

APPENDIX - GUIDE TO A TECHNICAL SPECIFICATION DOCUMENT

OVERVIEW

This section should give a basic overview of the project and the organisation behind it. An overview could include:

- **About your organisation** – A brief company background and history.
- **What problem are you trying to solve?** – Why is the project needed?
- **High-level project scope** – Is it a redesign of a few pages, a complete website overhaul, or a brand new website?
- **Target market** – An overview of who this website is targeted at. This could also be it's own section in the document.

PROJECT TEAM

- Amit Ch - CEO - amitch@avakar.org
- Gayathry Manuguri - CEO - learn@avakar.org
- Aditya Verma - Tech Intern - adityave@avakar.org
- Rishu Kumar - Tech Intern - rishuku@avakar.org
- Maitreyi - UI/UX designer- maitreyibh@avakar.org

GOALS

- Increase the number of signups on the coughloud website
- Have a presentable design
- Be future ready for next versions

PHASES

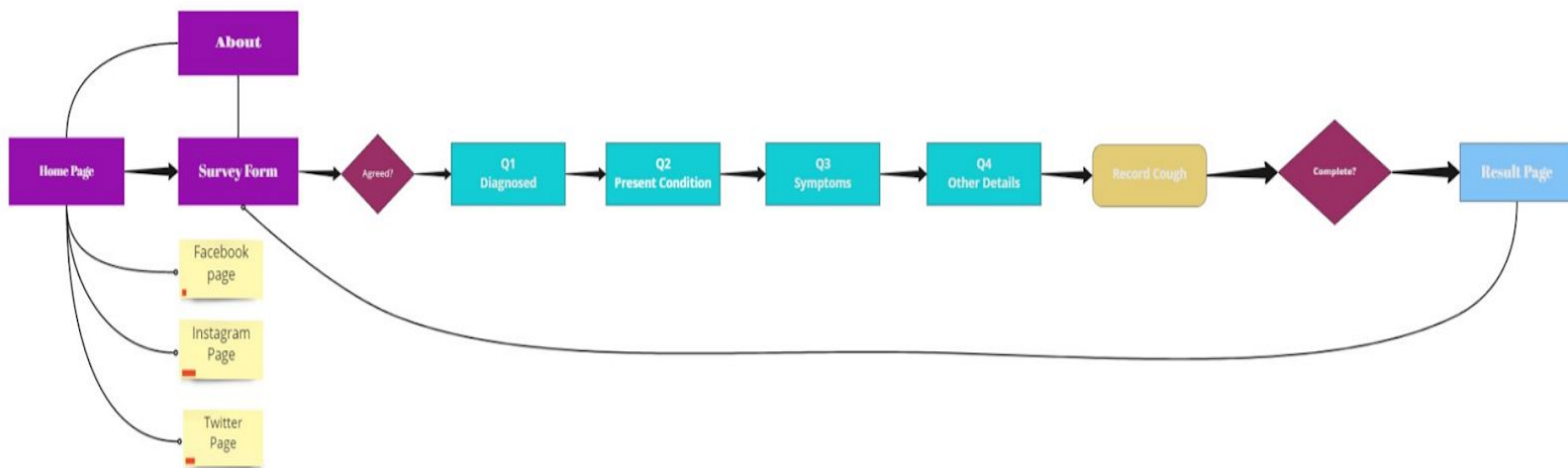
Phase 1 : Coughloud 1.0 (Prototype)

Phase 2 : Coughloud 1.2 (Enhanced design & developer code etc.)

Phase 3 : Coughloud 2.0 (AI integration, serverless deployment)

CONTENT STRUCTURE

Site Map



Content type data

CoughLoud is basically a survey that collects medical information from the patient. We intend to receive data from the user by the following questions:

	Question	Data received
1.	Have you been Diagnosed of COVID-19?	Diagnosis of patient
2.	Describe your present condition	Health status of patient
3.	Symptoms	Symptoms shown - fever, dry cough, shortness of breath, fatigue, headache, sore-throat, sneezing, chest pain, diarrhoea, nausea, etc.
4.	Other Details	Contact Number, State, City, Age, Email, Gender
5.	Record Cough	Audio recording of the cough

Page templates

The templates of the CoughLoud Website are as follows:

- Home
- About
- Consent Form
- Questions-
 - Diagnosis
 - Present Condition
 - Symptoms
 - Other Details
- Record Cough
- Result Page

Wireframes of mobile version-

https://www.figma.com/file/5RmY261e8kyd3ryNOUMsY8/CoughLoud_Mobile?node-id=0%3A1

Wireframes of PC version-

<https://www.figma.com/file/s9l91hvTokP4BZH8cGXyk7/CoughLoud?node-id=0%3A1>

If you have designs (wireframes or mockups) for these page templates please include them here.

DESIGN

The high-fidelity prototypes of the website are designed in Figma. The reason we used Figma is that it is easy to extract the images and other content in different sizes and extensions. It also provides the CSS of all the elements of the wireframe. Although it cannot be used as it is, it gives you an idea about the dimensions and basic styling of the element.

For Front-end development, we have used Vuejs framework. It is easy to learn and provides inbuilt Html attributes that easily link the front-end with the back-end.

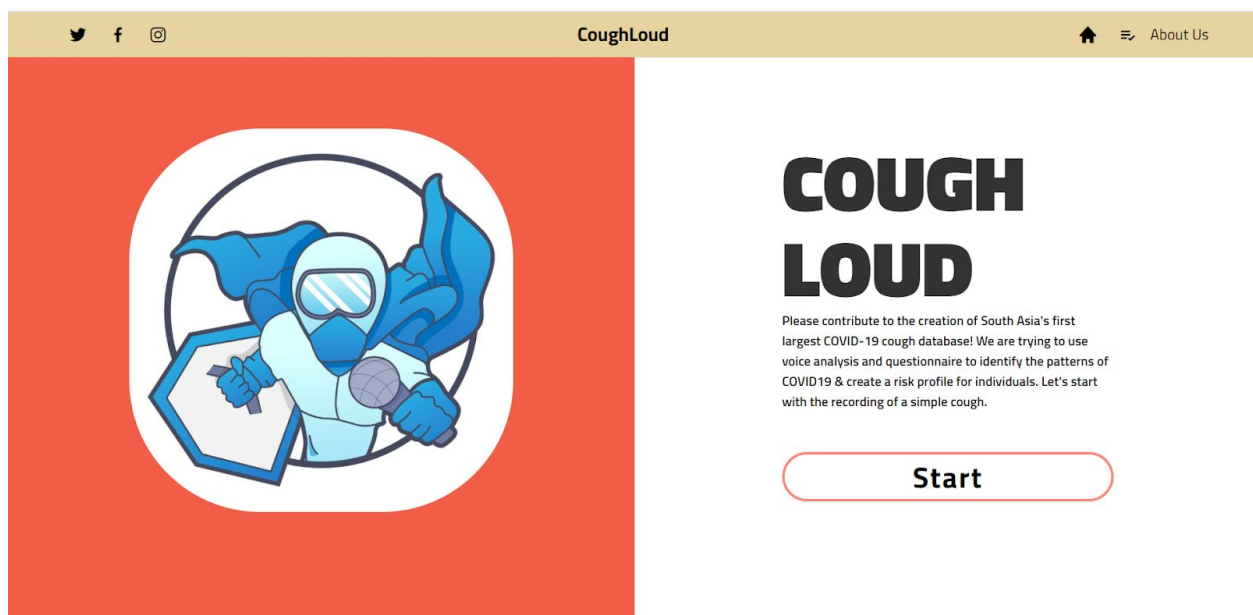
The structure of the project is quite simple. The webpage index.html uses a Vue instance defined in app.vue, which imports all the different webpages of

the website as Vue components. Any Vue component includes the Html code, the CSS and the script in the same .vue file.

