Construction on the basis of GPS

- Double click on treeobtheBasisofGPS.html file from left bar and it will open in a new window as show in below figure

- Now click on the Go live icon in vs code in the right below corner

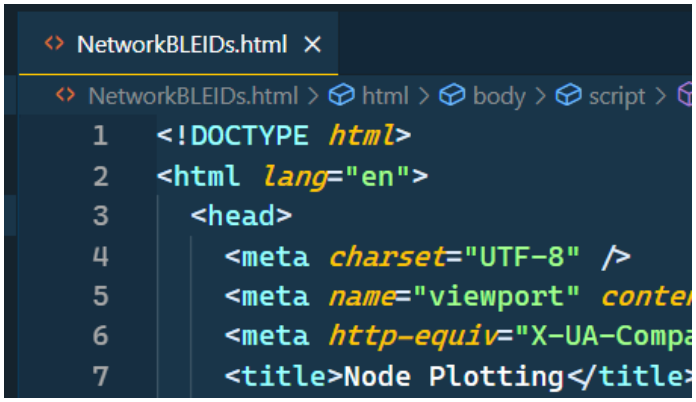


- The output will be generated on the browser page



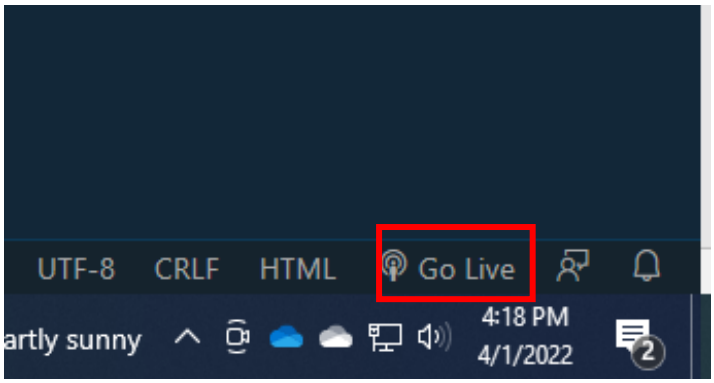
6.1.3 ISM Network Construction on the basis of Bluetooth IDs

- Double click on NetworkBLEIDs.html file from left bar and it will open in a new window as show in below figure

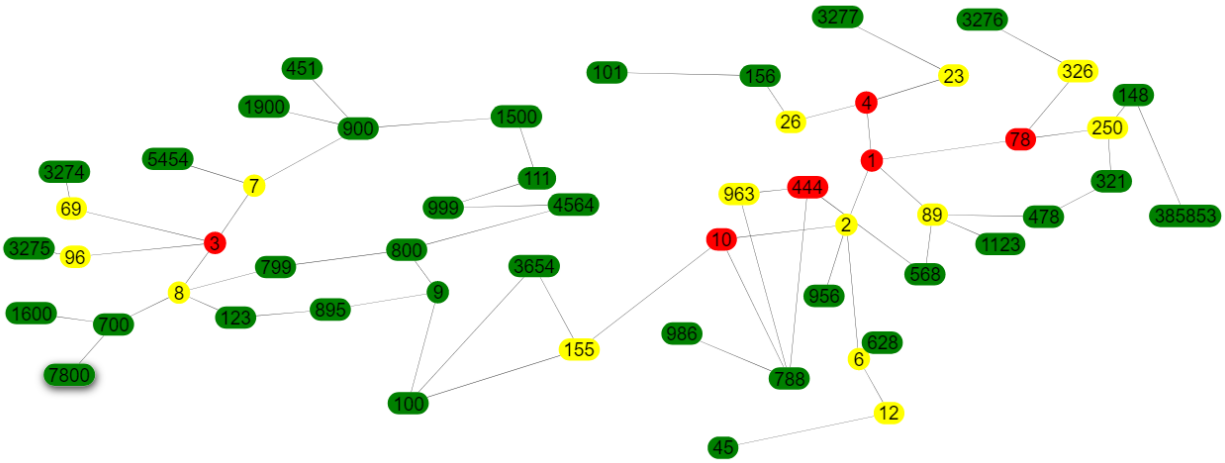


- Include library "<script src="https://cdn.zoomcharts-cloud.com/1/stable/zoomcharts.js"></script>" in your code body, you can download it from here ***https://zoomcharts.com/developers/en/overview/installation.html***

