

# HOPE Foundation

QUESTIONNAIRE TELEPHONIC/DIRECT
---------------------------------

## SECTION I: GENERAL INFORMATION

1. **Name & residential address of the student:** Mohammed Khurram  
#1839, 6<sup>th</sup> Main Road HBR Layout 3<sup>rd</sup> Block, Bangalore - 560043
2. **Contact Phone Number:** 9740607632
3. **Name & Address of College:** St. Aloysius Degree College, Bangalore -560001
4. **Degree UG/PG:** BCA
5. **Name of Parents /Guardian:** Mohammed Junaid
6. **Phone Number of Parent/Guardian:** 9741905913
7. **Occupation of Parents /Guardian:** Software Engineer
8. **Annual Income of Parents/Guardian:** 4.5 LPA
9. **Number of Dependents:** 3
10. **Have you dropped out of college:** No
11. **Percentage of Marks for last year:** 74.25

## **CASE STUDY**

### **FORMAT OF CASE STUDY**

#### **Intervention -**

**Q.** Why did you join the Python classes:

**Ans:** I attended the python classes as i was looking for language which is industry level and easy to understand too.

#### **Output -**

**Q.** How are you Feeling about your learning and how what influenced you?

**Ans:** I feel confident enough to enter into a corporate field related to python with all the knowledge I have grabbed from the online sessions. AI was the part of this programming language which had my eye on and that was one of my reasons to get influenced.

**Q.** Outcome?

How are you using the skills learned were you able to develop any program game/app etc?

**Ans:** Yes

**Q.** Impact?

**Ans:** The Impact from this course has been commendable for me as it helped me a lot knowing about python programming from a beginner level. I honed my communication and soft skills during the course.

**Q.** Pictures:

Good Quality Picture of the candidate Passport is also fine

Picture of Teacher teaching students (Practical if any) Zoom pictures

Picture of Class room Session (Theory if any) Zoom Pictures



**Onward Plans/Future Plans:**

Working in a product based company learn the culture and establishing my own startup

What is your dream for yourself.	To get placed in product based MNC with managerial position
What kind of support you need to fulfill your dream	Help me get placed in a company where I can refine my skills according to the industry standards.

**Any other Suggestions (Parents/Guardians/Students):****Comments by interviewer's:****Interview done by:****Name:****Date:****Time:****Place:**

Date	Batch - timing	Activity / Topics covered	Issues	Attendance	Remarks			
16/02/2023	6.00 - 7.00	Introduction to Syllabus and queries	No link and Id's	Good	Nil			
17/02/2023	6.00 to 7.30	Introduction and Installation started	Nil	Good	Nil			
18/02/2023	10.00 to 11.30	Real time applications and installed	Nil	Good	Nil			
20/02/2023	6.00 to 7.30	Explained Eduvahini portal and quiz	Nil	Good	Nil			
21/02/2023	6.00 to 7.30	Intro to Data structures- types, user defined DS	Nil	Good	Nil			
22/02/2023	6.00 to 7.30	Data structures- List and tuple	Nil	Good	Nil			
23/02/2023	6.00 to 7.30	Login credentials, resume, forms	Nil	Good	Nil			
24/02/2023	6.00 to 7.30	Regular expressions and Modules in python	Nil	Good	Nil			
25/02/2023	6.00 to 7.30	Project intro and 20 codes with output	Nil	Good	Nil			
27/02/2023	6.00 to 7.30	GUI application in python with code	Nil	Good	Nil			
28/02/2023	6.00 to 7.30	Interview Skills and LinkedIn Access Webinar	Nil	Good	Nil			
01/03/2023	6.00 to 7.30	Datastructures- Dictionary and Sets	Nil	Good	Nil			
02/03/2023	6.00 to 7.30	case study and feedback document submission	Nil	Good	Nil			
03/03/2023	6.00 to 7.30	libraries in python , Modules in python	Nil	Good	Nil			
04/03/2023	6.00 to 7.30	Control Statements in python with codes	Nil	Good	Nil			
06/03/2023	6.00 to 7.30	Submission on Eduvahini CH.1 Assignment	Nil	Good	Nil			
08/03/2023	6.00 to 7.30	Object oriented programming in python	Nil	Good	Nil			
09/03/2023	6.00 to 7.30	Function handling and web servers, database apps	Nil	Good	Nil			
10/03/2023	6.00 to 7.30	File handling Mechanism and Web servers	Nil	Good	Nil			
11/03/2023	6.00 to 7.30	Functional Handling with coding examples	Nil	Good	Nil			
13/03/2023	6.00 to 7.30	Algorithms and Data structures with codes	Nil	Good	Nil			
14/03/2023	6.00 to 7.30	Exception, errors and Handling exceptions	Nil	Good	Nil			
15/03/2023	6.00 to 7.30	Explained Assignments of 7th and 8th chapters	Nil	Good	Nil			
16/03/2023	6.00 to 7.30	Working on Mini project and first page of project	Nil	Good	Nil			
17/03/2023	6.00 to 7.30	Review on Assignments and Mini projects	Nil	Good	Nil			
18/03/2023	6.00 to 7.30	Review on Self Promotion Video and Viva	Nil	Good	Nil			
20/03/2023	6.00 to 7.30	VIVA preparation with Quizzes revision	Nil	Good	Nil			
21/03/2023	6.00 to 7.30	Corrections on project documentation	Nil	Good	Nil			
22/03/2023	6.00 to 7.30	Submission of 8 chapter assignments correction	Nil	Good	Nil			
23/03/2023	6.00 to 7.30	Advanced libraries - Scikit learn and matplotlib	Nil	Good	Nil			
24/03/2023	6.00 to 7.30	Revision on 9,10th chapters assignments	Nil	Good	Nil			
25/03/2023	6.00 to 7.30	Interview Questions on Python technical questions	Nil	Good	Nil			
26/03/2023	6.00 to 7.30	Interview Preparation for Ofxtrade Selected Candidates	Nil	Good	Nil			
27/03/2023	6.00 to 7.30	Revision on Mini project 2 modules	Nil	Good	Nil			
28/03/2023	6.00 to 7.30	PEP 8 technique and benefits of python libraries	Nil	Good	Nil			
29/03/2023	6.00 to 7.30	data types- mutable and immutable features, examples	Nil	Good	Nil			
30/03/2023	6.00 to 7.30	Data structures- djikstar and a* algorithm	Nil	Good	Nil			
31/03/2023	6.00 to 7.30	Assignment on Connecting Servers through web	Nil	Good	Nil			
03/04/2023	6.00 to 7.30	Discussed about interview questions and answers	Nil	Good	Nil			
04/04/2023	6.00 to 7.30	Byte array and frozen set explained with codes	Nil	Good	Nil			
05/04/2023	6.00 to 7.30	Sequential and boolean in python	Nil	Good	Nil			
06/04/2023	6.00 to 7.30	Project review for testing techniques and use cases	Nil	Good	Nil			
08/04/2023	6.00 to 7.30	Polymorphism revision with its types, code	Nil	Good	Nil			
11/04/2023	6.00 to 7.30	Object oriented programming- Encapsulation	Nil	Good	Nil			