Lets now briefly discuss the overall design and structure of the website-

Navbar- It is a fixed navbar present at the top of every webpage. It contains the links to the social media platforms, and the major sections of the website- Home (Landing page), Survey and About. All the icons or buttons in the navbar have a hover state in which it changes to white.

Landing Page - The landing page is the first page of the website, which briefly describes CoughLoud, and has a button to start the survey. The start button has a hover state in which the color changes to red.



About - This page contains the information about CoughLoud. It contains two paragraphs showing you the full information of what we are doing, and what we are aiming to achieve.

How much risk you at?

About Us

"Coughloud" is an initiative with a vision to perform risk assessment of COVID-19 from home. The world is on a standstill at the moment and we do not have enough testing kits for everybody. We are building different ways to assess the risk for COVID-19. Recent studies have shown evidence that identifying certain patterns assisted with artificial intelligence can identify lung diseases like flu, bronchiolitis and pneumonia. We believe the same may hold true for COVID-19.



We envision to build the database with indicated symptoms through data surveys and create AI

Consent form - This page is the beginning of the survey form. This page contains a lot of information like the purpose, title of the project, benefits & risks, confidentiality, liability, etc. and at last there is a checkbox that the user needs to select so that he/ she can go to the next page. The proceed button is disabled if the user has not selected the checkbox. There is also a hover effect on the proceed button.

Survey

Consent Form

Concise Summary

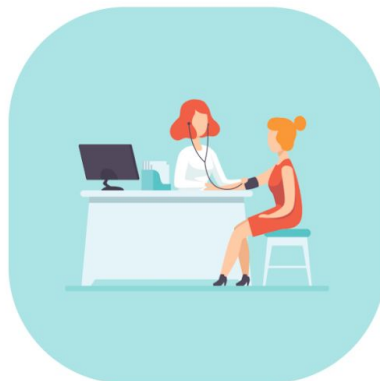
The purpose of this venture is to collect data that is useful for diagnosis of COVID-19 patients based on voice analysis. If successful, we hope to use the cough samples as a new screening method for early detection of COVID-19 and other respiratory infections. However, at this stage, we are collecting the audio information for internal training purposes, and are not currently using it to generate the risk assessment for COVID-19. The results shown are based only on the symptoms picked by the user. We have included those symptoms which have shown association with COVID-19. Your data is stored securely.

You will be shown a short questionnaire to collect demographic information, approximate location, as well as an indication whether or not you have been tested positive for COVID-19. You will be asked to provide a voice sample of your cough. If you are interested in learning more about this study, please continue to read below. This tool does not provide medical advice and it is intended for informational purposes only. It is not a substitute for professional medical advice, diagnosis or treatment. Never ignore professional medical advice in seeking treatment. Please consult your physician for any concerns.

☐ Agree to terms and conditions

Proceed

Diagnosis- This is the first question of the survey, i.e. whether the user has been tested for COVID-19 or not. It has three buttons as options to the question, each having a hover state same as the start button of the landing page.



Have you ever been diagnosed with COVID-19 in the past 30 days?




No, Tested Negative

Tested Positive



Not Tested




Present Condition- The second question of the survey, i.e., what is the present question of the user. It has four buttons as options to the question, and a health-meter below it. The reading of the meter changes is displayed only when the buttons get hovered and its value changes accordingly. Each of the buttons is assigned a boolean variable, which is true only when it is in the hover state. This boolean variable activates an advanced CSS of the corresponding value of the button in the health-meter.

CoughLoud

  About Us


COUGH

Describe your present condition.

OKA Little Sick

SickVery Sick

←●●●●→
AlrightVery Sick

<https://c3.healthwin.in/symptoms>

Symptoms- The third question of the survey, i.e, what are all the different symptoms of COVID-19 that the user might have. The webpage has a check-list for selecting the different symptoms, and a button for the case when the user does not have any symptoms.



In the past 20 days, did you experience any of the below symptoms ? *

These symptoms will help us understand and assess your risk. Select as many as applicable. The listed symptoms were indicated in the studies of several COVID 19 patients.

☒ Fever
☒ Dry Cough
☒ Shortness of Breath /
Difficult breathing
☒ Body pains / Fatigue
☐ Headaches

☐ Sore-throat
☐ Runny Nose / Sneezes
☐ Chest Pain
☐ Diarrhoea / Loose motions
☐ Nausea / Vomitting sensation

NO SYMPTOMS

Other Details - The last question of the survey, it is a form that asks the user to add information like contact number, state, city, gender, etc.

Other Details

Contact Number *

9835803029

State

Mumbai

Why we collect it: It will help us connect you to health providers and arrange for help in the time of need.

Age*

32

Email:



raj123@gmail.com

Gender*

Male

Current Location:

Mumbai

Record Cough - This webpage is for the collection of cough samples. It has some basic instructions about how to record the cough sample, and a button that starts and stops the recording. Once you've recorded your cough, you

can play the recording to check it, or you can continue to the results by clicking on the complete checkbox and then on the next button. You can proceed further only after recording the cough. This is again done using advanced CSS, the complete checkbox is displayed only when you've recorded the cough sample, and the next button is displayed only when you click on the complete checkbox.

Record Cough

Please ensure you are in a quiet place away from noise. Press the button and cough 5 times. Listen to the recording, and re-record if needed.

Press Take survey button after recording.

Note: Please try to cough even if you don't have cough like symptoms, as all types of coughs will make our algorithms better.



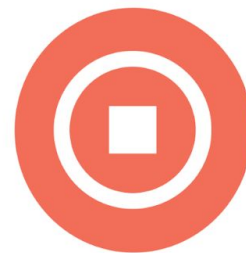
Your contribution helps !

Record Cough

Please ensure you are in a quiet place away from noise. Press the button and cough 5 times. Listen to the recording, and re-record if needed.

Press Take survey button after recording.

Note: Please try to cough even if you don't have cough like symptoms, as all types of coughs will make our algorithms better.



Your contribution helps !

Record Cough

Please ensure you are in a quiet place away from noise. Press the button and cough 5 times. Listen to the recording, and re-record if needed.

Press Take survey button after recording.

Note: Please try to cough even if you don't have cough like symptoms, as all types of coughs will make our algorithms better.

