**SEC : K19KH**

**L P U The**

**Leading University**

**In the World**

REPORT ON C-BOT

Github link - https://github.com/ekesel/cbot

****

|  |  |  |
| --- | --- | --- |
| Name | Reg. No. | Roll. No. |
| Ekaansh Sahni | 11903816 | 21 |
| VikramAdtya | 11901842 | 11 |
| Karan Thakur | 11906442 | 37 |



Submitted T0 : - [ Asst. professor at LPU ] Ankita Wandhawan

Contents

1. Introduction
2. Key Features
3. Methodology
4. Techniques Used
5. Code Screenshots
6. DFD
7. ER
8. Flowchart
9. Output Screenshots
10. Bibliography

INTRODUCTION

**In order to spread awareness among the citizens on the novel coronavirus outbreak in India, 3 LPU students has made a webapp that will help people to track Covid-19 infections more accurately and effectively. The 'C-BOT' webapp, which is currently available on localhost, keep the user updated about number of case rising in India every day.C-BOT has a user friendly interface and one just has to simply register/signup to the webapp for its use. The webapp is easy accessible and can be accessed almost anywhere, or from any device. Every person who use the C-Bot will be helping in the fight against coronavirus (COVID-19). The webapp will help the user understand where and how quickly the virus is spreading, so it can respond quickly and effectively. Besides this, the webapp also has a dedicated chatbot that gives you a series of options to determine if you have any symptoms of Covid-19 as well as inform you about the various facilities and updates from the health ministry along with a series of helpline numbers nationally. The webapp also provides you with the best hospitals options nearby your state. With the launch of this webapp, our objective seeks to limit the spread of the Covid-19 cases in India as well as help to create self-awareness among the citizens with relevant information on the infection.**

KEY FEATURES

**Symptoms**

**If feel unwell, you can use the C-BOT to check if your symptoms could be related to coronavirus you (COVID-19). The webapp will give you a list of potential symptoms and you can then choose the ones that apply to you. It will then tell you if your symptoms suggest you have coronavirus.**

**Daily Update**

**If a user wish to find the total number of cases in India or prior any other state, he just has to write the name of the state and hit enter. The C-BOT automatically retrieves the data from server and present it quickly on the screen.**

**Precaution**

**C-BOT provides the user best precaution and remedies available for its health and safety. It seeks to full-fill all the preventive measures one must take in order to stay protected from infection.**

**Medical Facilities**

**If a user wish to find the nearest medical facility available in his state, he just has to command the C-BOT for medical facilities. The C-BOT will present the series of list of hospitals present in his state. Not only has he had access to the list of his state but also of any other prior state.**

**Vaccine Updates**

**C-BOT is small co-vid Wikipedia that gives you the latest info on vaccines and their updates.**

MTHODOLOGY

**This project includes a series of work divided between the 3 individuals. Each individual has its own role and responsibility which was carried out perfectly without any negligence.The C-BOT development include the number of reasonable hours and is not subject to copyright.The project development include 3 team mates whose duties are described a below:-**

**Ekaansh Sahni**

**His work include implementation of web framework in Django which is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.**

**Vikram Adytya**

**He was responsible for carrying out the making of code modules that includes the aspects like live corona updates, symptoms and precautions.**

**Karan Thakur**

**He was responsible to design the login and signup for the C-BOT interface. He also designed the image of bot and the background used in the C-BOT.**

TECHNIQUE USED

**Django**

**Django is a python-based free and open-source web framework that follows the model-template-views (MTV) architectural pattern. It is maintained by the Django software Foundation (DSF). Django primary goal is to ease the creation complex, database-driven websites. The framework emphasizes reusability and plug ability of components, less code, low coupling, rapid development. Django also provides as optional administrative create, read, update, delete interface that is generated dynamically through introspection and configured admin models.**

**HTML5**

**Hypertext Markup Language (HTML) is the standard mark up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browser service HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items**

**Python3**

**Python is an interpreter, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.**

**Beautiful Soup4**

**Beautiful Soup is a Python package for parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping. Beautiful Soup was started by Leonard Richardson, who continues to contribute to the project, and is additionally supported by Tide lift, a paid subscription to open-source maintenance. It is available for Python 2.7 and Python 3.**

**JavaScript**

**JavaScript often abbreviated as JS, is a programming language that conforms to the ECMA Script specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. Java Script enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behaviour, and all major web browser shave a dedicated JavaScript engine to execute it. JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually Node.js. They are also embedded in a variety of applications created with frameworks such as Electron and Cordova.**

**CSS**

**Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.**

**Bootstrap4**

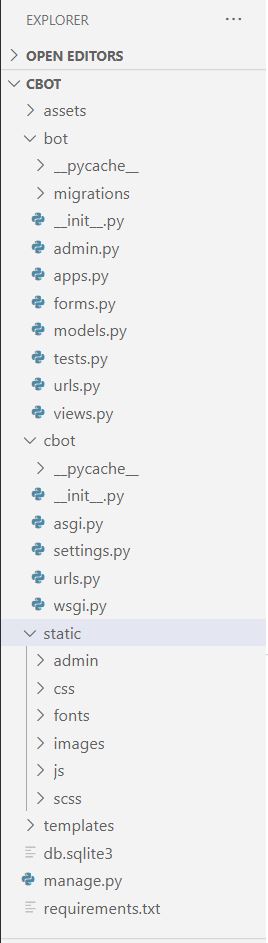
**Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.**

**Postgre SQL**

**Postgre SQL also known as Postgre, is a free and open-source relational database management system (RDBMS) 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. In 1996, the project was renamed to Postgre SQL to reflect its support for SQL. After a review in 2007, the development team decided to keep the name Postgre SQL and the alias Postgre.**

**Postgre SQL features transactions with Atomicity, Consistency, Isolation, Durability (ACID) properties, automatically updatable views, materialized views, triggers, foreign keys, and stored procedures. It is designed to handle a range of workloads, from single machines to data ware houses or Web services with many concurrent users. It is the default database for mac OS Server, and is also available for Linux, FreeBSD, Open BSD, and Windows.**

**CODE SCREENSHOTS**

**Project Hierarchy -**

**Our project name is cbot and it contains –**

1. **Assets folder**
2. **Bot folder**
3. **Cbot folder**
4. **Static folder**
5. **Templates folder**
6. **Db.sqlite3 file**
7. **Manage.py**
8. **Requirements.txt**

**The static folder contains all the images, css, fonts, js, scss needed for the project. By the static folder assets folder is generated by django when we collect static through commands.**

**Templates folder contains all the HTML files which are needed to run the web app.**

**Db.sqllite3 file has the database connection.**

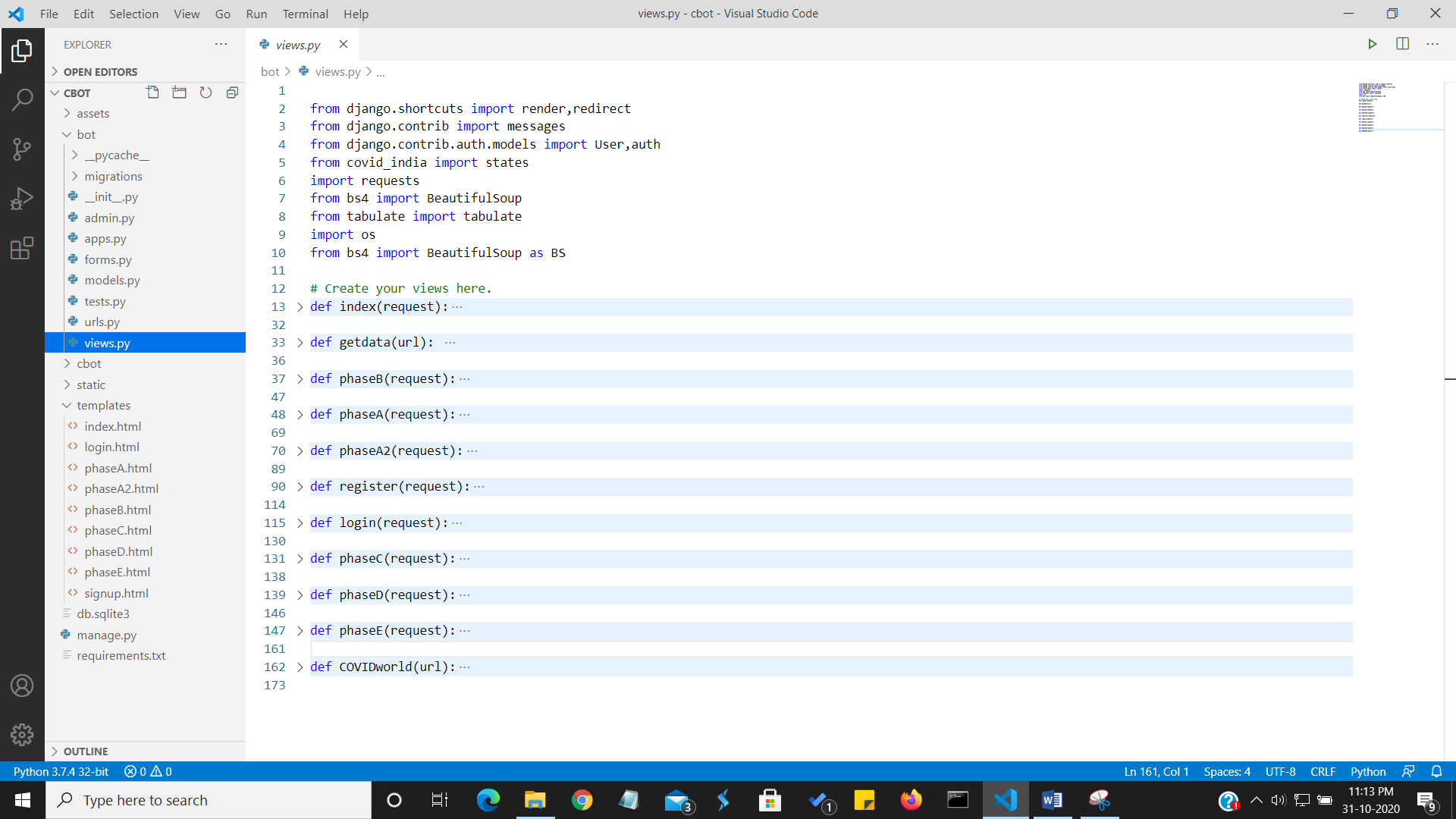
**Manage.py is a django generated python file to control the project.**