12/04/2023	6.00 to 7.30	Django models and abstract	Nil	Good	Nil			
13/04/2023	6.00 to 7.30	Discussed on Django framework	Nil	Good	Nil			
17/04/2023	6.00 to 7.30	Discussion on Documentation for project	Nil	Good	Nil			
18/04/2023	6.00 to 7.30	Project review on Mini projects	Nil	Good	Nil			
19/04/2023	6.00 to 7.30	Technical coding preparation with examples	Nil	Good	Nil			
20/04/2023	6.00 to 7.30	Project documentaion explained	Nil	Good	Nil			
21/04/2023	6.00 to 7.30	Mini project uploading and queries discussed	Nil	Good	Nil			
22/04/2023	6.00 to 7.30	Grading on assignment and explained output	Nil	Good	Nil			
24/04/2023	6.00 to 7.30	Interview Preparation Session	Nil	Good	Nil			
26/04/2023	6.00 to 7.30	Aptitude Question and Answers Preparation	Nil	Good	Nil			
27/04/2023	6.00 to 7.30	Project review and evaluation	Nil	Good	Nil			
28/04/2023	6.00 to 7.30	Prepared students with quantitative aptitude and job links	Nil	Good	Nil			
29/04/2023	6.00 to 7.30	Quantitative aptitude test -1 done	Nil	Good	Nil			
01/05/2023	6.00 to 7.30	Correction on test and evaluation	Nil	15	Nil			
02/05/2023	6.00 to 7.30	Verbal ability preparation with 2 links	Nil	17	Nil			
03/05/2023	6.00 to 7.30	Verbal Ability Test -2 done	Nil	17	Nil			
04/05/2023	6.00 to 7.30	Worked on Code and technical aptitude questions	Nil	14	Nil			
05/05/2023	6.00 to 7.30	Preparation on technical test	Nil	15	Nil			
06/05/2023	6.00 to 7.30	Technical Round Test -3 done	Nil	10	Nil			
08/05/2023	6.00 to 7.30	Explained testing tools and use cases for documentation	Nil	14	Nil			
09/05/2023	6.00 to 7.30	Explained ER diagram and Use case diagram for project	Nil	14	Nil			
10/05/2023	6.00 to 7.30	Final project review with 3 teams	Nil	18	Nil			
11/05/2023	6.00 to 7.30	Final project review with 5 projects	Nil	16	Nil			
12/05/2023	6.00 to 7.30	Preparation for Project viva questions and soft skill classes	Nil	17	Nil			
13/05/2023	6.00 to 7.30	Final Project Viva test and interview Session	Nil	20	Nil			
15/05/2023	6.00 to 7.30	Introduction to soft skill classes	Nil	15	4 students have university exams			
16/05/2023	6.00 to 7.30	Instructions followed in documentation and VIVA	Nil	15	4 students have university exams			
17/05/2023	6.00 to 7.30	Updated Status on assignments, attendance & Test marks to the students	Nil	12	4 students have university exams			
18/05/2023	6.00 to 7.30	Django Framework topics discussed for interview	Nil	15	4 students have university exams			
19/05/2023	6.00 to 7.30	Soft skill class -2 and explained once again on big blue App to students	Nil	16	4 students have university exams			
20/05/2023	6.00 to 7.30	Discussed on Project submission and documentation check - team wise	Nil	14	3 girls didnot join soft skill class due to network issue			