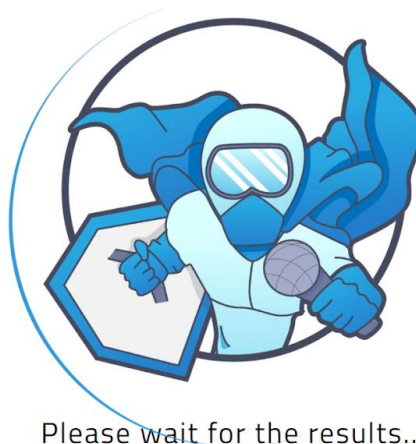


 Play Recording

☒ Complete

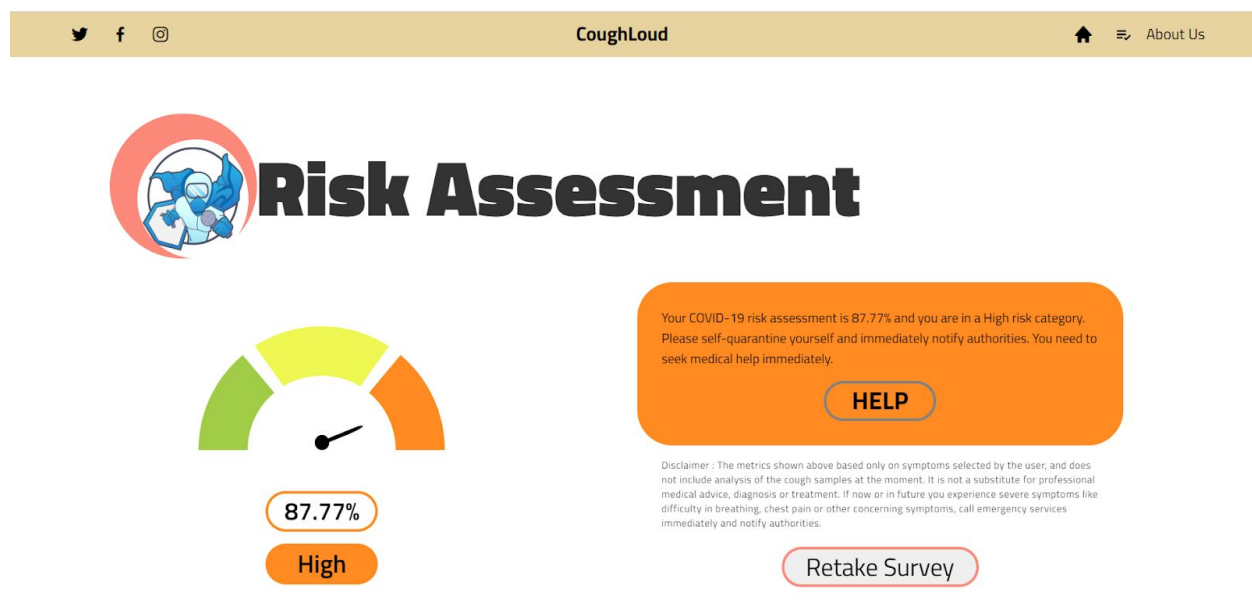


Waiting page - It is a loading page that is displayed while the results are being calculated in the backend. It has a loading animation around the CoughLoud mascot which has a rotation time of two seconds.



Please wait for the results...

Result page - This is the final page of the survey, It displays the risk assessment of the user calculated on the basis of the information given by the user. The percentage value is displayed below a speedometer-type animation, made using CSS. The pointer has a fade-out type propagation with an animation time of two seconds. It also has a small analysis of the predicted health status which is according to the risk percentage. This is also done using advanced CSS property of Vuejs. Finally, you could get to the website of the ministry of health by clicking on the help button, in case you think you need help. Else, you can retake the survey for someone else.



Responsive designs

We have designed the structure of the webpages in such a way that while following the design of the wireframes, we can make it compatible with different devices and window sizes.

Most of the web pages have their content in two different div attributes- left and right. When the size of the window is decreased, the 'right' div gets aligned below the 'left' div. If the window size is further decreased, the CSS properties of the affected elements of the webpage get scaled accordingly. As far as the mobile templates are concerned, their properties are inherited both from the wireframes and the PC version with a small window size. This ensures a smooth change in the alignment and design of the webpage which is clearly seen as you increase or decrease the screen size.

Talking about the responsiveness of the Navbar, the contents get scaled according to the width of the window/device; the height of the navbar is constant, i.e., 55 px. When the contents tend to overlap each other, the title in the center of the navbar disappears, which also follows the navbar design of the mobile wireframes.

FUNCTIONALITY

Back-end:

The main thing to keep in mind is that the data needs to be passed between components so that it can be saved after all the calculations are done. For this, we used bus events in vuejs. After passing those datas, we need to save it somewhere so that it can be retrieved up to our desire.

For the backend integration and saving those datas , we used firebase. For using firebase in vuejs application, just copy and paste the **firebase configuration** in main.js file, which is created when you create a firebase project. If you forgot to copy the firebase configuration, don't worry, you can still get the firebase configuration from project settings.

After the firebase is set up for the application, we created an instance of the firebase and used it for storing the relevant information.

We used a firestore database to store our data in the backend and for storing the cough data, we used firebase storage.

For the code part, we have defined most of the functions in the recordCough component. Uptil then, we were asking users to fill in the information.

Functions in recordCough component:

- `handlePlayCough()` - this function hides the mic button and displays a header at the top of the webpage and displays a mic-on image where you can record your cough.
- `handleRecordCough()`- this function hides the mic-on image and again displays the mic image so that you can record again if you want.
- `startRecording()`- this function starts recording by using `mediaDevices` and it is using `stream` to get the audio. By using `(rec.record())`, we can easily record audio. `rec` is an instance of `recorder-js`.
- `playRecording()`- this function plays the recording by using the `audioPlayer.play()` . `audioPlayer` is an inbuilt html tag.
- `stopRecording()`- this function stops the recording by using `rec.stop()` . After stopping the recording, it creates a .wav file by using `createDownloadLink()`.
- `pauseRecording()`- this function pauses the recording by `audioPlayer.pause()` .
- `resumeRecording()`- this function resumes the recording by using `audioPlayer.play()` .

- `createDownloadLink()`- this function takes `blob(file)` as an argument and creates the link for the given file. We can pass this url to the src of the audio player so that it will get to know what to play.
- `calculateRisk()`- this function calculates the risk% where we have given weights on the symptoms and it changes the value of `riskP` .
- `uploadAudio()`- this function is basically doing the most important work. At first, it is creating a storage ref in firebase storage and then it is uploading the `blobA(file)` to the firebase. The uploading file returns an upload task variable. We have added an event listener '`state_changed`' to this variable so that when it's state gets changed, we can do something. We are adding data to the firestore database when it's state is being changed.

Google SpreadSheet:

This is basically used to show our data to a non technical person. We used the most recent way to add data from the firestore database to a google spreadsheet. You can find the code on the above github link.

By this, you can display any firebase project data to a google sheet.

You just need three things - `client_email`, `client_key`, project id. You can get these details from the services account from firebase project settings.

Open a google spreadsheet and go to the tools option in the menu bar, then click on Script editor and paste the github code here (keep in mind to change the '`email`', '`key`' and '`project id`' of the code). Come to your sheet, refresh it and you will see a firestore option in the menu bar. Click on it and then click on the import option and you are done.

You will see every data of firebase in your google sheet.