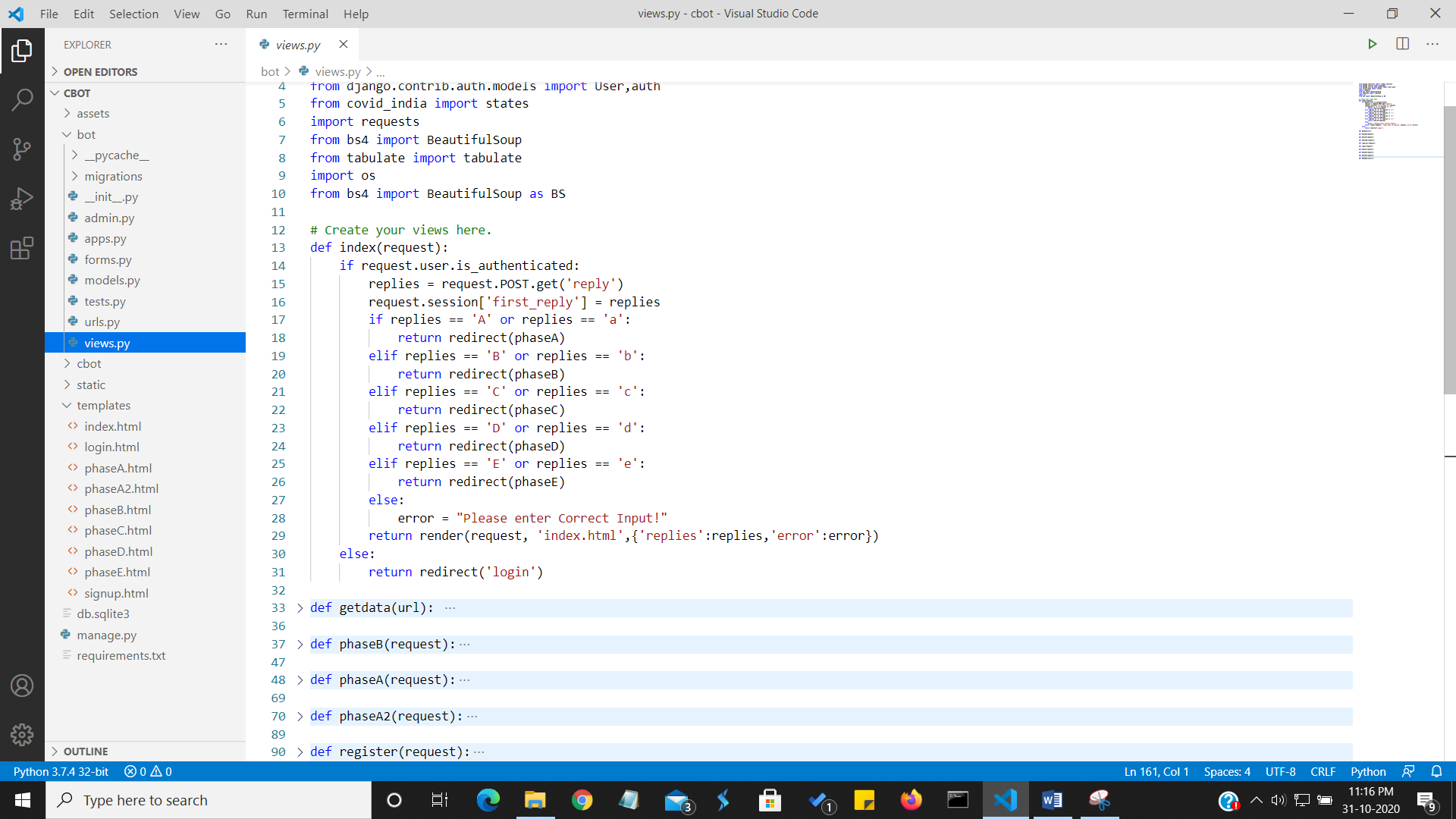
**Cbot folder contains the main urls.py and settings.py of the project.**

**We created the bot app from django and django auto generated the bot folder. The bot main code is present**

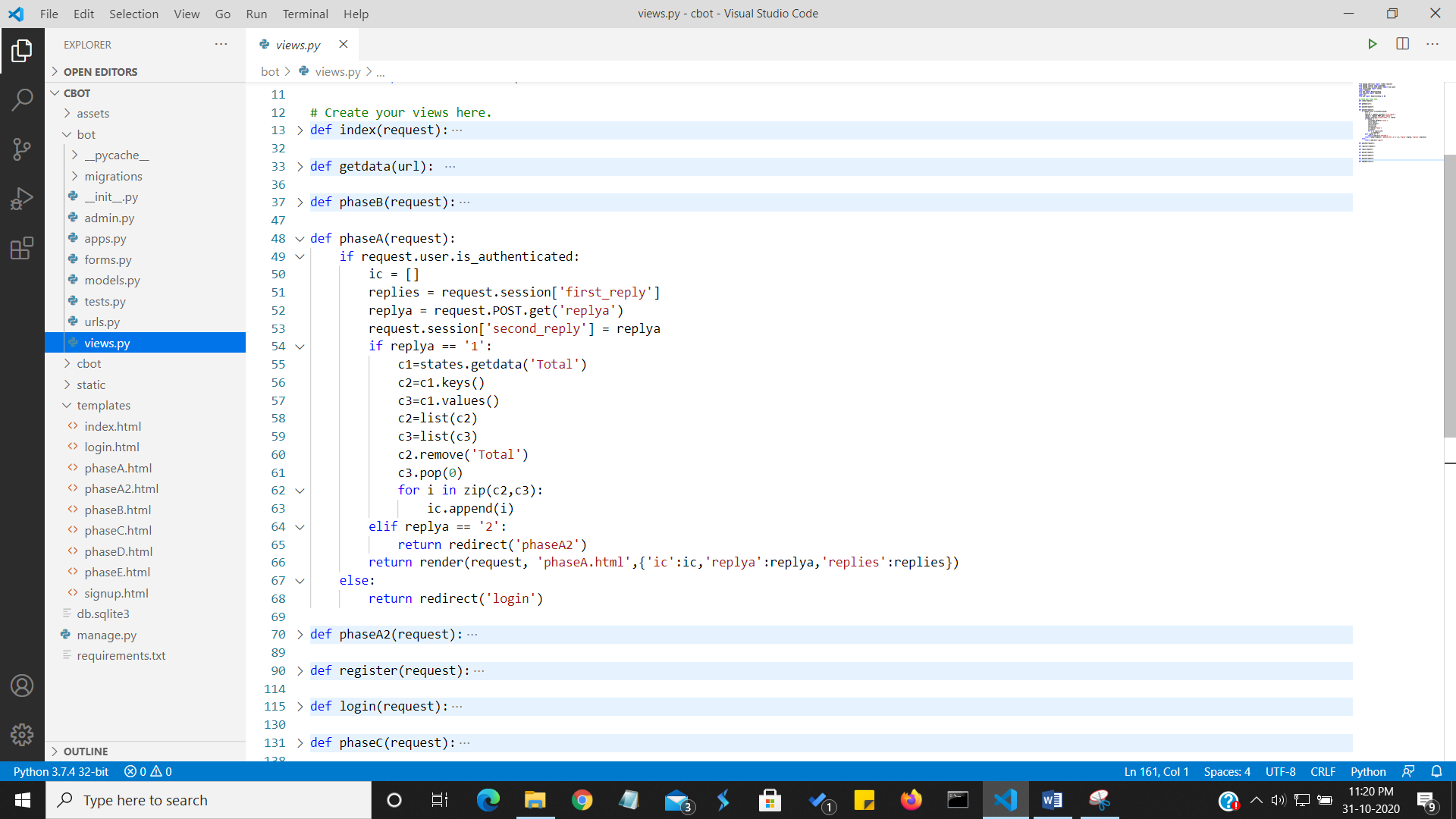
**In the views.py file.**

**Views.py file along with the template folder hierarchy –**

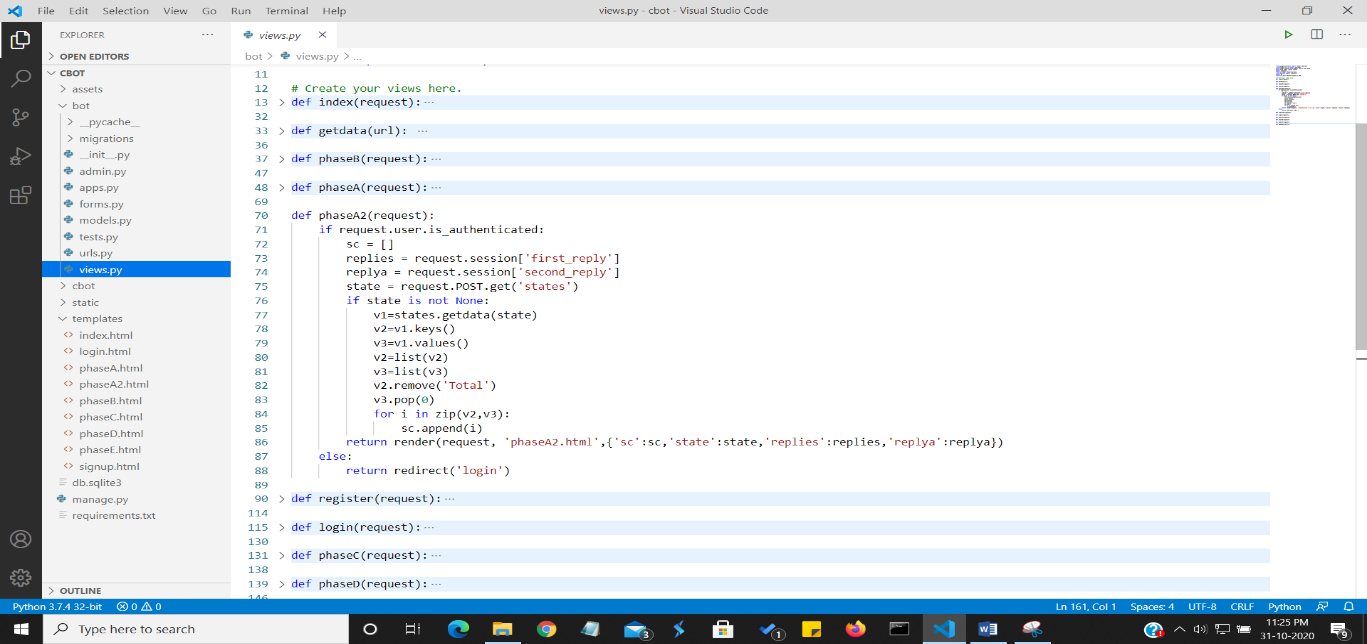


**This is the overall views.py folder. On the left hand side, you can see the templates folder containing all the html files. Now comes screenshot of each function declared in views.py –**

**This index function is our main function here and is linked to the index.html file. On the index.html the user is asked to choose a module and according to the user input the webpage redirects itself to another function (p.s. I have called them phases). Suppose a user enters A in web app then he gets redirected to phaseA.html having the phaseA function working. Of course, we check in the starting that the user is logged in to the bot or not, If he is not authenticated, we send him back to the login page.**

****

**This is the phaseA function which is called if the user enters A i.e. He chooses our first module -** Get Live Updates of Covid-19 Virus in India including details about any state!