Project Report On

**“MEDICAL INSURANCE COST  
PREDICTION”**

Submitted to **HOPE FOUNDATION**



In partial fulfillment for the award of degree of  
**Bachelor of Computer Application (BCA)**

Submitted by

**NISHALINI.S**

**KA.AIACCXHF.0689**

Guided by  
**Ms. AMALI SUNITHA P**

**St. Aloysius degree College – Bengaluru**  
2022

Project Report On

**“MEDICAL INSURANCE COST  
PREDICTION”**

Submitted to **HOPE FOUNDATION**



This is to Certify that the Student **NISHALINI.S (KA.AIACCXHF.0689)**  
hassatisfactorily completed his project

**“MEDICAL INSURANCE COST  
PREDICTION”  
”**

during the year 2021 – 2022 in the partial fulfilment of Bachelor of  
Computer Application [BCA].

**Name & Signature of project Guide:**

## **INDEX**

SL.NO	CONTENT	
1	ABSTRACT	
2	LANGUAGES	
3	SYSTEM DOMAIN	
4	DATAFLOW DIAGRAMS	
5	CODING AND SCREENSHOTS	
6	CONCLUSION	
7	BIBLIOGRAPHY	



## **ABSTRACT**

Insurance is a policy that helps to cover up all loss or decrease loss in terms of expenses incurred by various risks. A number of variables affect how much insurance costs. These considerations of different factors contribute to the insurance policy cost expression

Machine Learning in the insurance sector can make insurance more effective. In the domains of computational and applied mathematics the machine learning (ML) is a well-known research area. ML is one of the computational intelligence aspects when it comes to exploitation of historical data that may be addressed in a wide range of applications and systems.

There are some limitations in ML so; Predicting medical insurance costs using ML approaches is still a problem in the healthcare industry and thus it requires few more investigation and improvement. Using the machine learning algorithms, this study provides a computational intelligence approach for predicting healthcare insurance costs.

## **1.Languages used for the creation of the project:**

1. Front end: html,css,javascript.
2. Back end: Python

## **2. SYSTEM DOMAIN**

### **Hardware specifications**

**Processor- Intel core i3**

**RAM – 8GB**

**System Type – 64 bit OS**

### **Platform specification:**

- Windows operating system  
html,css,javascript.
- Python 3.7.1

## **3. APPLICATION DOMAIN**

We live on a planet full of threats and uncertainty. Including People, households, durables, properties are exposed to different risks and the risk levels can vary. These risks range from risk of health diseases to death if not get protection, and loss in property or assets. But risks cannot usually be avoided, so the world of finance has developed numerous products to shield individuals and organizations from these risks by using financial capital to shield them Therefore Insurance is one of the policies that either decreases or removes loss costs incurred by various risks. The value of insurance in the lives of individuals.

It is therefore very critical to carry out these tasks with high accuracy. So, the possibility of human mistakes are high so insurance agents also use different tools to

calculate the insurance premium. And thus ML is beneficial here. ML may generalize the effort or method to formulate the policy. These ML models can be learned by themselves. The model is trained on insurance data from the past. The model can then accurately predict insurance policy costs by using the necessary elements to measure the payments as its inputs. This decreases human effort and resources and improves the company's profitability. Thus the accuracy can be improved with ML. Our goal is to predict insurance costs.