- Now click on the Go live icon in vs code in the right below corner



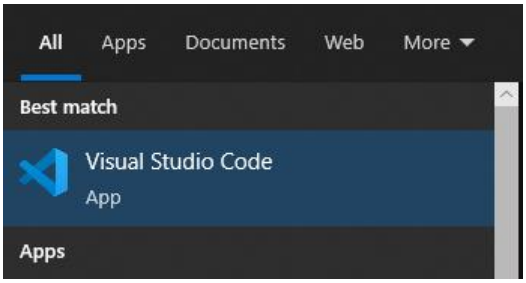
- The output will be generated on the browser page



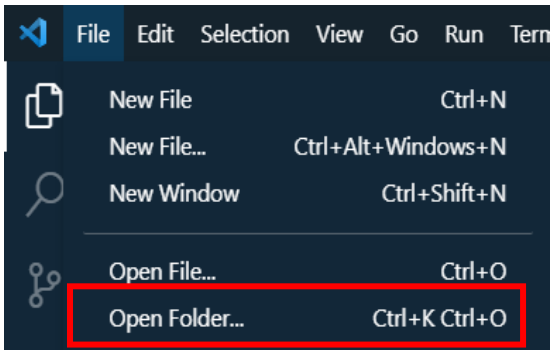
Geo Fencing

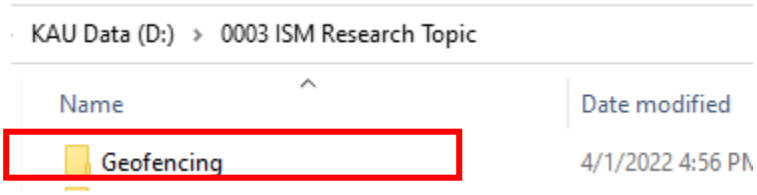
7.1 Website Development

- To run this file first install node.js as explained in 5.1.1.1.
- Install Visual Studio code as explained in 2.2
- Open VS code from your PC

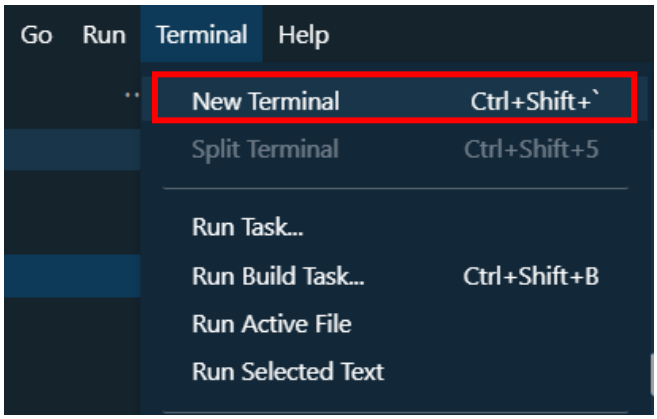


- Click on file and locate to the folder named Geofencing



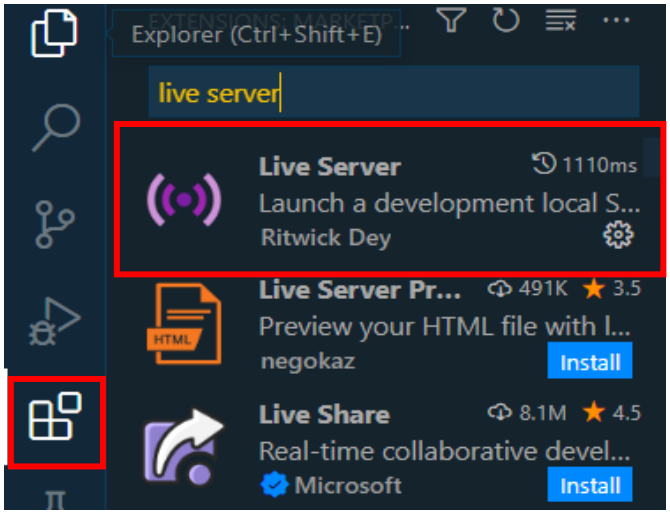


- Open the new terminal in VS code, it will open below the screen and will show the location of the project