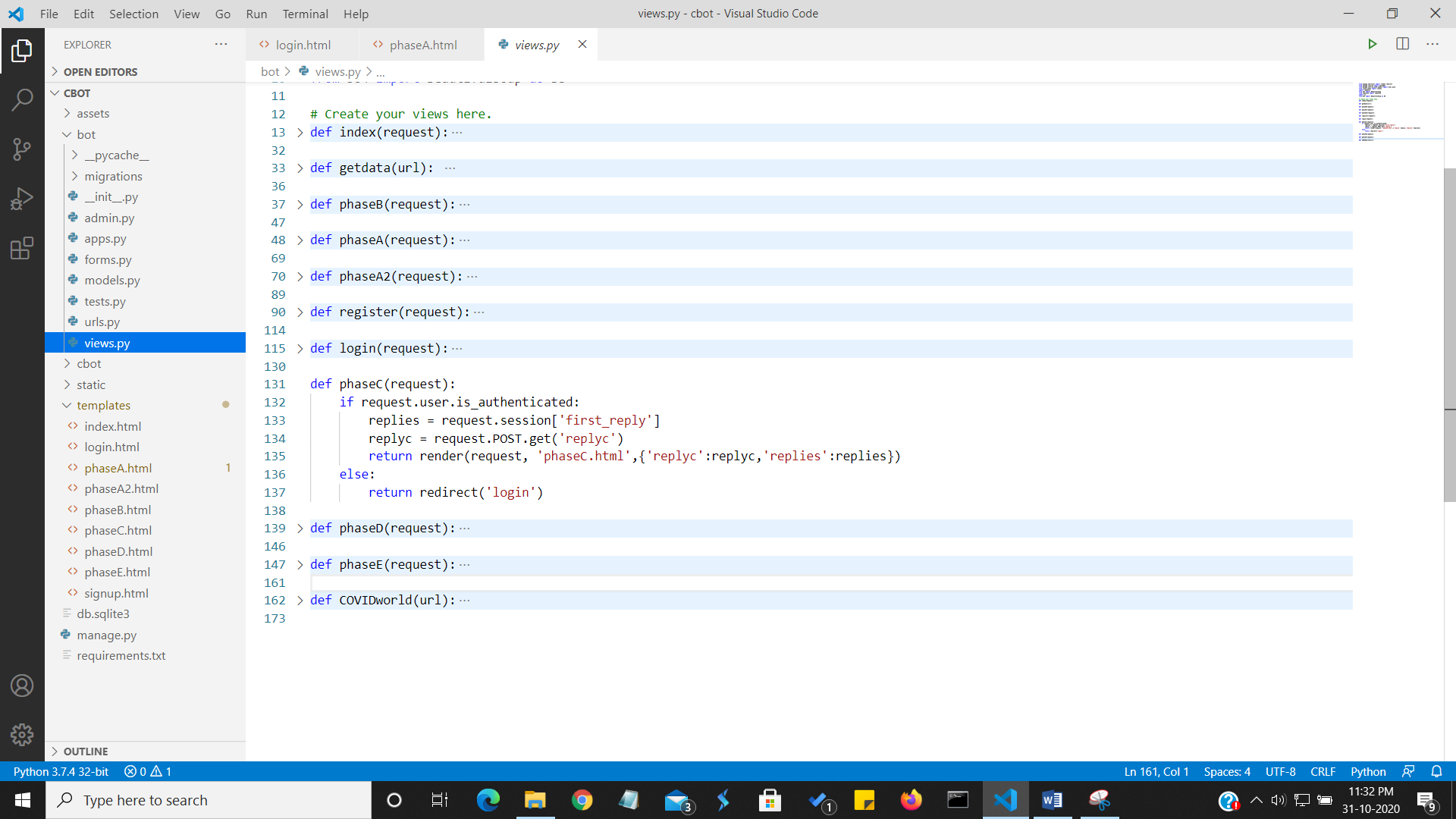
**Here we store his choice in the replies variable and then make a session variable first\_ans and store it in the database for future use. In this function the user is further asked to choose between two smaller modules. If he choose the first module, He gets the desired result and if he chose the second module, he gets redirect to further phaseA2.html or you can say phaseA2 function in the backend.**

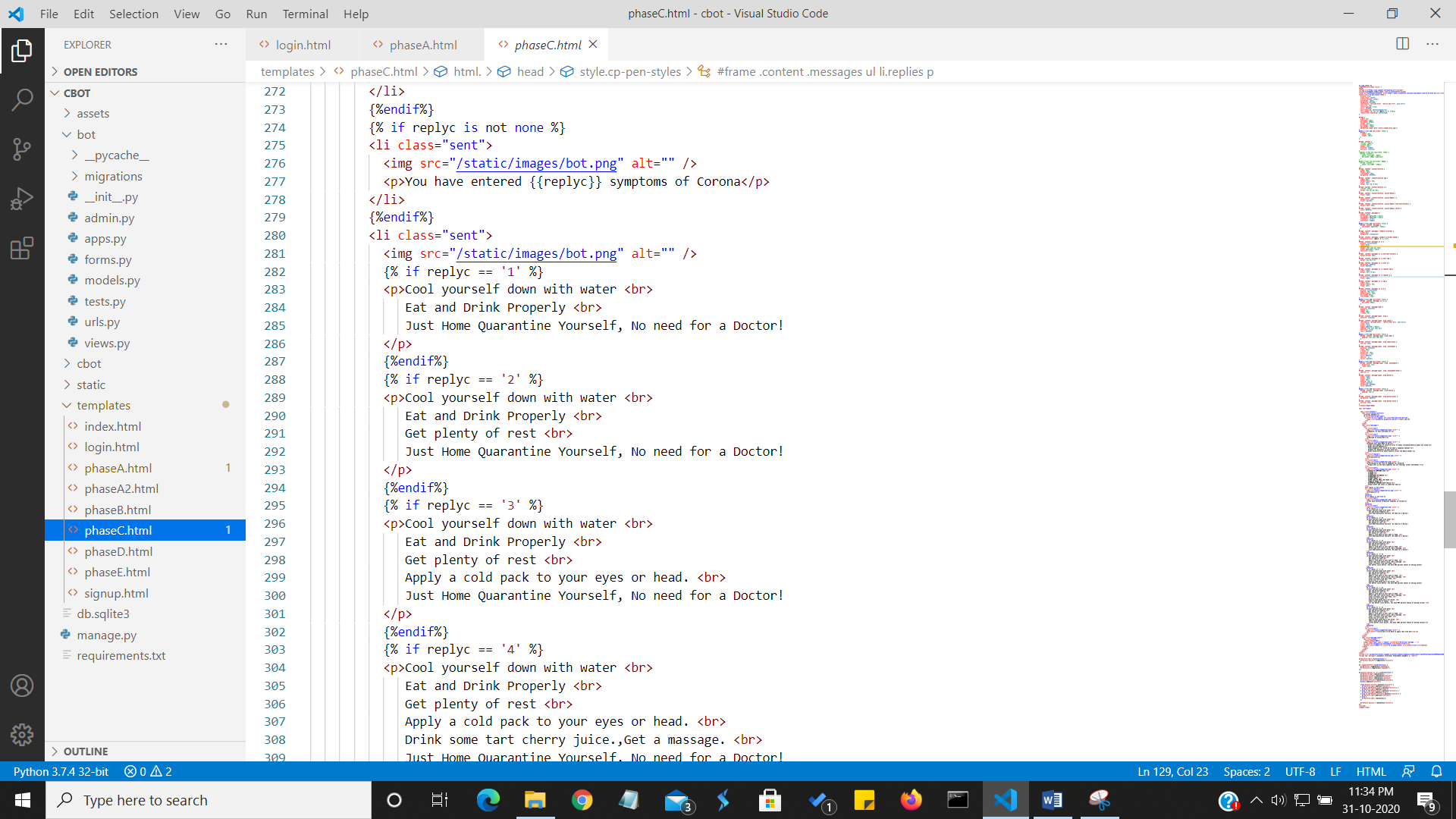
**In this function, We first get the inputs which user already gave in previous functions through sessions and pass it to html file. Then we use the covid-india library to get the desired result and storing it in list sc. Then we pass this list to the html file.**

****

**This is our Module B i.e. Get Vaccine Updates. We first check for user authentication then we proceed further and store user input into session. We then use the BeautifulSoup Library to Web- Scrap the data from website and store it in our res list to pass it to HTML file.**

**We then move to our third module(C) i.e. Symptoms Checker. In this we mainly just handle the input and output in the function. We used Jinja format to put the code in the HTML file.**

****

****

**To dynamically change data in Web App, we are utilizing the Jinja2 Format in HTML file to use the python code. The code simply checks the input number and prints precautions and advice according to that.**