## **EXPECTED OUTCOMES**

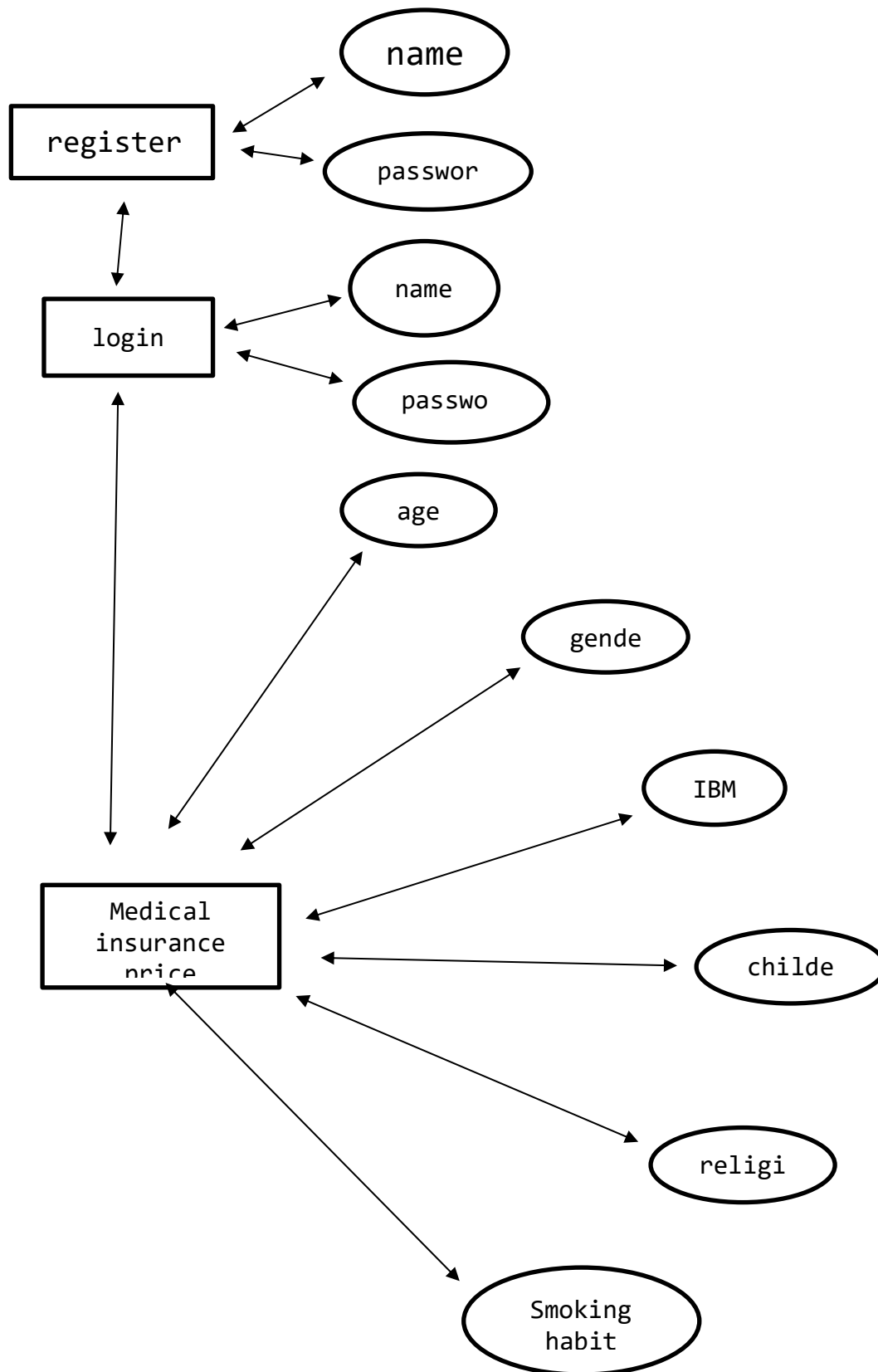
Predicting medical insurance costs is a complex task that relies on a variety of factors, including individual health, demographics, lifestyle, and more. The outcomes of a medical insurance cost prediction model can be used for several purposes. Here are some potential outcomes and use cases:

1. **Premium Estimation:** Predicting insurance costs for individuals or groups, which helps insurance companies determine the appropriate premium rates to charge customers. This outcome can assist in setting competitive pricing while maintaining profitability.
2. **Underwriting Risk Assessment:** Evaluating the risk associated with insuring a particular individual or group, which can help insurance companies decide whether to provide coverage and at what cost.
3. **Policy Customization:** Personalizing insurance policies to meet individual needs and risk profiles. Predictive models can help tailor coverage options and cost based on specific requirements and preferences.
4. **Cost Control:** Assisting individuals and companies in managing their healthcare costs by providing insights into potential expenses and suggesting ways to mitigate them through lifestyle changes, preventive care, or wellness programs.
5. **Fraud Detection:** Identifying potentially fraudulent claims by comparing predicted costs with actual claims data. Unusual discrepancies may indicate fraudulent activities.
6. **Population Health Management:** Identifying high-risk populations and designing targeted interventions and wellness programs to reduce future healthcare costs and improve overall health outcomes.
7. **Claims Processing:** Streamlining the claims process by automatically reviewing and approving or flagging claims for further investigation based on predicted cost estimates.
8. **Resource Allocation:** Helping healthcare providers allocate resources more effectively by predicting patient healthcare costs and needs, thus optimizing staffing, equipment, and budgeting.
9. **Consumer Education:** Educating policyholders about the factors that influence their insurance costs and how they can make informed decisions to manage and reduce their healthcare expenses.
10. **Regulatory Compliance:** Ensuring that insurance companies comply

with regulations and maintain fairness in pricing and risk assessment.

It's important to note that accurate medical insurance cost prediction models should consider a wide range of variables, including age, gender, pre-existing conditions, lifestyle factors, geographic location, and more. Machine learning algorithms, statistical models, and data analysis techniques are commonly used to build predictive models for medical insurance costs. These models can help improve the efficiency, fairness, and transparency of the insurance industry while providing individuals and organizations with the information they need to make informed decisions about their coverage.

## ER model for Medical Insurance Price Prediction



## CODING AND SCREENSHOTS

## Frontend

### Registration page (HTML and CSS):

```
<!DOCTYPE html>

<html>

<head>

  <title>Register</title>

  <style>

    @keyframes fadeIn {

      0% {

        opacity: 0;

      }

      100% {

        opacity: 1;

      }

    }

    body {

      font-family: Arial, sans-serif;

      background: linear-gradient(to bottom, #D0BDF4,

#97E0E7); /* Light violet and sky blue gradient background */

      margin: 0;

      padding: 0;

      display: flex;

      flex-direction: column;

      align-items: center;

      height: 100vh;

    }

  }

</style>

</head>

</html>
```

```

h1 {
    text-align: center;
    color: #333; /* Dark gray text color */
}

.form-container {
    max-width: 300px;
    padding: 50px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2); /* Light gray
box shadow */
    animation: fadeIn 1s;
    background: #fff; /* White background */
    border-radius: 10px;
    margin-top: 20px; /* Add margin to separate the form
container from the header */
}

label {
    display: block;
    margin-top: 10px;
    color: #333; /* Dark gray label text color */
}

input[type="text"],
input[type="password"] {
    width: 100%;
    padding: 10px;
    margin-top: 5px;
    border: 1px solid #ccc; /* Light gray border */
    border-radius: 5px;
    background: #f4f4f4; /* Light gray background */
    color: #333; /* Dark gray text color */
}