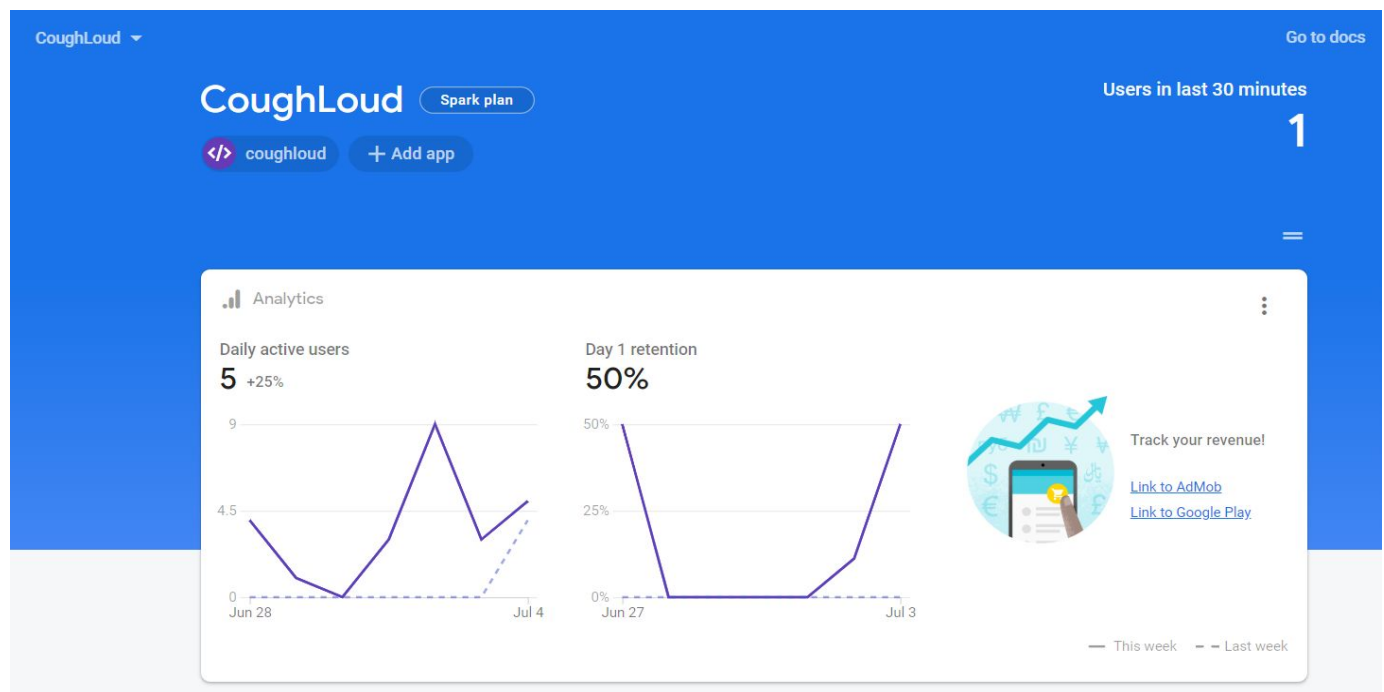
The best thing here is that you don't need to touch the backend code. So it's like if your project is live and you were told to get those data, you don't need to do any changes in the backend.

In the code of google-spreadsheet, there is a function `getFirestore()` which is connecting the firebase project to google sheet. There is another function

`importFromFirestore()` which is creating an instance of firestore and getting all the field values of the data and storing it in `allDocuments`. Then, we added `allDocuments` to the sheet by using the `appendRow()` method. We have defined a function `onOpen()` to display the firestore option in the menu bar so that if a user clicks on that button, it will refresh the sheet and will retrieve all the data from the firestore.

TRACKING PERFORMANCE

Firebase provides an inbuilt analytics where we can see how many users have enrolled so far. We can track if this app crashes or there are some performance related issues. Firebase has inbuilt features like crashlytics, performance. We just need to add the sdk file to our webapp.



HOSTING REQUIREMENTS

Hosting is done by the firebase. Firebase provides an easy way to host the web application in no time. Follow the below steps to host your vuejs application:

- Go to the firebase console and select your project.
- Click on the hosting from the left main drawer and it will ask you to follow the steps.
- First of all, install firebase cli by pasting this command (**npm install -g firebase-tools**) in cmd or your code editor terminal.
- Run **npm run build** command in your code editor terminal. It will build a dist folder containing all your images and stuffs required for hosting.
- Then click on next and sign in to google by **firebase login** command and initiate your project by **firebase init** command.
- Firebase init command asks you to follow the instructions and then you can host the project by **firebase deploy** command.

There is another approach if you are facing any difficulty in firebase init command. In this, you don't need to use firebase init command. Just follow the below steps:

- Go to the firebase console and select your project.
- Click on the hosting from the left main drawer and it will ask you to follow the steps.
- Read the steps and just click on next and again next and finally click on go to console.
- Install firebase cli by pasting this command (**npm install -g firebase-tools**) in cmd or your code editor terminal.
- Go to your code editor terminal and run **npm run build** command. It will create a dist folder containing images and other stuff required for hosting.

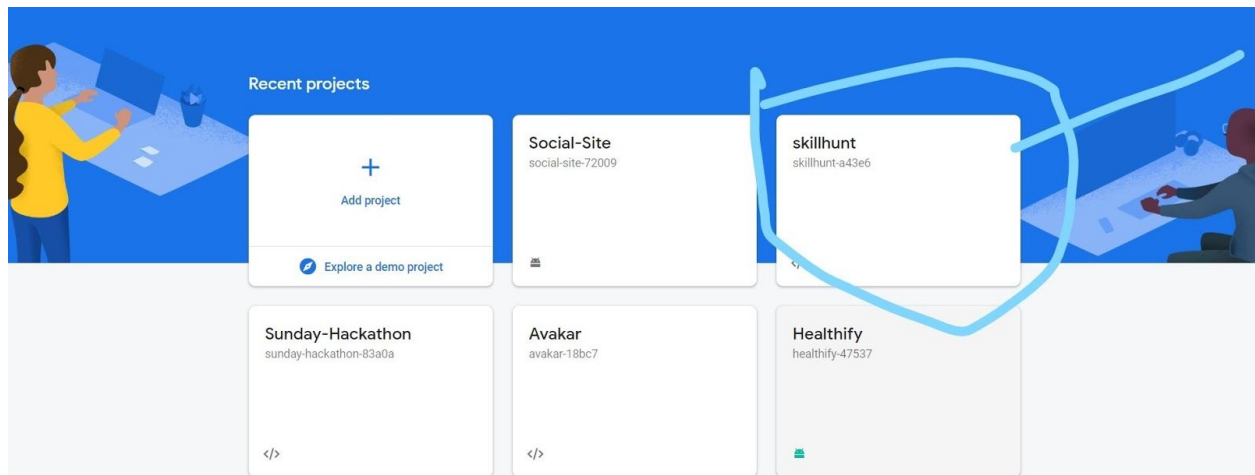
- Create a public folder in your project directory and copy and paste the dist folder and index.html file.
- You can do the hosting on your local computer by running the firebase **serve** command.
- Finally, run the **firebase deploy** command and it will deploy your webapp to the server and generate a link which you can share to anybody.