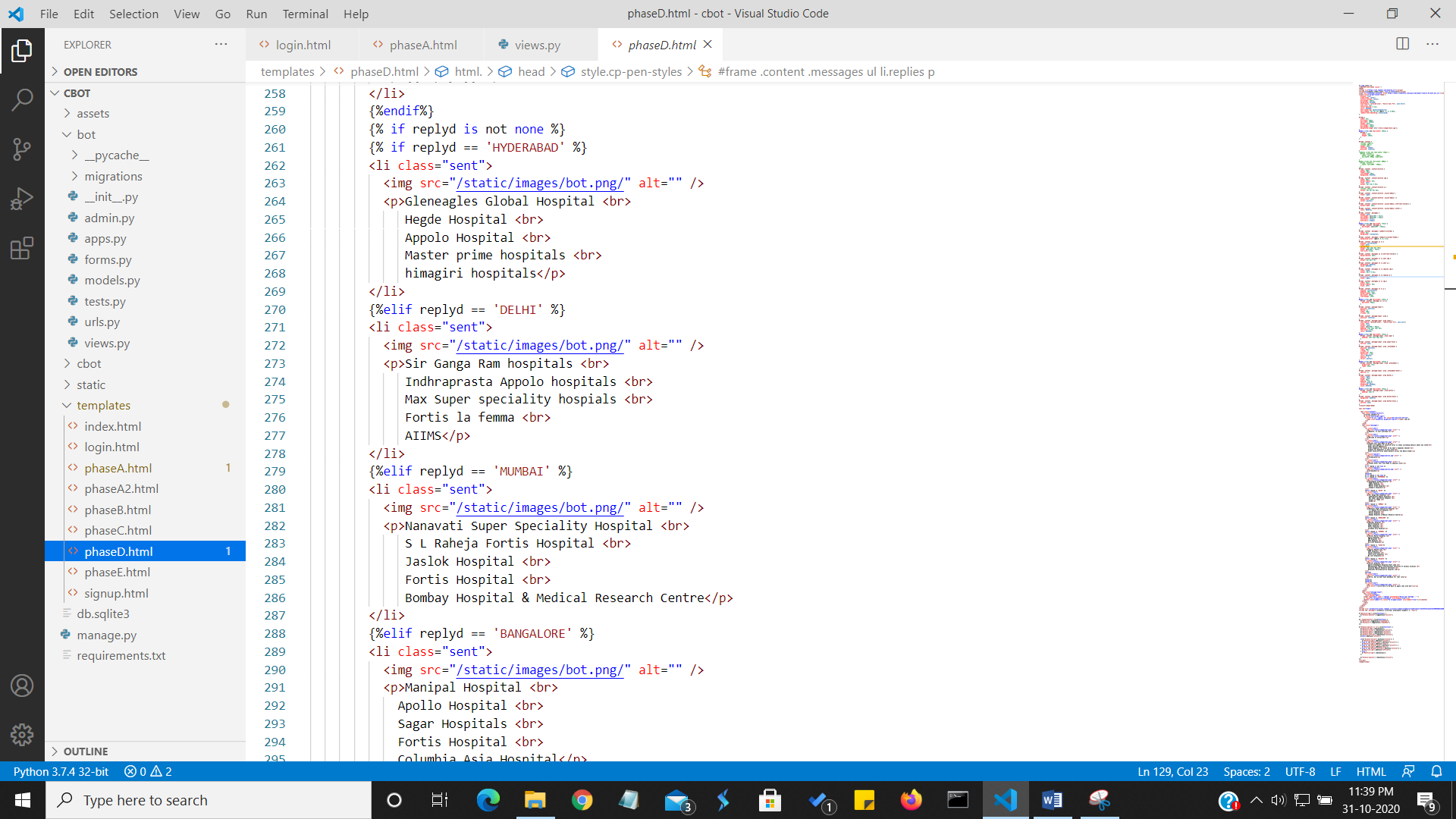
**We move on to D module I.e. Check for Hospitals in your city.**

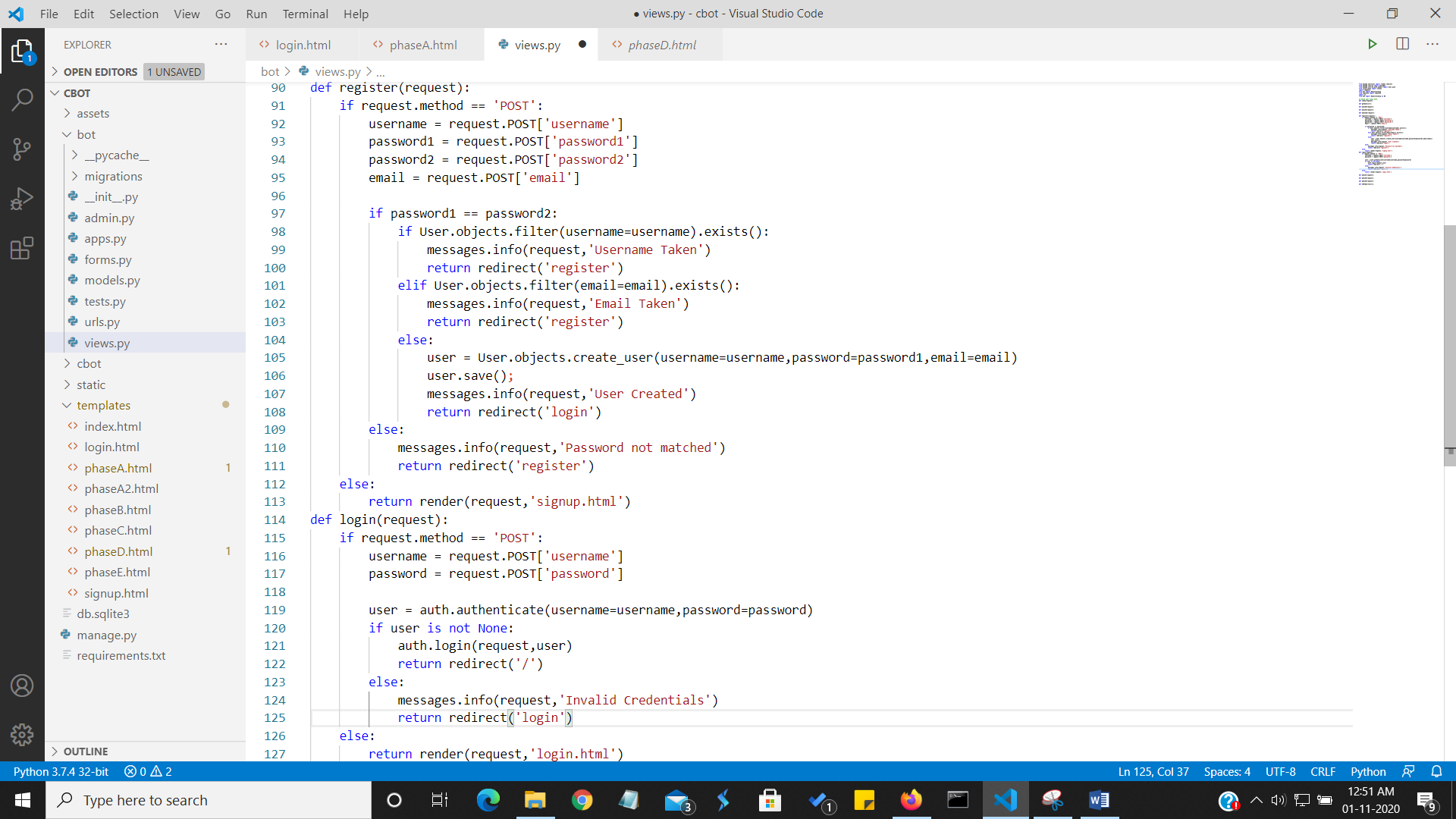
**Doing this module was a bit tricky as we earlier thought to use Google’s Nearby Search API to automatically web scrap Hospitals data from Google Maps but we needed funds to create a billing account in Google so we aborted this idea and went to a manual approach.**

**We manually added few hospital data of some metropolitan cities and print those results on user Input.**

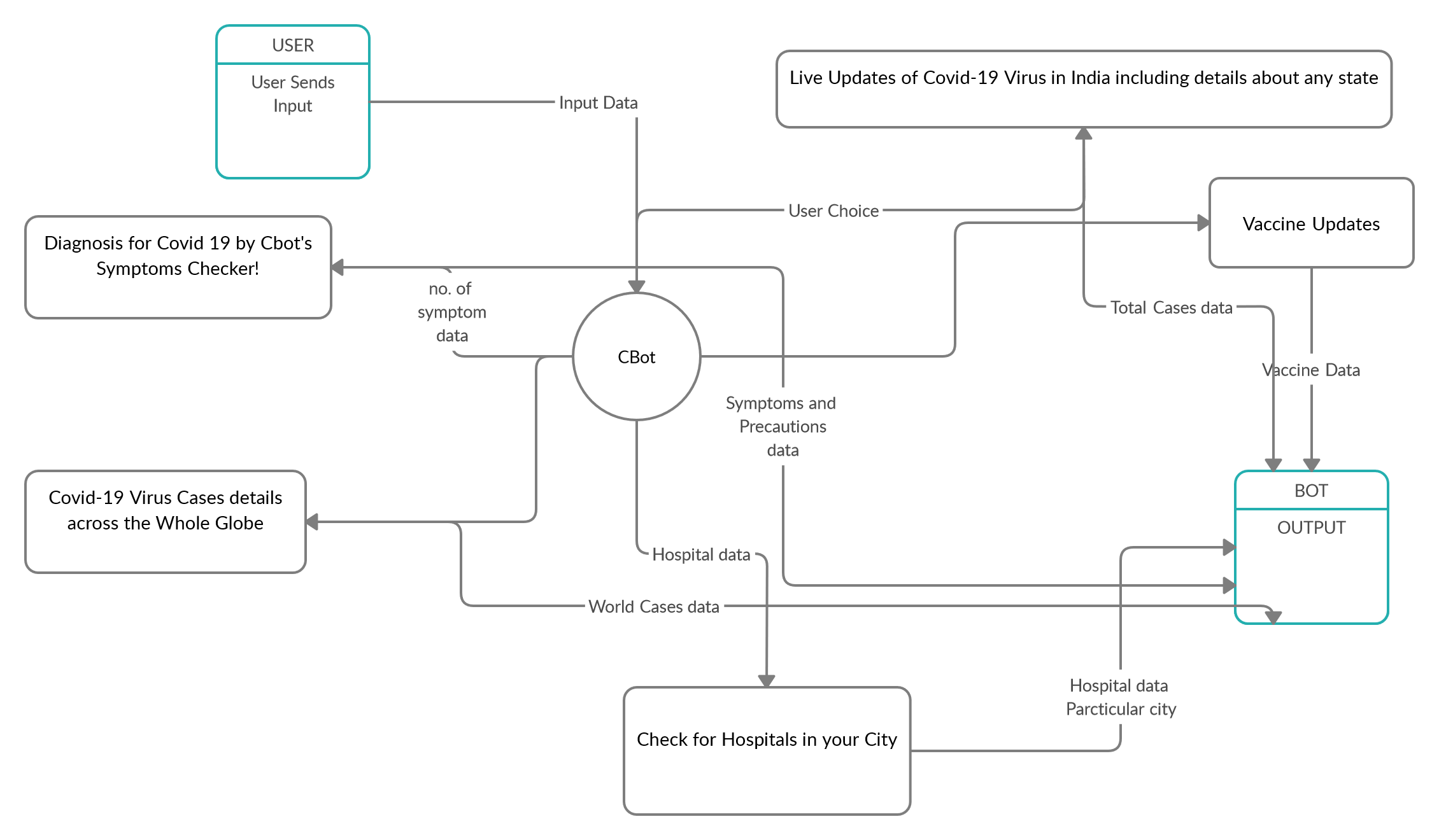
**The phaseC and PhaseD functions are nearly same, most part is done on the HTML file by using Jinja to integrate Python codes.**

**Our Last Module(E) i.e. Get World Data of Covid-19. We did this by again using web scraping to get the current active, total and death cases all over India.**