```



```

input[type="text"]:focus,
input[type="password"]:focus {
    outline: none;
    border-color: #3E50B4; /* Sky blue border color on focus
*/
}

button[type="submit"] {
    background: linear-gradient(to bottom, #3E50B4,
#5BC1ED); /* Sky blue gradient button background */
    color: #fff; /* White text color */
    padding: 10px 15px;
    border: none;
    cursor: pointer;
    animation: fadeIn 1s;
    border-radius: 5px;
}

button[type="submit"]:hover {
    background: linear-gradient(to bottom, #5BC1ED,
#3E50B4); /* Darker sky blue gradient button background on hover */
}

a {
    display: block;
    text-align: center;
    margin-top: 10px;
    color: #3E50B4; /* Sky blue link text color */
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

```

```

        </style>
</head>
<body>
    <h1>Register</h1>
    <div class="form-container">
        <form method="POST" action="/register">
            <label for="username">Username:</label>
            <input type="text" name="username" id="username"
required>
            <label for="password">Password:</label>
            <input type="password" name="password" id="password"
required>
            <button type="submit">Register</button>
        </form>
        <center><a href="/login">Login</a></center>
    </div>
</body>
</html>

```

### Login Page (HTML and CSS) :

```

<!DOCTYPE html>
<html>
<head>
    <title>Register</title>
    <style>
        @keyframes fadeIn {
            0% {
                opacity: 0;
            }
            100% {
                opacity: 1;
            }
        }
    </style>

```

```

    }

    body {
        font-family: Arial, sans-serif;
        background: linear-gradient(to bottom, #D0BDF4,
#97E0E7); /* Light violet and sky blue gradient background */
        margin: 0;
        padding: 0;
        display: flex;
        flex-direction: column;
        align-items: center;
        height: 100vh;
    }

    h1 {
        text-align: center;
        color: #333; /* Dark gray text color */
    }

    .form-container {
        max-width: 300px;
        padding: 50px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.2); /* Light gray
box shadow */
        animation: fadeIn 1s;
        background: #fff; /* White background */
        border-radius: 10px;
        margin-top: 20px; /* Add margin to separate the form
container from the header */
    }

    label {
        display: block;
        margin-top: 10px;

```

```

        color: #333; /* Dark gray label text color */
    }

    input[type="text"],
    input[type="password"] {
        width: 100%;
        padding: 10px;
        margin-top: 5px;
        border: 1px solid #ccc; /* Light gray border */
        border-radius: 5px;
        background: #f4f4f4; /* Light gray background */
        color: #333; /* Dark gray text color */
    }

    input[type="text"]:focus,
    input[type="password"]:focus {
        outline: none;
        border-color: #3E50B4; /* Sky blue border color on focus
*/
    }

    button[type="submit"] {
        background: linear-gradient(to bottom, #3E50B4,
#5BC1ED); /* Sky blue gradient button background */
        color: #fff; /* White text color */
        padding: 10px 15px;
        border: none;
        cursor: pointer;
        animation: fadeIn 1s;
        border-radius: 5px;
    }

    button[type="submit"]:hover {
        background: linear-gradient(to bottom, #5BC1ED,

```

```

#3E50B4); /* Darker sky blue gradient button background on hover */
    }

    a {
        display: block;
        text-align: center;
        margin-top: 10px;
        color: #3E50B4; /* Sky blue link text color */
        text-decoration: none;
    }

    a:hover {
        text-decoration: underline;
    }
</style>
</head>
<body>
    <h1>Register</h1>
    <div class="form-container">
        <form method="POST" action="/register">
            <label for="username">Username:</label>
            <input type="text" name="username" id="username"
required>
            <label for="password">Password:</label>
            <input type="password" name="password" id="password"
required>
            <button type="submit">Register</button>
        </form>
        <center><a href="/login">Login</a></center>
    </div>
</body>
</html>

```

### **Home page Template (html):**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8"/>
  <meta name="viewport" content="width=device-width, initial-
scale=1, maximum-scale=1.0"/>
  <title>Medical Cost Prediction</title>

  <!-- CSS -->
  <link
href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
  <link href="../../static/css/materialize.css" type="text/css"
rel="stylesheet" media="screen,projection"/>
</head>

<body>
  <nav class="purple darken-4 accent-3" role="navigation">
    <div class="nav-wrapper container"><a id="logo-container"
href="#" class="brand-logo">Logo</a>
    <ul class="right hide-on-med-and-down">
      <li><a href="#">Navbar Link</a></li>
    </ul>

    <ul id="nav-mobile" class="sidenav">
```

```

        <li><a href="#">Navbar Link</a></li>
    </ul>
    <a href="#" data-target="nav-mobile" class="sidenav-
trigger"><i class="material-icons">menu</i></a>
</div>
</nav>

<div class="section no-pad-bot" id="index-banner">
    <div class="container">
        <br><br>
        <h1 class="header center red-text">Medical Cost
Prediction</h1>
        <div class="row center">
            <h5 class="header col s12 light black-text">Predict the cost
for your Medical Insurance!
            <br>
            </h5>
        </div>