After the hosting is done, we need to create a domain name for the website so that if a user types that name in any search engine, it will display our website. For generating a custom domain link, follow the below steps:

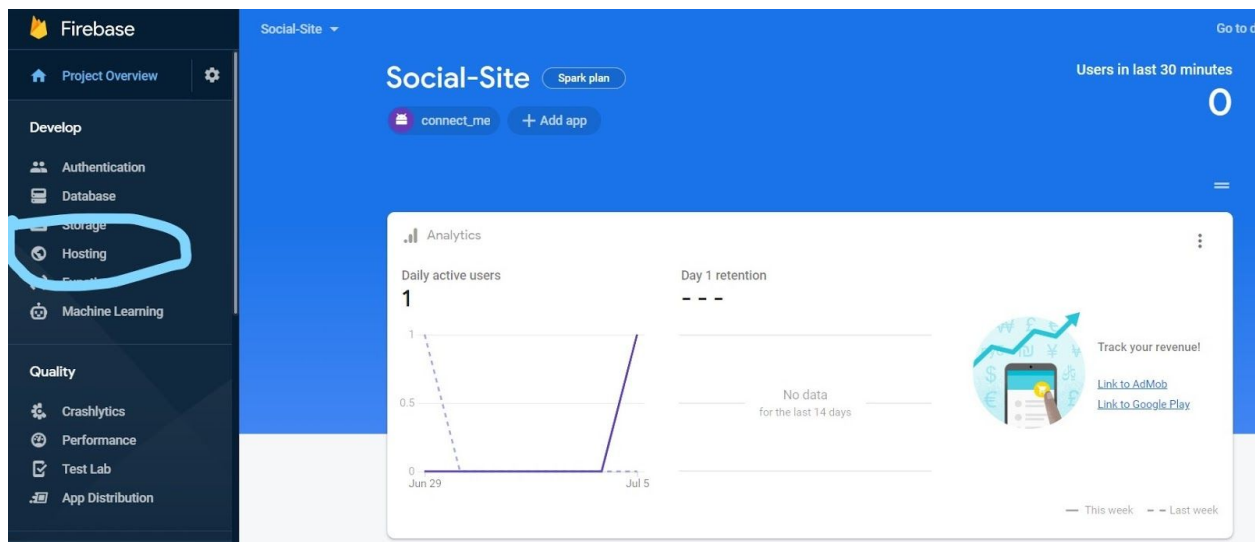
- Go to the firebase console and select your firebase project.
- Click on hosting from the left main drawer.
- In the dashboard section , you will see a custom domain blue button.
- Click on it and it will ask you to add a domain name.
- Add a domain name and it will load the verify ownership page. Copy the google-site-verification txt file.
- Go to any domain purchase site and buy a domain for your website. In our case, we used goDaddy site.
- Go to the dns part and paste the google-site-verification and follow the instructions.
- Come to your firebase console and click on verify, you need to wait for a couple of hours and it will be verified.
- If you face any difficulty in creating a custom domain, watch this tutorial: <https://www.youtube.com/watch?v=Bcn5e57PpUc>.

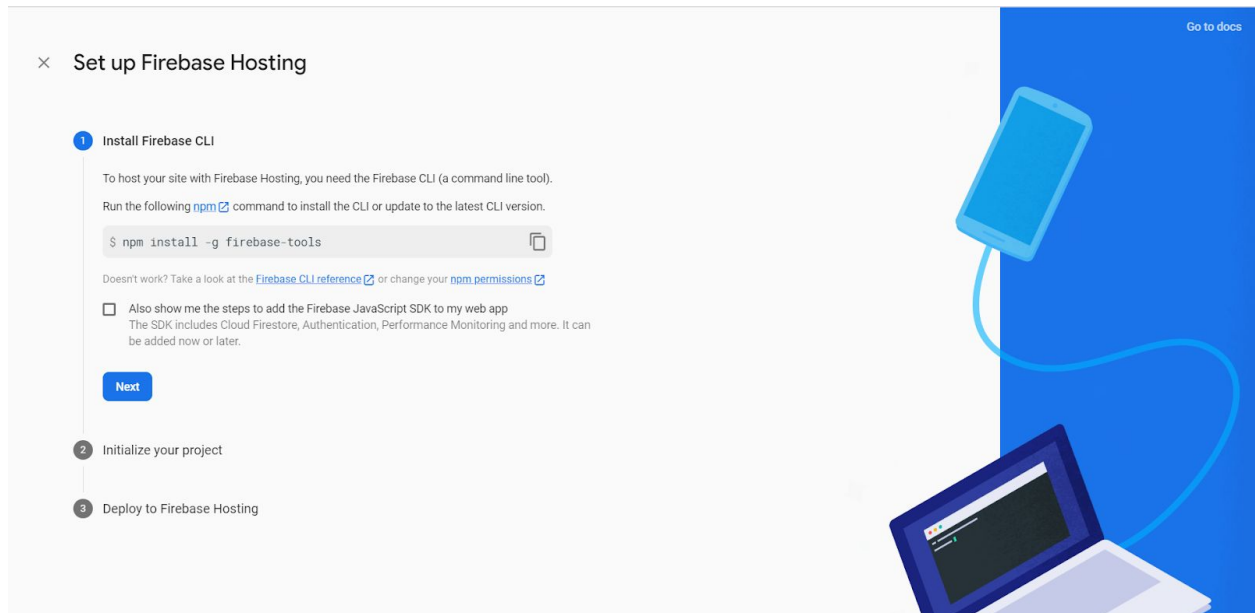
Steps for firebase hosting:

1. Select your firebase project

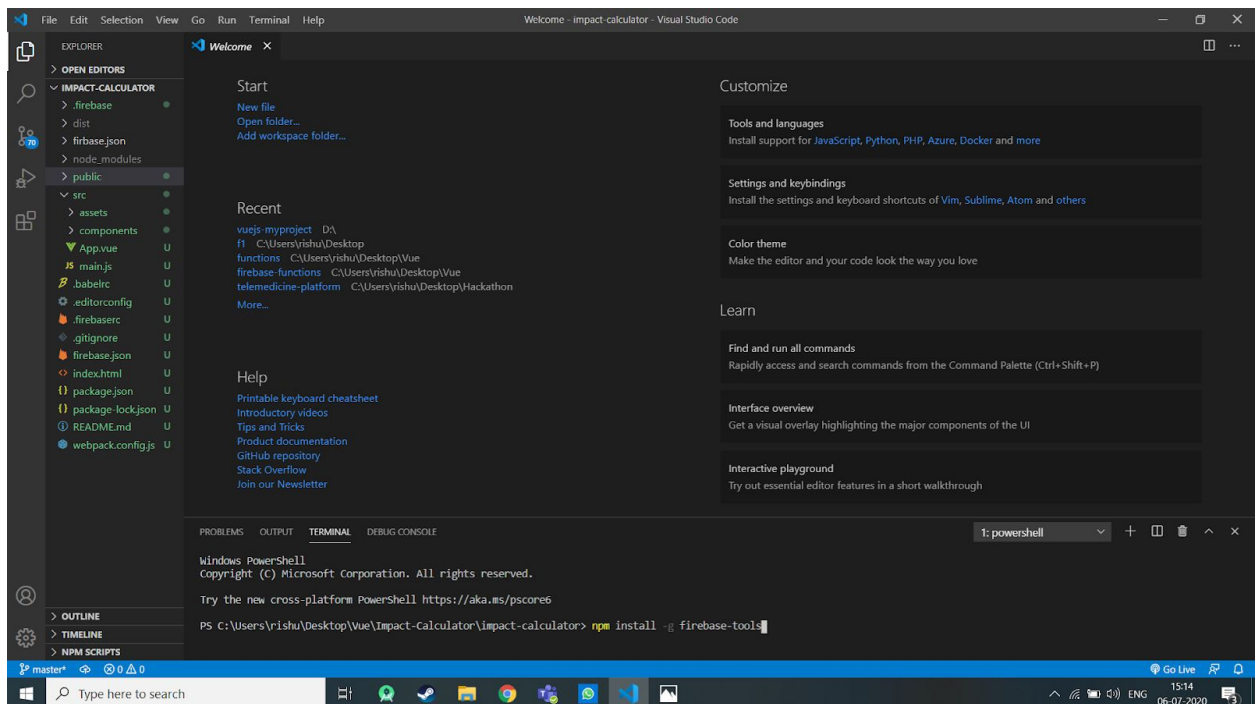


2. Click on hosting and then click on next, just ignore these instructions and finally click on console.

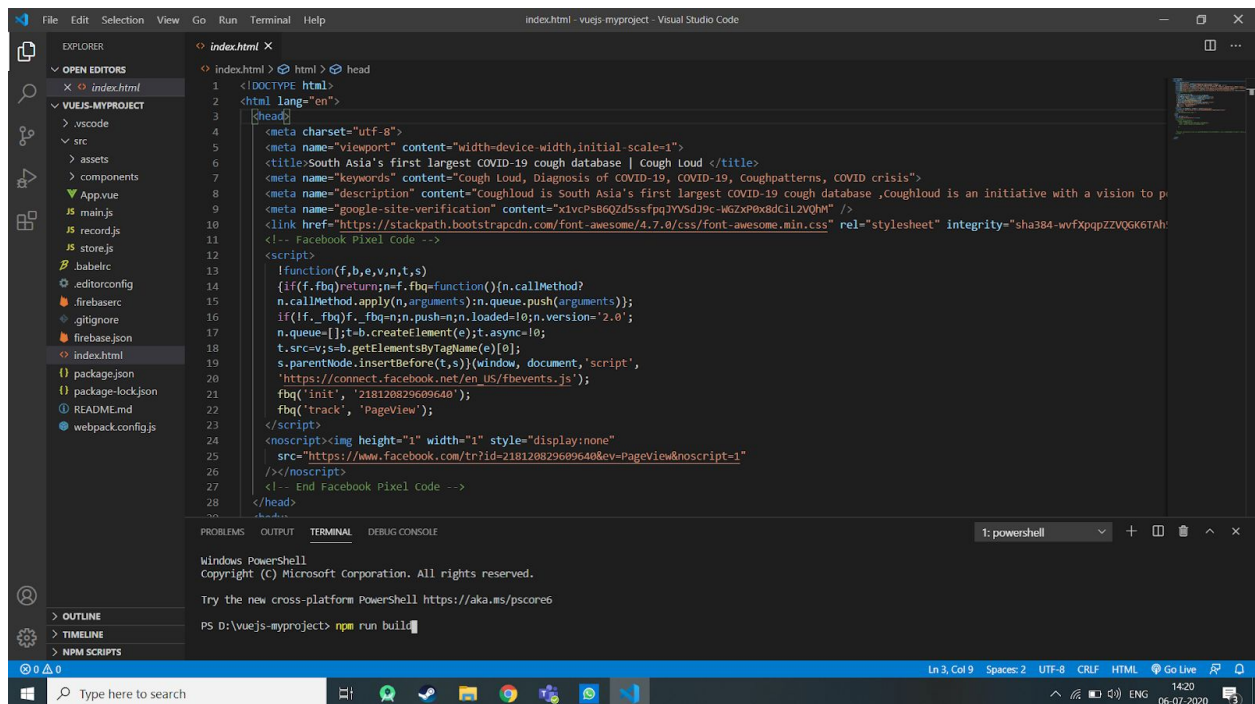




3. Run **`npm install -g firebase-tools`** in your code editor terminal. It will install the firebase-cli so that we can deploy our vuejs application.



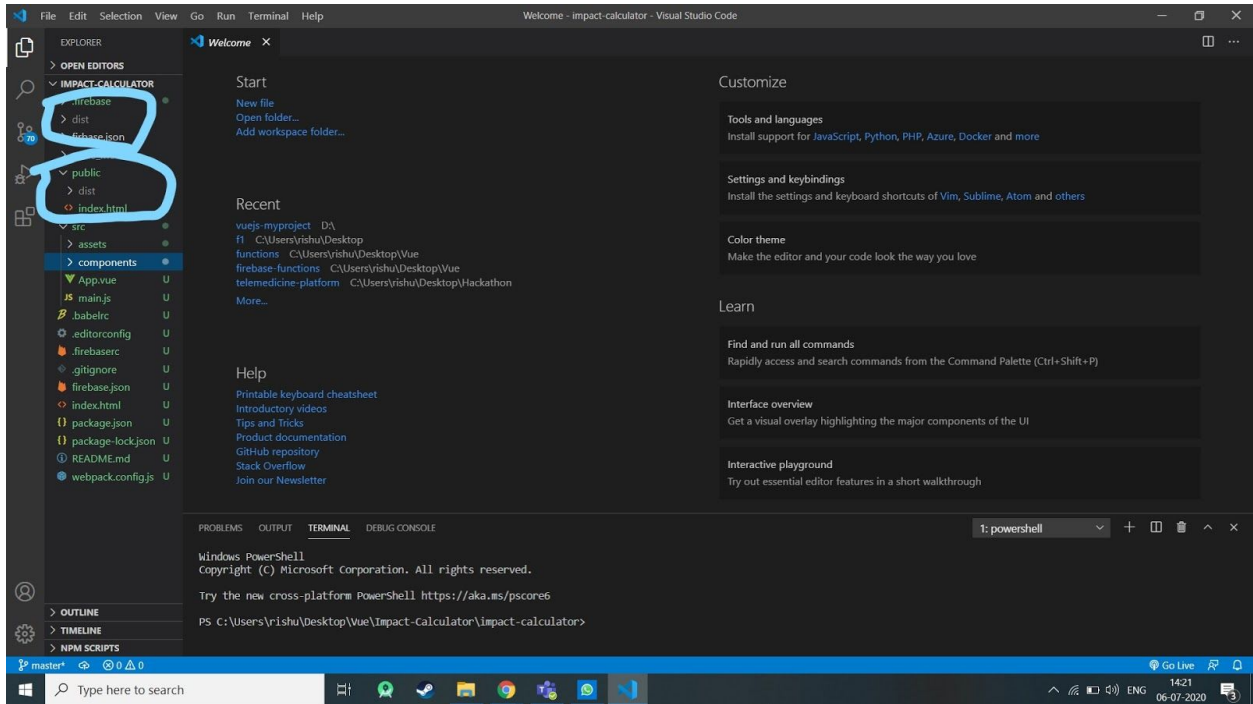
4. In your code editor, run `npm run build`. It creates a `dist` folder. Create a public folder and copy and paste the `dist` folder and `index.html` file in the public folder.