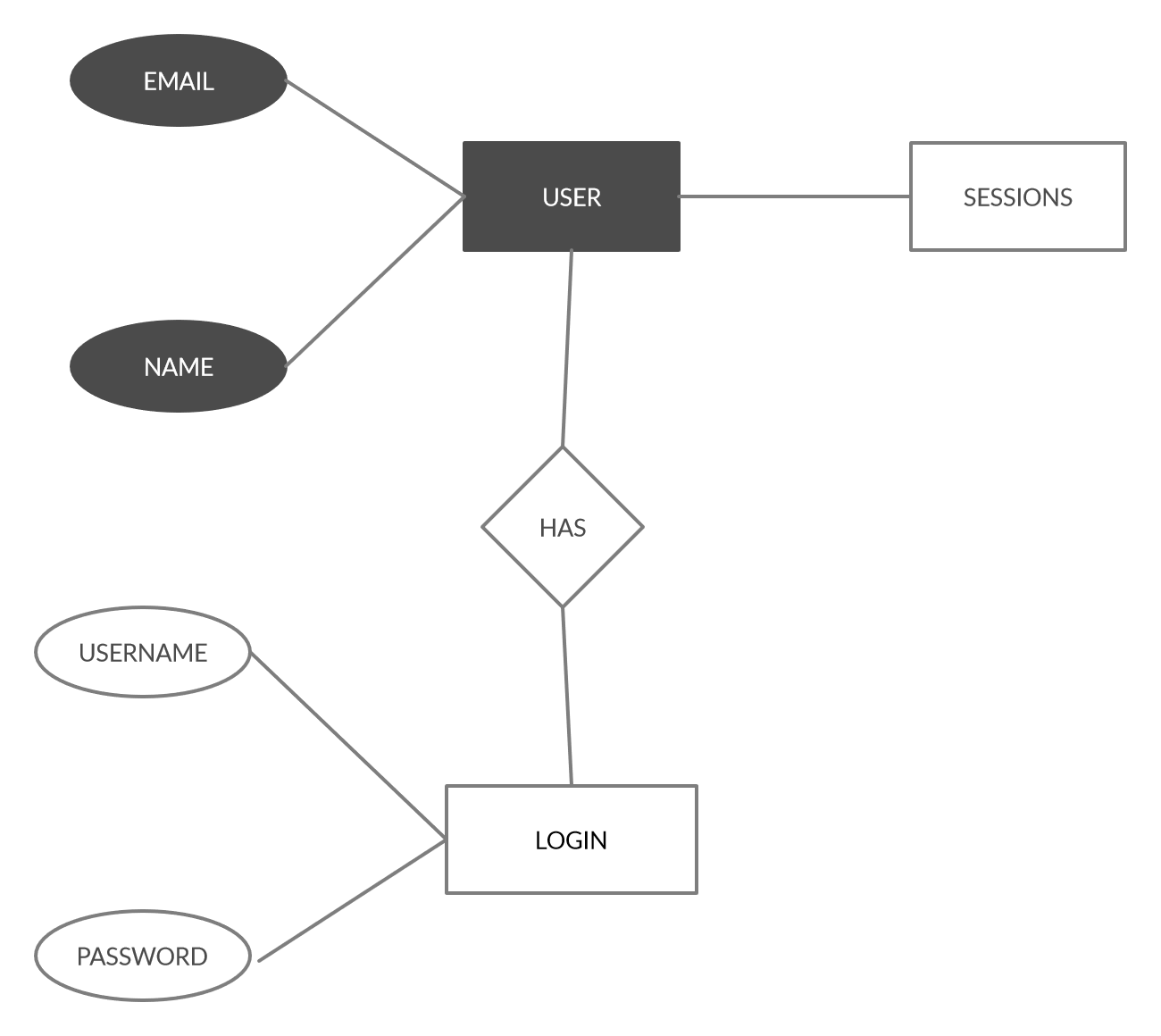
**  
**

**After that we have the login function where we use the User table created in Postgre SQL automatically in django to check for login and registration.**