<div class="row">
    <form action="/predict" method="post" class="col s12">
        <div class="row">
            <div class="input-field col s4">
                <label for="first_name" style="font-size:
20px"><b>Age</b></label>
                <br>
                <input placeholder="Age" name="age" id="age" type="text"
class="validate">
            </div>

            <div class="input-field col s4">
                <label for="_name" style="font-size:
20px"><b>Gender</b></label>

```

```
        <br>
        <input id="_name" name="Gender" placeholder="0 - Male/1
- Female" type="text" class="validate">
    </div>
```

```
    <div class="input-field col s4">
        <label for="last_name" style="font-size:
20px"><b>BMI</b></label>
        <br>
        <input id="bmi" name="bmi" placeholder="BMI"
type="text" class="validate">
    </div>
```

```
    <div class="input-field col s4">

        <label for="last_name" style="font-size:
20px"><b>Childrens</b></label>
        <br>
        <input id="children" name="children" placeholder="0 for
None" type="text" class="validate">

    </div>
```

```
    <div class="input-field col s4">
        <label for="smoker" style="font-size: 20px"><b>Do you
Smoke?</b></label>
        <br>
        <input id="smoker" name="smoker" placeholder="1 - Yes/0
- No" type="text" class="validate">
    </div>
```

```
    <div class="input-field col s4">
        <label for="region" style="font-size: 20px"><b>Which
Region?</b></label>
```



```

        <br>
        <input id="region" name="region" placeholder="0-NW/1-
NE/2-SE/3-SW" type="text" class="validate">
    </div>
</div>

<div class="row center">

    <button type="submit" class="btn-large waves-effect waves-
light lamber lighten-1">Predict Probability</button>
</div>
</form>
</div>
<div class="dark-text">
<br>
    <h4>{{pred}}</h4><br>
</div>
</div>
</div>

```

```

    <br><br>
</div>
</div>>

```

```

<footer class="page-footer light-blue lighten-1">
    <div class="container">
        <div class="row">
            <div class="col l6 s12">
                <h5 class="white-text">Company Bio</h5>
                <p class="grey-text text-lighten-4">We are a team of
college students working on this project like it's our full time
job. Any amount would help support and continue development on this
project and is greatly appreciated.</p>

```

```

</div>

<div class="col l3 s12">
  <h5 class="white-text">Settings</h5>
  <ul>
    <li><a class="white-text" href="#">Link 1</a></li>
    <li><a class="white-text" href="#">Link 2</a></li>
    <li><a class="white-text" href="#">Link 3</a></li>
    <li><a class="white-text" href="#">Link 4</a></li>
  </ul>
</div>

<div class="col l3 s12">
  <h5 class="white-text">Connect</h5>
  <ul>
    <li><a class="white-text" href="#">LinkedIn</a></li>
    <li><a class="white-text" href="#">GitHub</a></li>
  </ul>
</div>
</div>
</div>
<div class="footer-copyright">
  <div class="container">
    Made by <a class="orange-text text-lighten-3"
href="http://materializecss.com">Materialize</a>
  </div>
</div>
</footer>

<!-- Scripts-->
<script src="https://code.jquery.com/jquery-
2.1.1.min.js"></script>
<script src="../static/js/materialize.js"></script>
<script src="js/init.js"></script>

```

```
</body>
</html>
```

### **Prediction page TEMPLATE ( HTML):**

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"/>
    <style>
/* fallback */
@font-face {
  font-family: 'Material Icons';
  font-style: normal;
  font-weight: 400;
  src: url(https://fonts.gstatic.com/s/materialicons/v140/flUhRq6tzZclQEJ-Vdg-
IuiaDsNc.woff2) format('woff2');
}

.material-icons {
  font-family: 'Material Icons';
  font-weight: normal;
  font-style: normal;
  font-size: 24px;
  line-height: 1;
  letter-spacing: normal;
  text-transform: none;
  display: inline-block;
  white-space: nowrap;
  word-wrap: normal;
  direction: ltr;
  -webkit-font-feature-settings: 'liga';
  -webkit-font-smoothing: antialiased;
}

</style>
</head>

  <body>
    <nav class="blue darken-4 accent-3">
      <div class="nav-wrapper container"><a id="logo-
container" href="#" class="brand-logo">Logo</a>

    </nav>
```

```

<div class="section no-pad-bot" id="index-banner">
  <div class="container">
    <br><br>
    <h3 class="header red-text">Predicted Amount : </h3>
    <h3>{{pred}}</h3>
  </div>
</div>
</body>

</html>

```

## Backend

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

```

```
%matplotlib inline
```

```

data = pd.read_csv('insurance.csv')
data.head()

```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900		0	yes southwest	16884.92400
1	18	male	33.770		1	no southeast	1725.55230
2	28	male	33.000		3	no southeast	4449.46200
3	33	male	22.705		0	no northwest	21984.47061

age	sex	bmi	children	smoker	region	charges
4	32	male	28.880	0	no northwest	3866.85520

In [3]:

```
data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         1338 non-null   int64
1   sex         1338 non-null   object
2   bmi         1338 non-null   float64
3   children    1338 non-null   int64
4   smoker      1338 non-null   object
5   region      1338 non-null   object
6   charges     1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