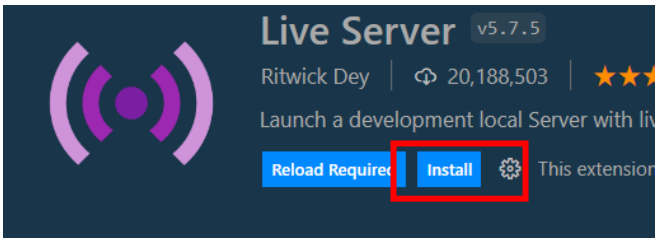


- Need to install the library libraries one by one enter in the terminal and wait to install after completion install the next one Command:
  npm install node-csv
  npm install mongoose
  npm install ejs
  npm install express session
  npm install node-csv
  One example is given below

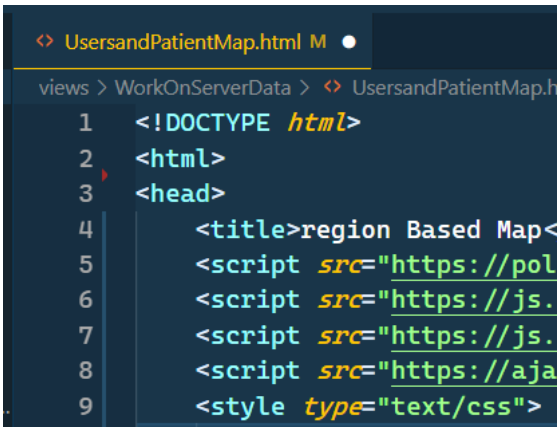
- Click on the Extensions icon and write live server in the search bar and click on it



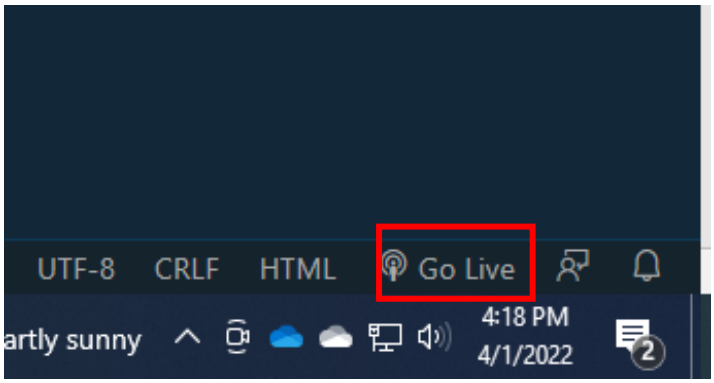
- Now click on the install button and install it



- Double click on file in folder views/WorkOnServerData/UsersandPatientMap.html file from left bar and it will open in a new window as show in below figure



- Now click on the Go live icon in vs code in the right below corner



- The output will be generated on the browser page

References

1. Triantafyllidis A, Kondylakis H, Votis K, Tzovaras D, Maglaveras N, Rahimi K. Features, outcomes, and challenges in mobile health interventions for patients living with chronic diseases: A review of systematic reviews. *Int J Med Inform.* 2019 Dec;132:103984.doi : 10.1016/j.ijmedinf.2019.103984.

2. Kouroubali A, Koumakis L, Kondylakis H, Katehakis DG. An integrated approach towards developing quality mobile health apps for cancer. In: Moumtzoglou A, editor. *Mobile Health Applications for Quality Healthcare Delivery*. Hershey, PA: IGI Global; 2019. pp. 46–71.

3. Smith AC, Thomas E, Snoswell CL, Haydon H, Mehrotra A, Clemensen J, Caffery LJ. Telehealth for global emergencies: implications for coronavirus disease 2019 COVID-19) *J Telemed Telecare.* 2020 Jun;26(5):309–313.doi: 10.1177/1357633X20916567.

4. Katehakis D, Kavlentakis G, Stathiakis N, Logothetidis F, Kouroubali A, Kondylakis H, Petrakis Y, Tzikoulis V, Kostomanolakis S. An outbreak response tool to effectively support surveillance of suspect, probable and confirmed incidence cases while staying safe in COVID-19. 20th IEEE International Conference on Bioinformatics and Bioengineering; October 2020; Virtual conference. 2020.