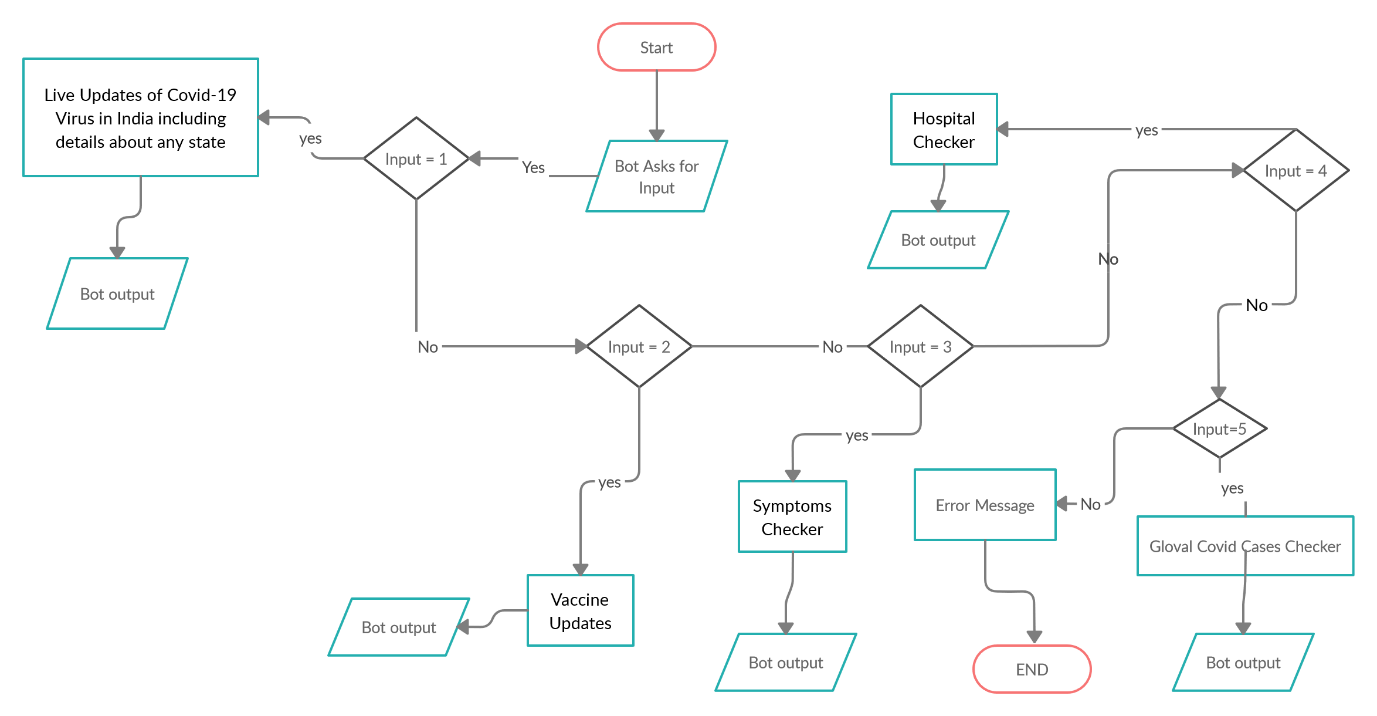
**DATA FLOW DIAGRAM**

****

**ER DIAGRAM**

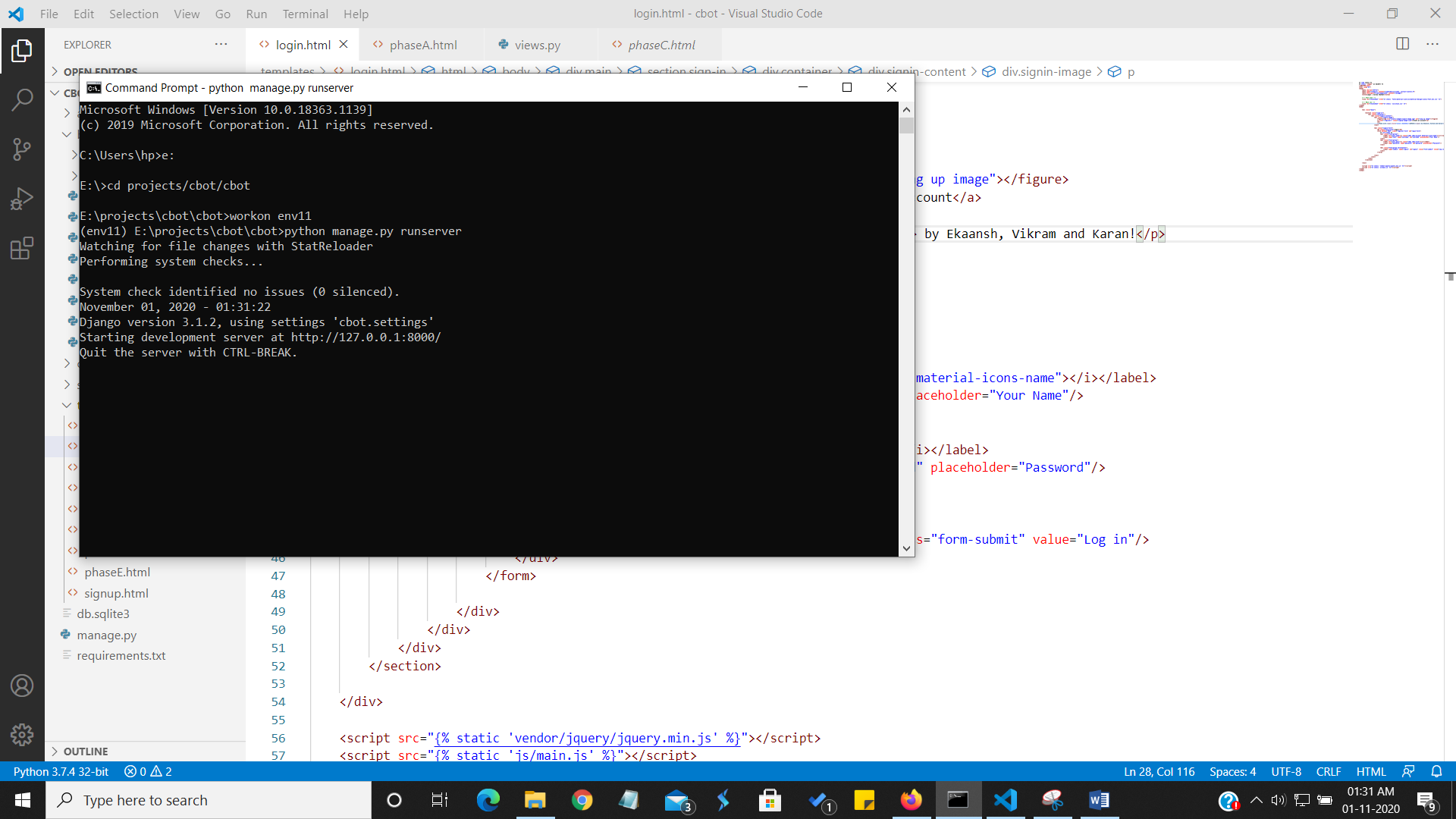
****

**FLOWCHART**

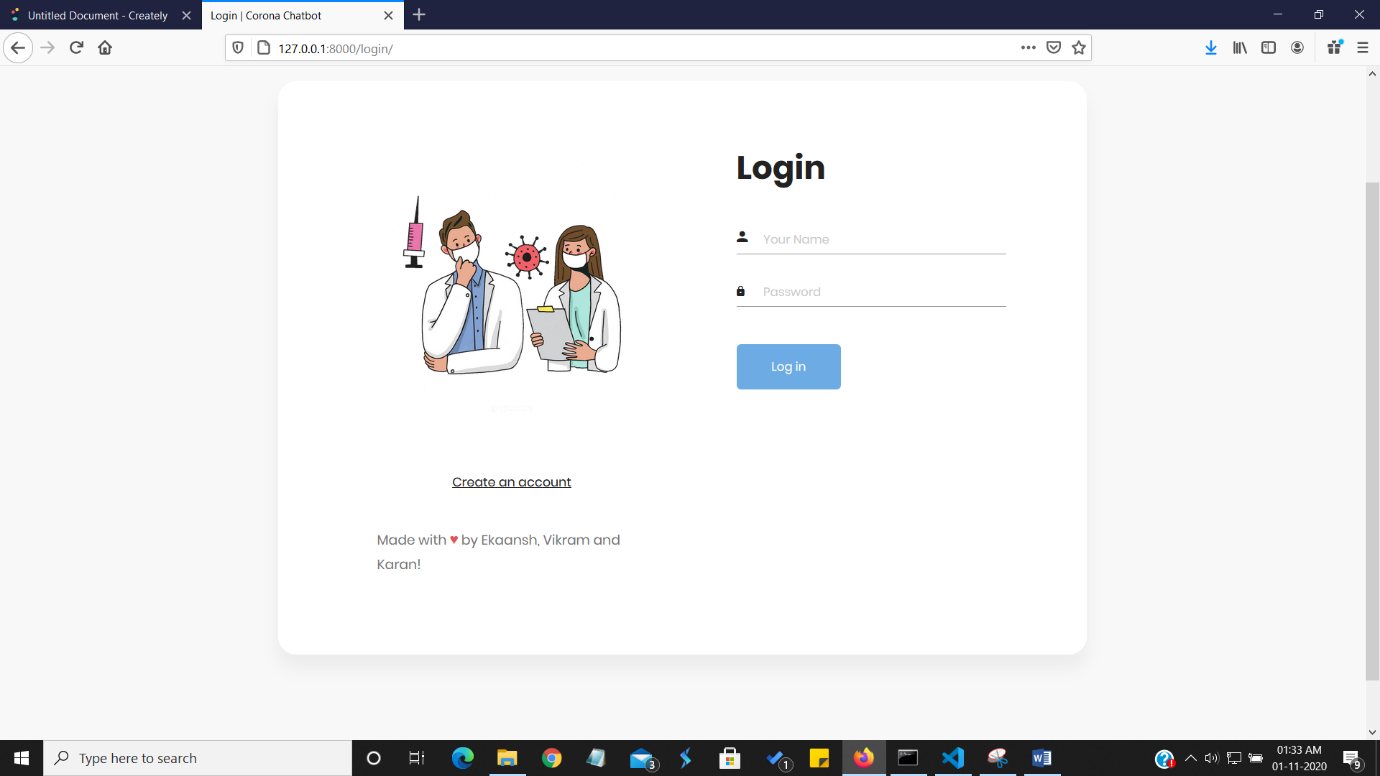
****

**OUTPUT SCREENSHOTS**

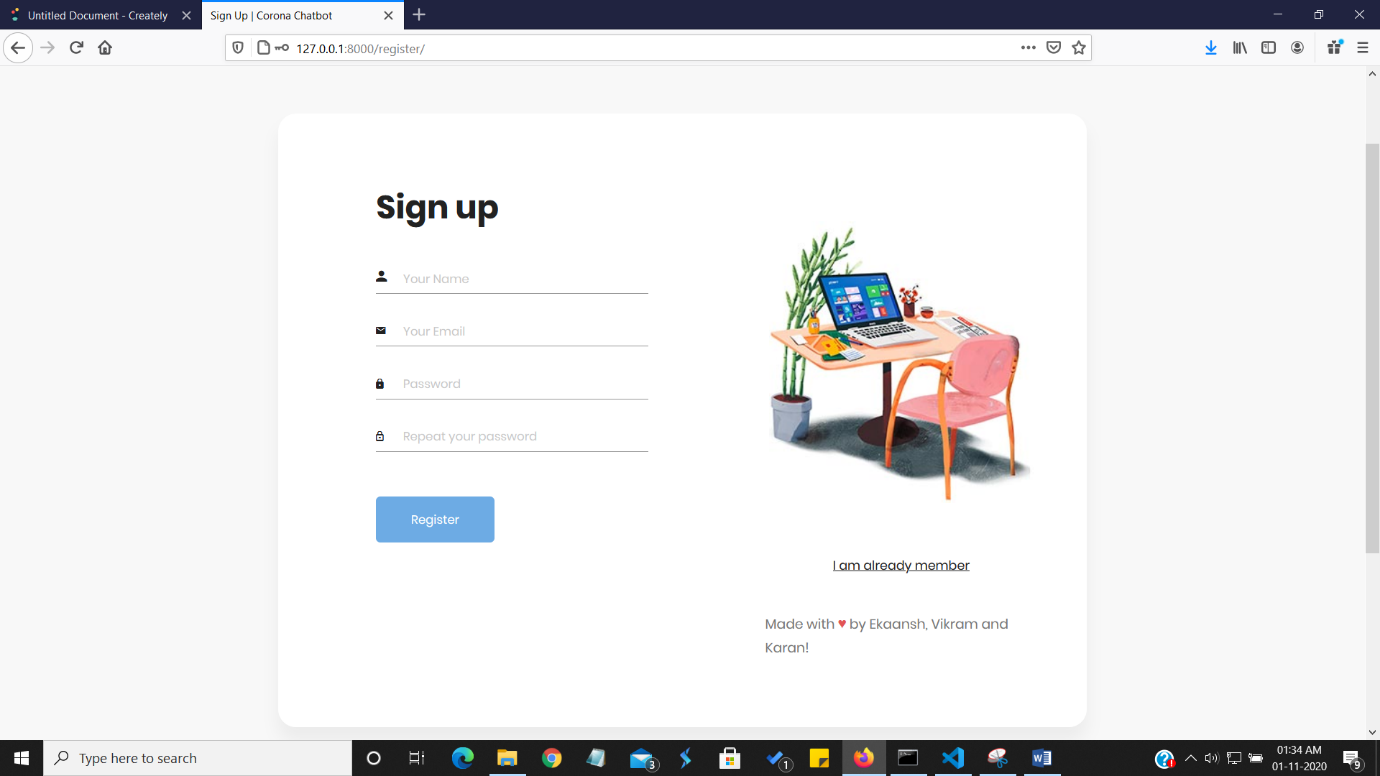
**Server Run --**

****

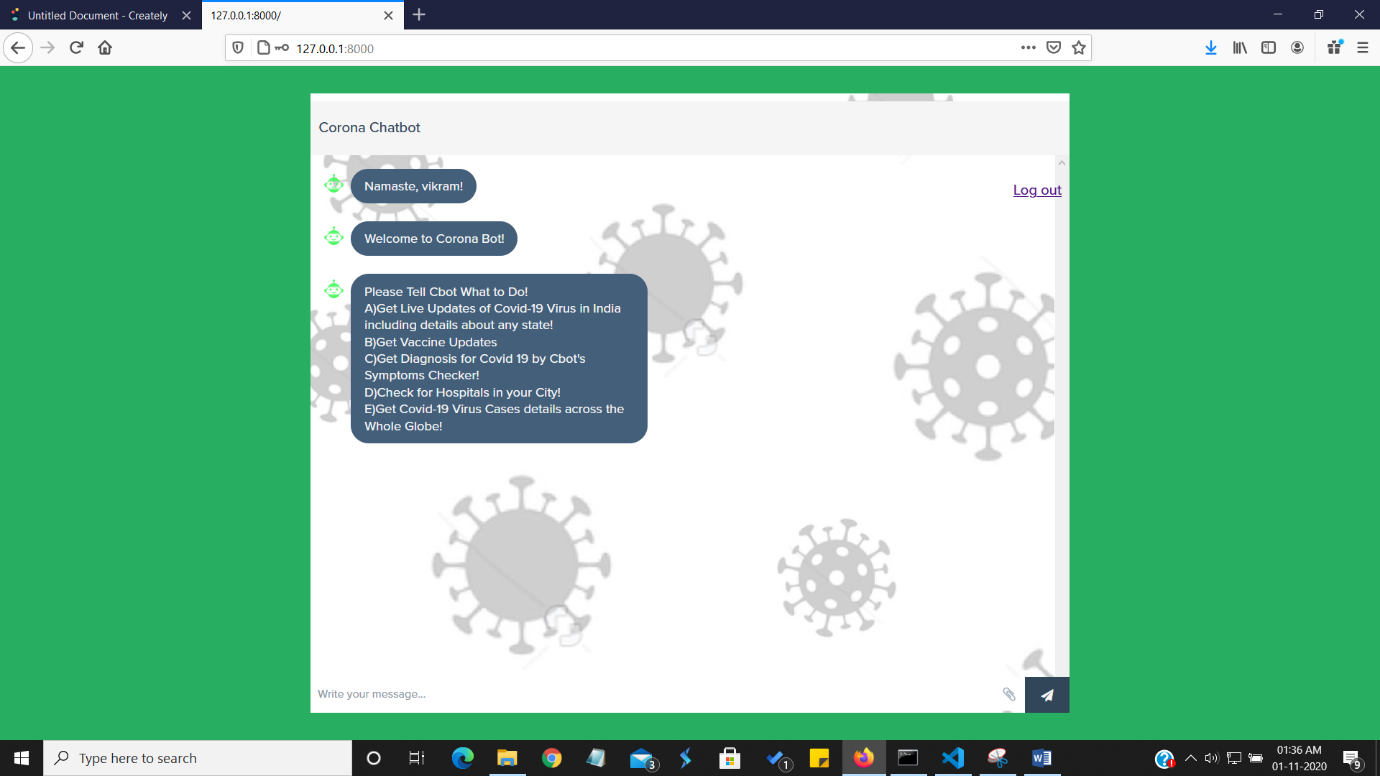
**Login --**

****

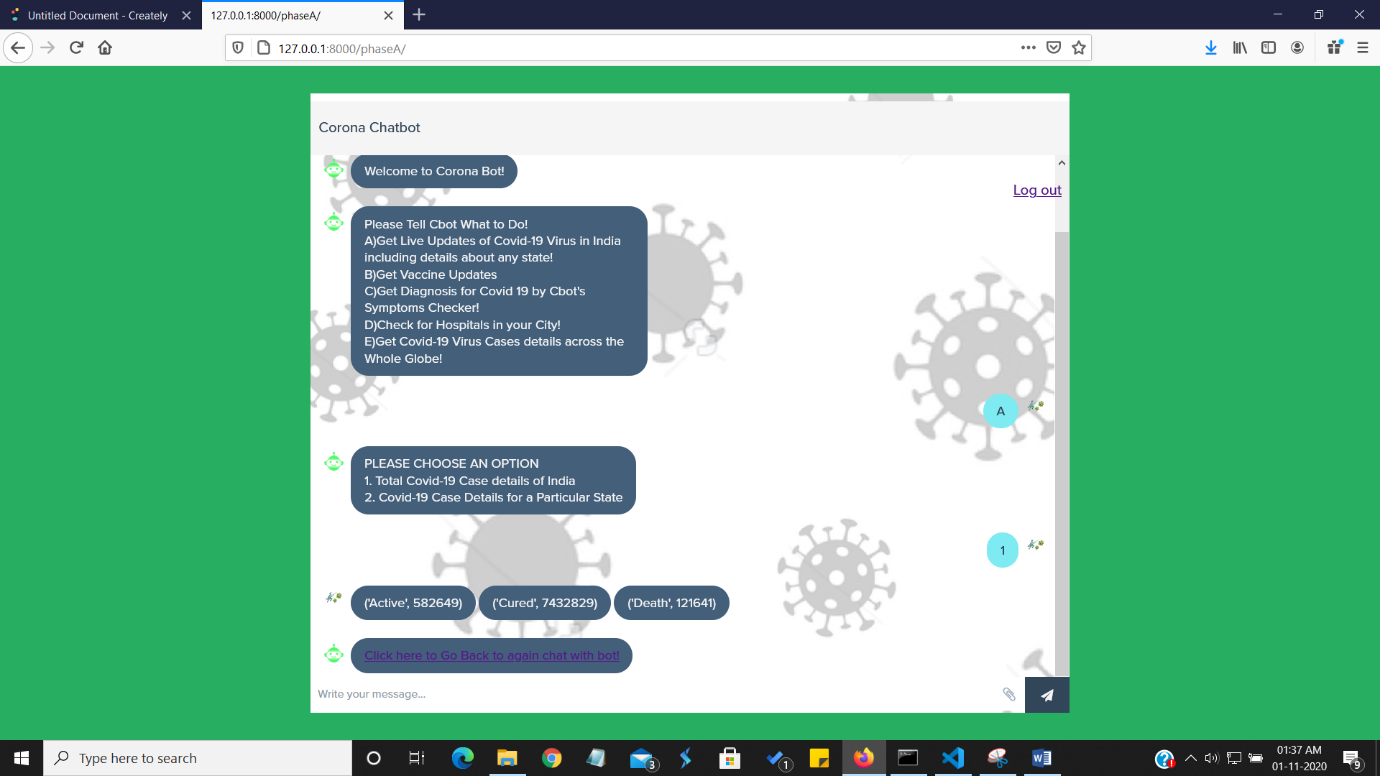