## There are no missing values as such

In [4]:

```
data['region'].value_counts().sort_values()
```

Out [4]:

```
northeast    324
southwest    325
northwest    325
southeast    364
Name: region, dtype: int64
```

In [5]:

```
data['children'].value_counts().sort_values()
```

Out [5]:

```
5    18
4    25
3   157
2   240
1   324
0   574
Name: children, dtype: int64
```

## Converting Categorical Features to Numerical

In [6]:

```
clean_data = {'sex': {'male': 0, 'female': 1},
              'smoker': {'no': 0, 'yes': 1},
              'region': {'northwest': 0, 'northeast': 1, 'southeast': 2, 'southwest': 3}
            }
```

```
data_copy = data.copy()
data_copy.replace(clean_data, inplace=True)
```

In [7]:

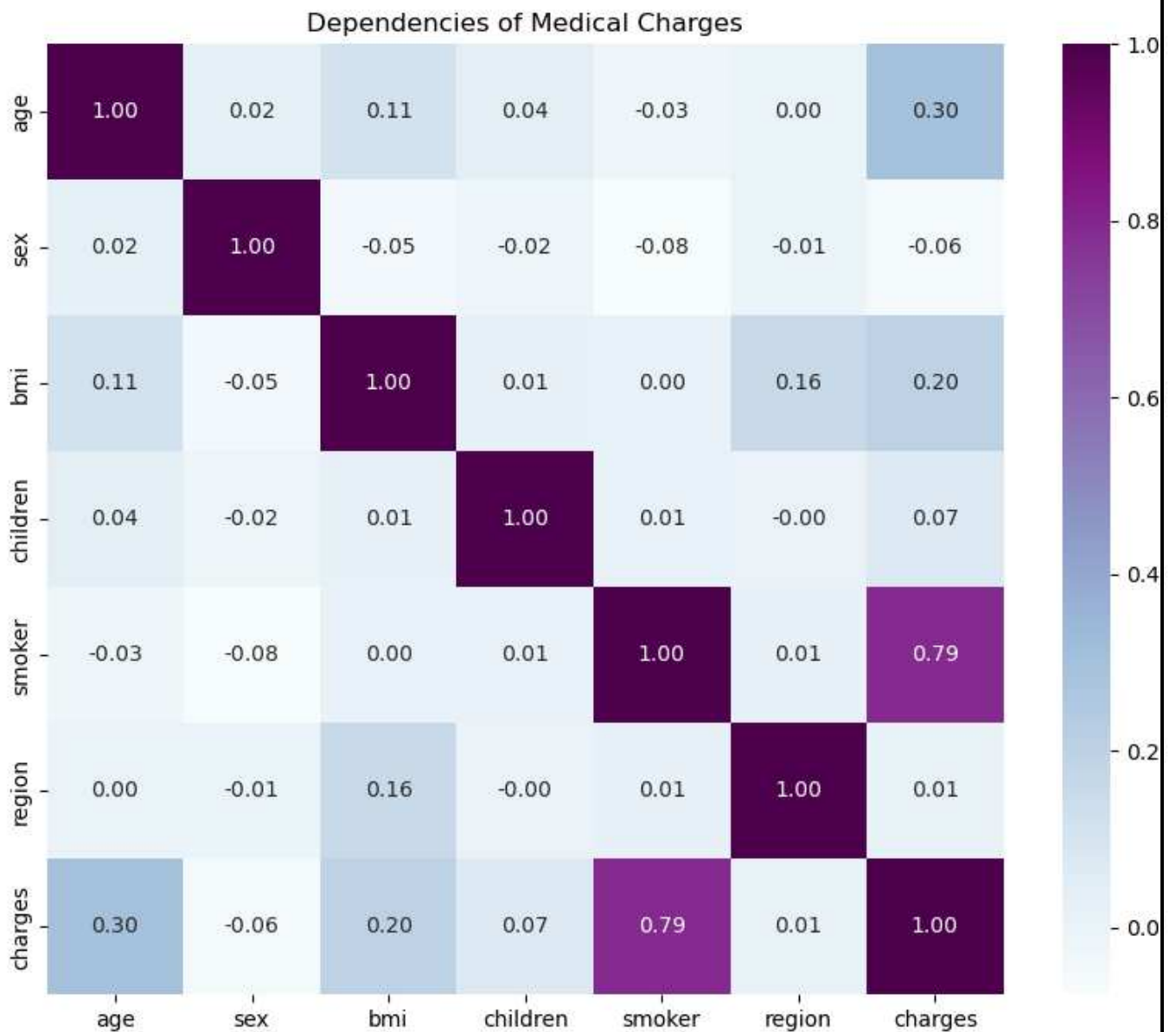
```
data_copy.describe()
```

Out[7]:

	Age	sex	bmi	children	smoker	region	charges
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	0.494768	30.663397	1.094918	0.204783	1.514948	13270.422265
std	14.049960	0.500160	6.098187	1.205493	0.403694	1.105572	12110.011237
min	18.000000	0.000000	15.960000	0.000000	0.000000	0.000000	1121.873900
25%	27.000000	0.000000	26.296250	0.000000	0.000000	1.000000	4740.287150
50%	39.000000	0.000000	30.400000	1.000000	0.000000	2.000000	9382.033000
75%	51.000000	1.000000	34.693750	2.000000	0.000000	2.000000	16639.912515
max	64.000000	1.000000	53.130000	5.000000	1.000000	3.000000	63770.428010