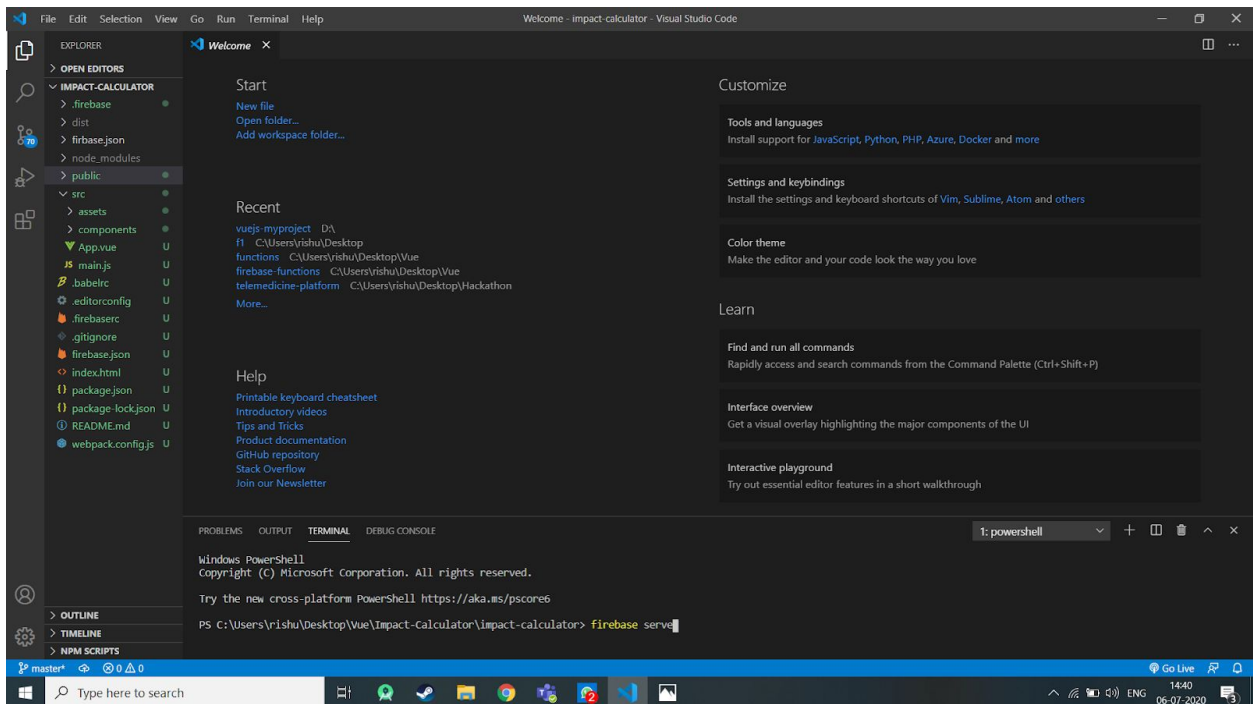
The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left displays the project structure for 'vueljs-myproject', including files like `App.vue`, `main.js`, `record.js`, `store.js`, `index.html`, `package.json`, `package-lock.json`, `README.md`, and `webpack.config.js`. The index.html file is open in the editor, showing the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width,initial-scale=1">
6     <title>South Asia's first largest COVID-19 cough database | Cough Loud </title>
7     <meta name="keywords" content="Cough Loud, Diagnosis of COVID-19, COVID-19, Coughpatterns, COVID crisis">
8     <meta name="description" content="Coughloud is South Asia's first largest COVID-19 cough database ,Coughloud is an initiative with a vision to p
9     <meta name="google-site-verification" content="x1vcPsB6QZd5sfpqJVVSd39c-WGZxP6x8dCil2VQhM" />
10    <link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet" integrity="sha384-wvfxpqpZzVQK6TAH
11    <!-- Facebook Pixel Code -->
12    <script>
13      !function(f,b,e,v,n,t,s)
14      {if(!f.fbq)return;n=f.fbq=function(){n.callMethod?
15      n.callMethod.apply(n,arguments):n.queue.push(arguments)};
16      if(!f._fbq)f._fbq=n;n.push=n;n.loaded=!0;n.version='2.0';
17      n.queue=[];t=b.createElement(e);t.async=!0;
18      t.src=v;s=b.getElementsByTagName(e)[0];
19      s.parentNode.insertBefore(t,s)}(window, document,'script',
20      'https://connect.facebook.net/en_US/fbevents.js');
21      fbq('init', '218120829609640');
22      fbq('track', 'PageView');
23    </script>
24    <noscript></noscript>
27    <!-- End Facebook Pixel Code -->
28  </head>
```

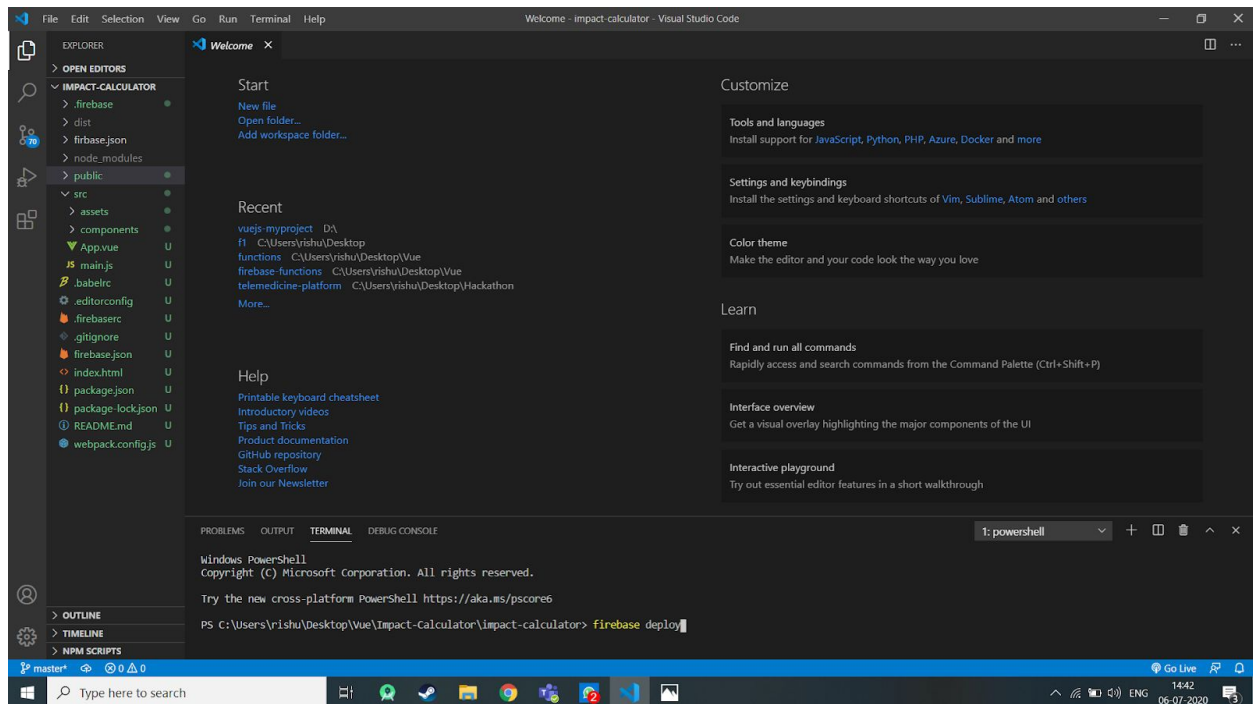
The terminal window at the bottom shows the command `npm run build` being executed in a PowerShell environment. The status bar at the bottom indicates the current file is `index.html` at line 3, column 9, with 2 spaces, in UTF-8 encoding, with CRLF line endings, and HTML file type.



5. You can check your hosting locally by running firebase serve in the terminal



6. Finally, you can deploy your vuejs application by running **firebase deploy**.



It will take some time and give you a link for the hosted website.

When we host our vuejs app with firebase, it gives us an automatically generated link. Here is the firebase link for CoughLoud1.2 website:

<https://coughloud-faaa0.web.app/>

After the hosting, we gave a custom domain link to our website.

check out the website here: coughloud.healthwin.in

ASSUMPTIONS

One of the most common problems that projects run into is that parties have made assumptions about who is responsible for certain tasks.

The classic example is who adds the content. Often people commissioning websites (rightly) assume that the web company will add all of the content. However, often this is not the case and the client receives an 'empty' version of their website.

Your website specification document should include *everything* that is needed for this project to be completed successfully.

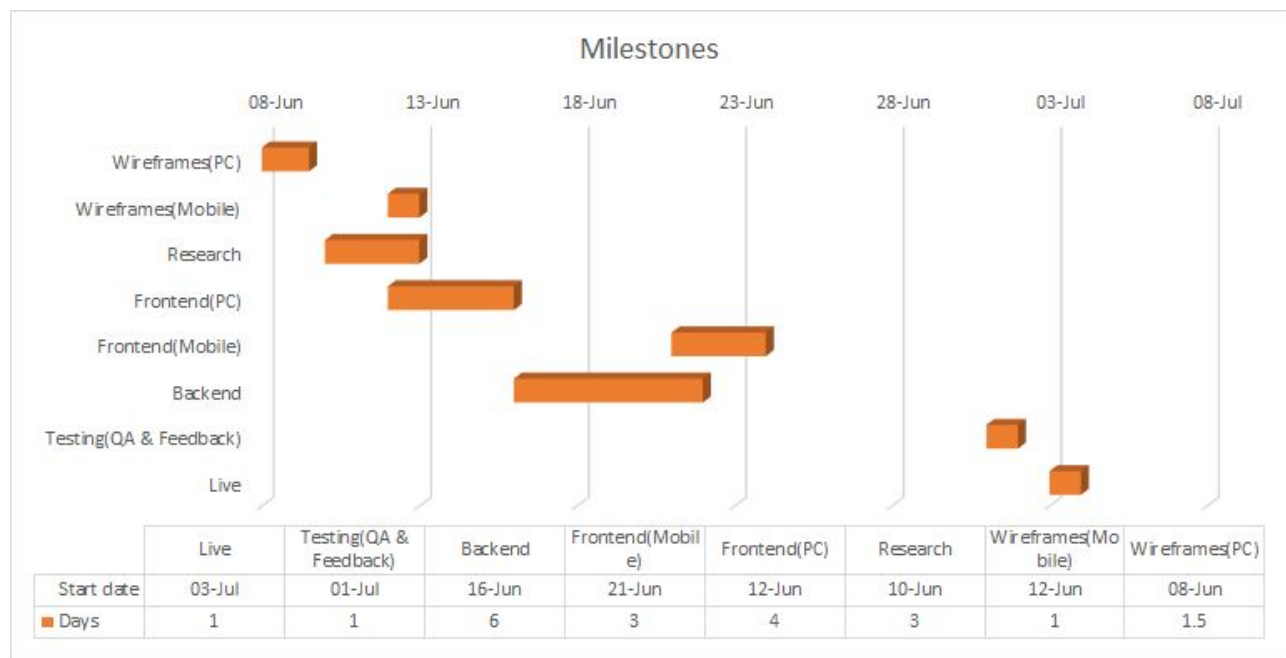
Some common assumptions to think about include:

- Content addition
- Design and layout customisation options
- Migrating the site to the live server
- Ongoing maintenance
- SEO
- Hosting

MILESTONES

Milestone	Description	Time estimated	Time taken
Wireframes (PC)	Designing the high-fidelity prototypes for the PC version of the website	1.5 days	1.5 days
Wireframes (Mobile)	Designing the high-fidelity prototypes for the Mobile version of the website	1 day	1 day
Frontend(PC)	Designing the webpages of the PC version and making them compatible for different window sizes	3 days	4 days