**Signup --**

****

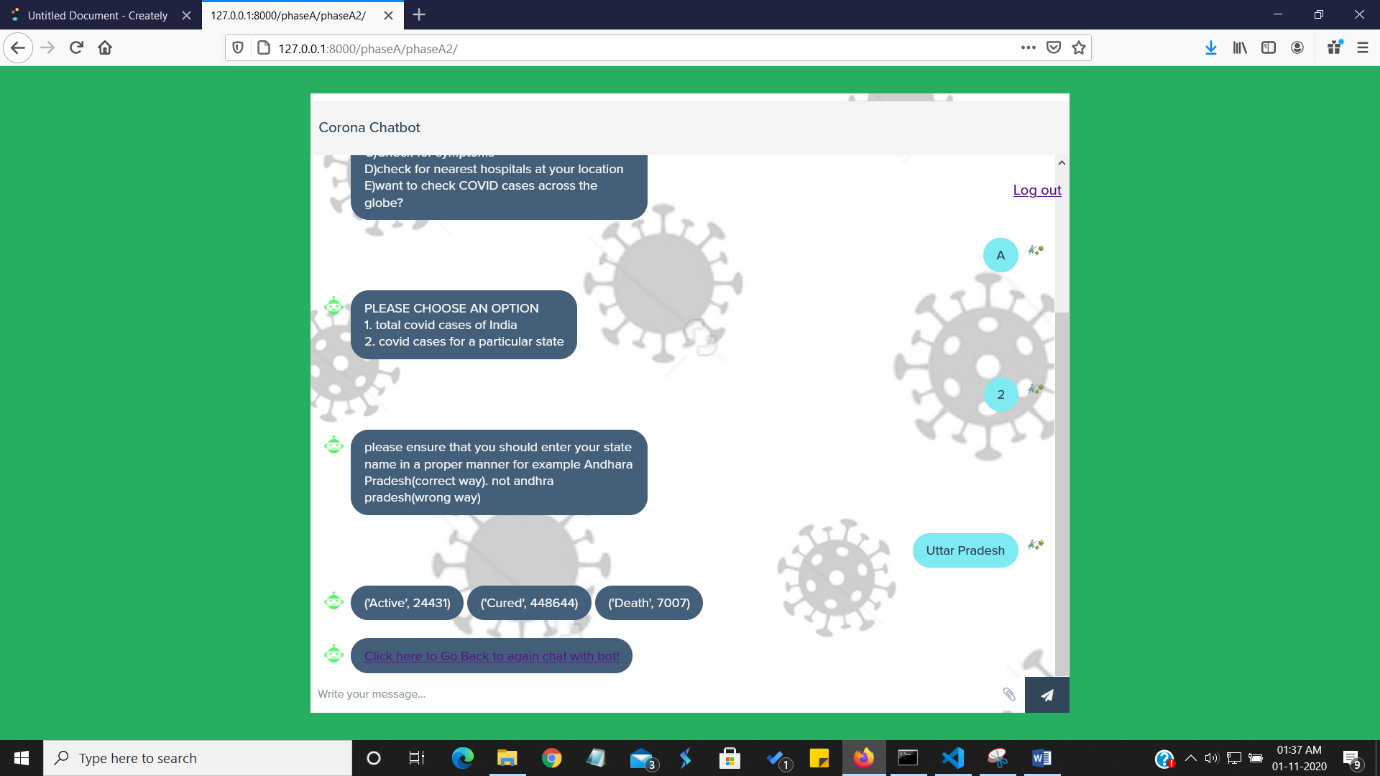
**Index --**

****

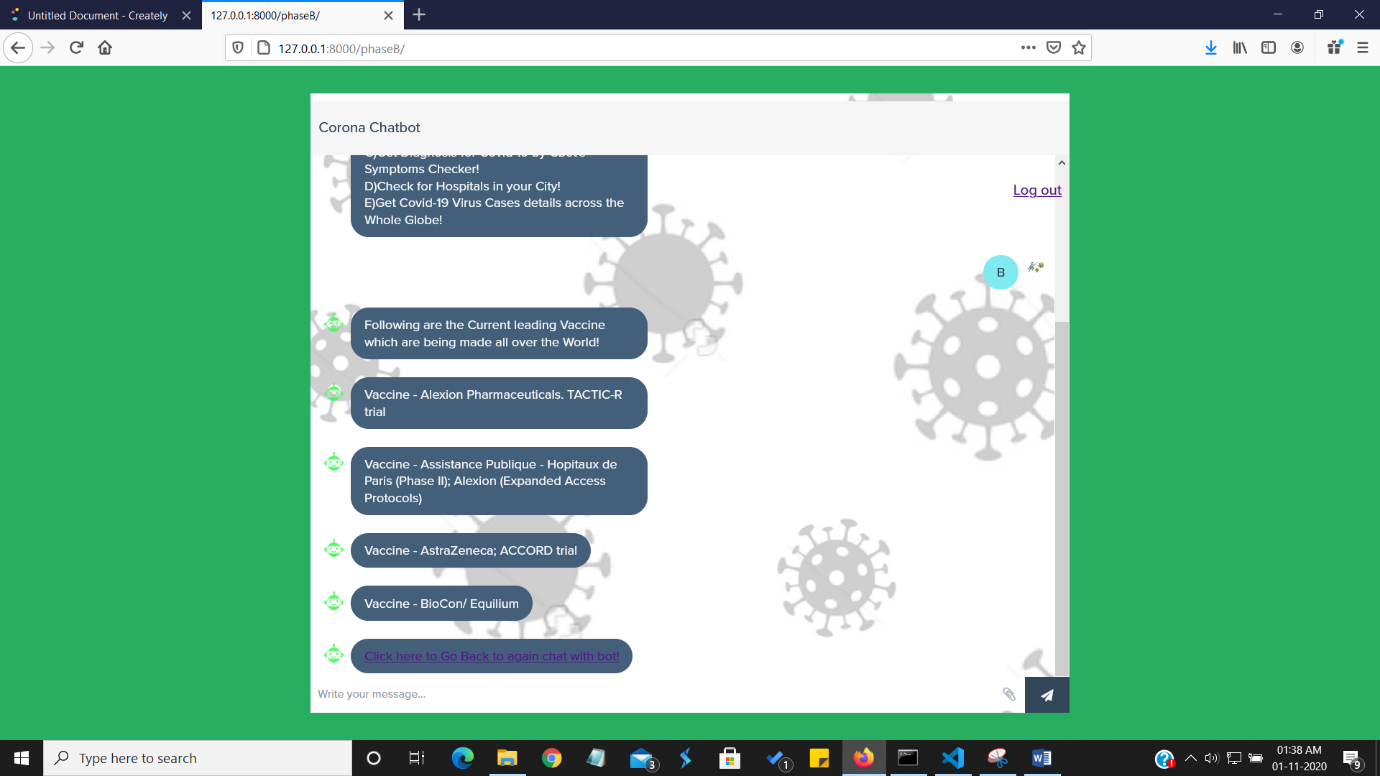
**Phase A --**

****

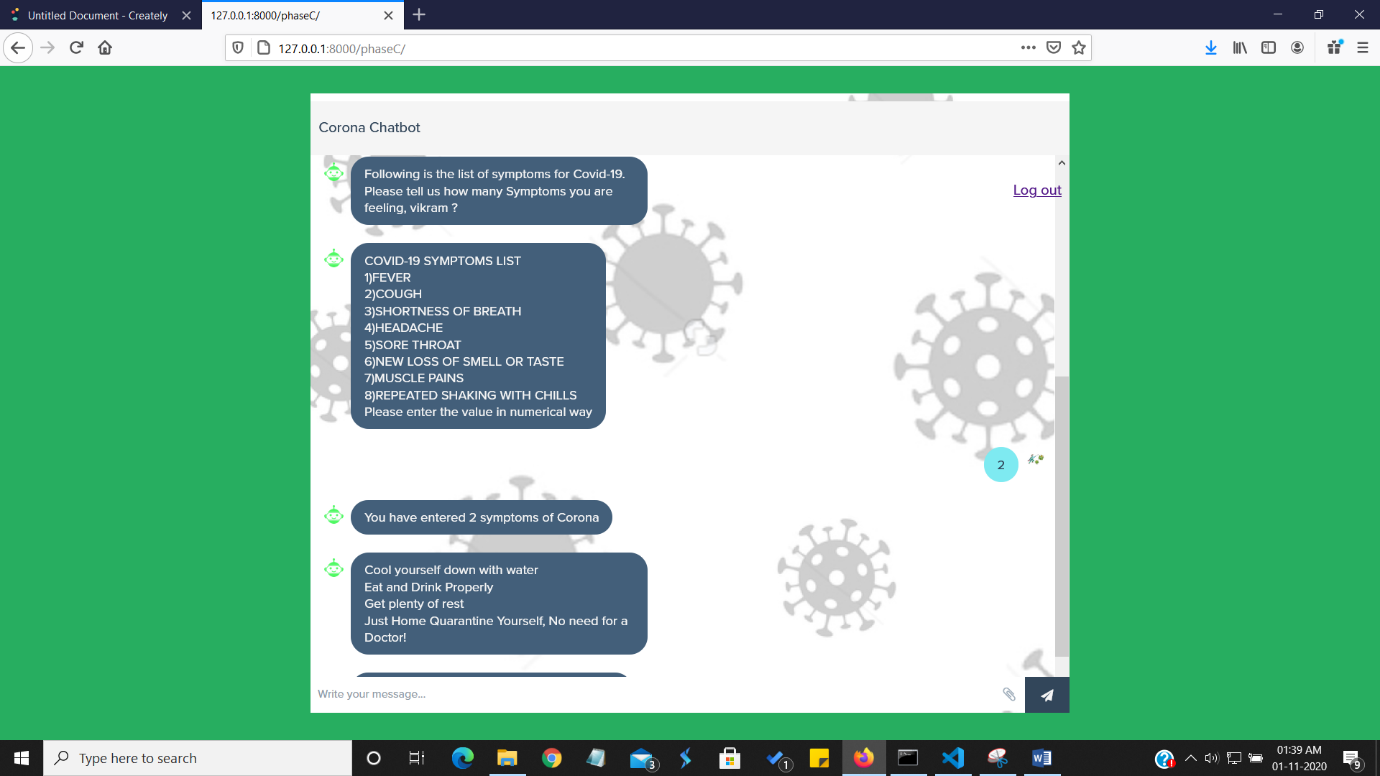
**Phase A2 -**

****

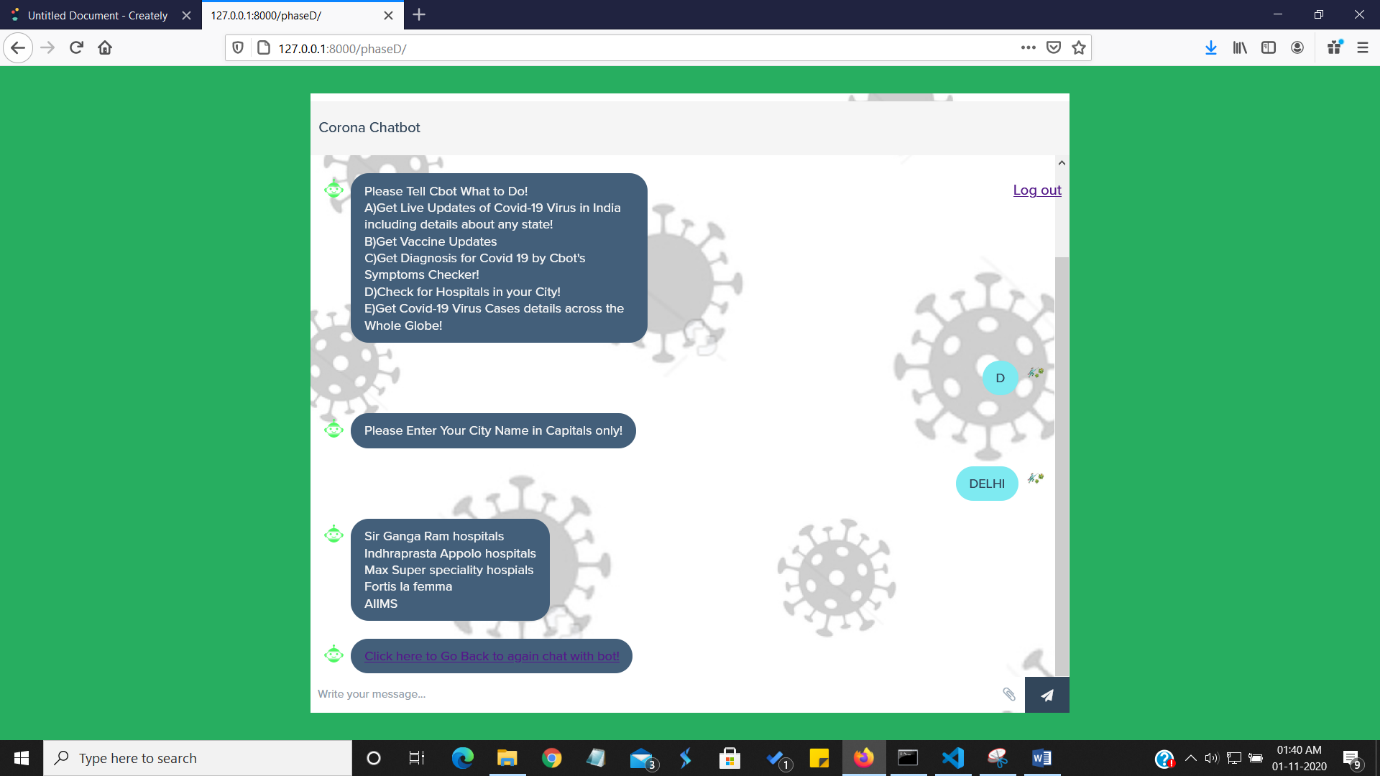
**Phase B --**

****

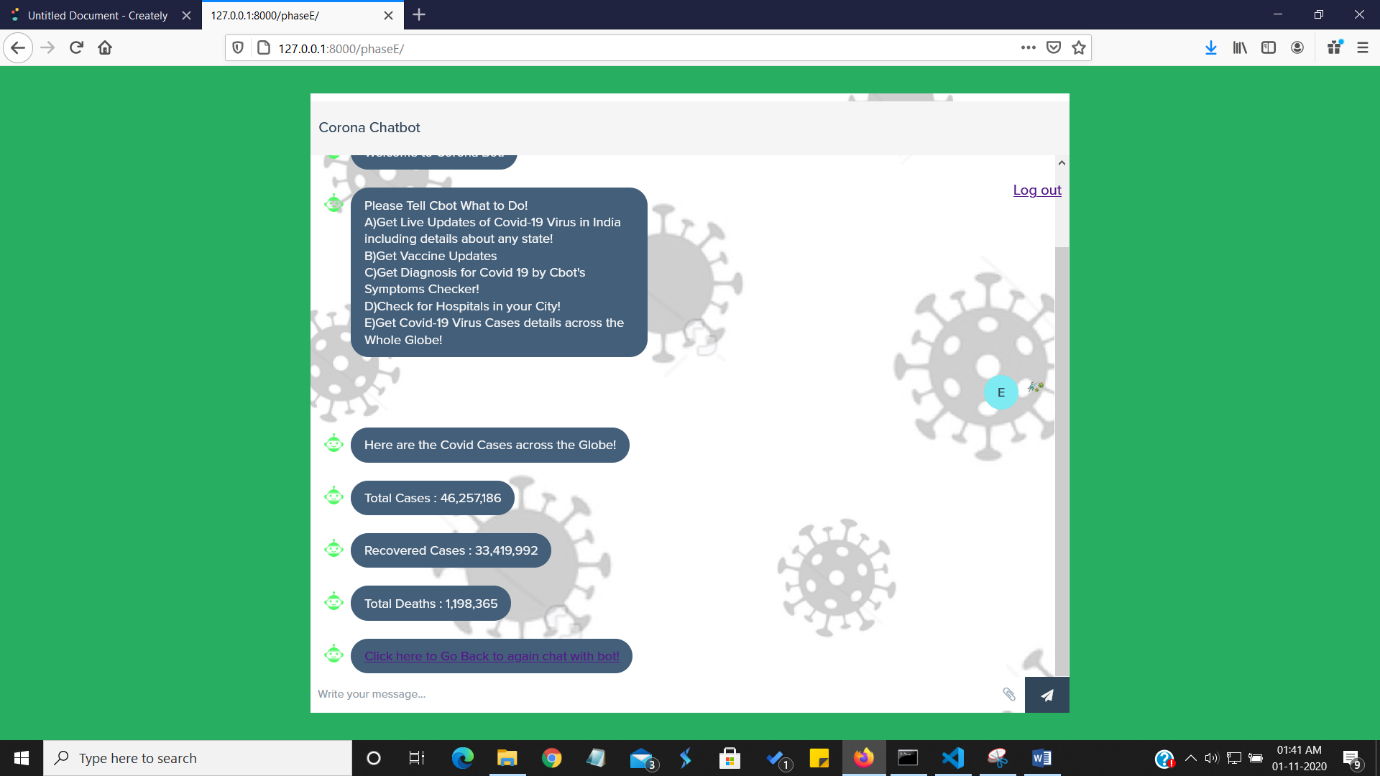
**Phase C --**

****

**Phase D --**

****

**Phase E --**

****

**Bibliography**

* **We learnt Beautiful soap form geeks for geeks website.**
* **We used a python library named ‘covid-India’ which tells us number of cases in India. We got this library from Mr.Goswami sir (github).**
* **We used python anywhere website to host our web application but that is in process as we need a billing account for that and we are short on funds.**
* **We took the reference of css, html, java script form Color lib website.**