Frontend(Mobile)	Designing the webpages of the	3 days	3 days

	Mobile version and making them compatible for different devices.		
Back end	Saving the relevant information to firestore database and couch data to firebase storage	4 days	6 days
Testing (QA & Feedback)	Making final modifications on the basis of the user experience.	1 day	1 day
Live	Hosting the vuejs application with firebase and giving a custom domain to the hosted application	1 day	1 day



The project took around 15-16 days to complete. The timeline of the entire project is something like this-

Week 1- We dedicated the first three days in learning the basic concepts of Vuejs and setting up the required software into our systems. Next we started working on the wireframes of the PC version and made them responsive for different window sizes.

Week 2- We researched about the backend, and made ourselves familiar with the basics of firebase, cloud functions, etc. By taking references from different sources (mentioned in the references below) and the existing code of the previous version, we completed most of the backend part. This took us probably around four days. Next, we started working on the mobile templates, and simultaneously made it compatible with different screen resolution. We also looked into how we can push our survey data into a google spreadsheet.

In the next few days, after some testing and getting feedback from the designer and the superiors, we made some final changes in the frontend, and finished the backend by connecting the firebase firestore data with a google spreadsheet. Finally, we hosted the website on the existing custom domain using firebase hosting.

REFERENCES

Below are the links in a structured manner so that if you plan to build a project like this, you can follow these:

- Previous CoughLoud github code link: <https://github.com/cluster-one1/covid-19>
- Previous CoughLoud github code link for cloud functions: <https://github.com/cluster-one1/covid19-functions>. It is basically used to display the data to a non-technical person. This firebase functions takes data and displays it to the google-spreadsheet.
- For any design requirements: <https://css-tricks.com/>
- Vuejs tutorial for beginners: https://www.youtube.com/watch?v=5LYrN_cAJoa&list=PL4cUxeGkcC9gQcYgjhBoeQH7wiAyZNRya.
- For more details on vuejs, check out it's documentation: <https://vuejs.org/v2/guide/>
- For passing data between components so that it can be saved later: <https://medium.com/@kashifazmi94/sharing-data-between-component-in-vue-js-acfd71e05815>

- For recording audio, we used recorder-js. For more details check out here: <https://github.com/addpipe/simple-recorderjs-demo/blob/master/js/app.js>
-
- Firebase firestore database tutorial: https://www.youtube.com/watch?v=4d-gIPGzmK4&list=PL4cUxeGkcC9itfjle0ji1xOZ2cjRGY_WB. By this, we can save our data to firestore cloud database.
- Firebase cloud functions tutorial: https://www.youtube.com/watch?v=udHm7I_OvJs&list=PL4cUxeGkcC9i_aLkr62adUTJi53y7OjOf. Cloud functions is basically used to speed up our application. It lets us run backend code on serverless framework in response to triggers in firebase features.
- Deploying vuejs application with firebase: <https://medium.com/@rachidsakara/how-to-deploy-vue-js-applications-with-firebase-hosting-40cfa7f724e4> .
- Generating a custom domain for projects with firebase: <https://www.youtube.com/watch?v=Bcn5e57PpUc>
- CoughLoud1.2 github code link: https://github.com/Rishu-kmr-Avakar/CoughLoud_1.2
- Google-spreadsheet github code link: https://github.com/Rishu-kmr-Avakar/CoughLoud_1.2/blob/master/google-sp.txt.