In [8]:

```
corr = data_copy.corr()
fig, ax = plt.subplots(figsize=(10,8))
sns.heatmap(corr,cmap='BuPu',annot=True,fmt=".2f",ax=ax)
plt.title("Dependencies of Medical Charges")
plt.savefig('Cor')
plt.show()
```



## Smoker, BMI and Age are most important factor that determines - Charges

Also we see that Sex, Children and Region do not affect the Charges. We might drop these 3 columns as they have less correlation

In [9]:

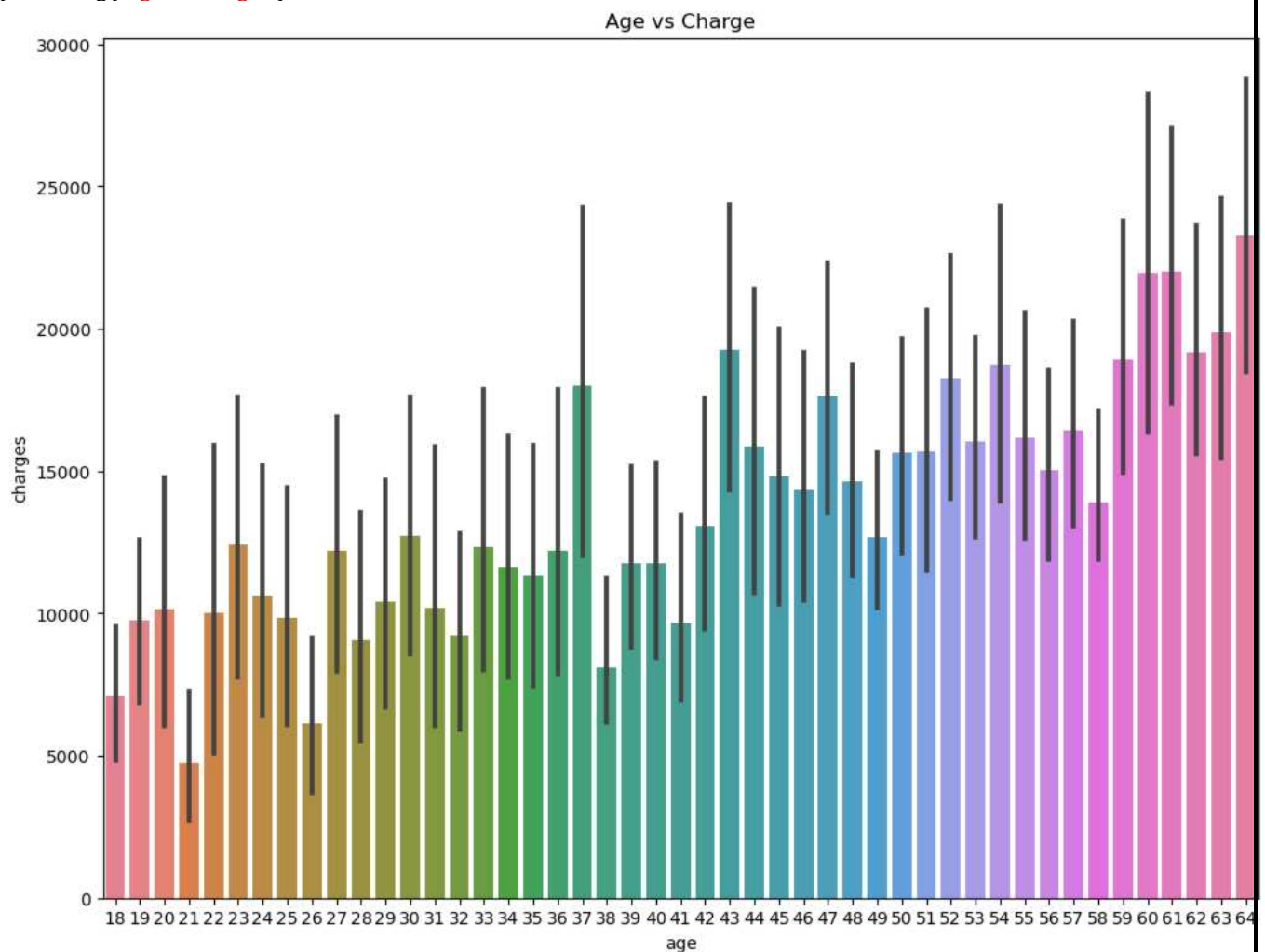
```
print(data['sex'].value_counts().sort_values())
print(data['smoker'].value_counts().sort_values())
print(data['region'].value_counts().sort_values())
female      662
male        676
Name: sex, dtype: int64
yes         274
no         1064
Name: smoker, dtype: int64
northeast   324
```

```
southwest      325
northwest      325
southeast      364
Name: region, dtype: int64
```

**Now we are confirmed that there are no other values in above pre-processed column, We can proceed with EDA**

In [10]:

```
plt.figure(figsize=(12,9))
plt.title('Age vs Charge')
sns.barplot(x='age',y='charges',data=data_copy,palette='husl')
plt.savefig('AgevsCharges')
```



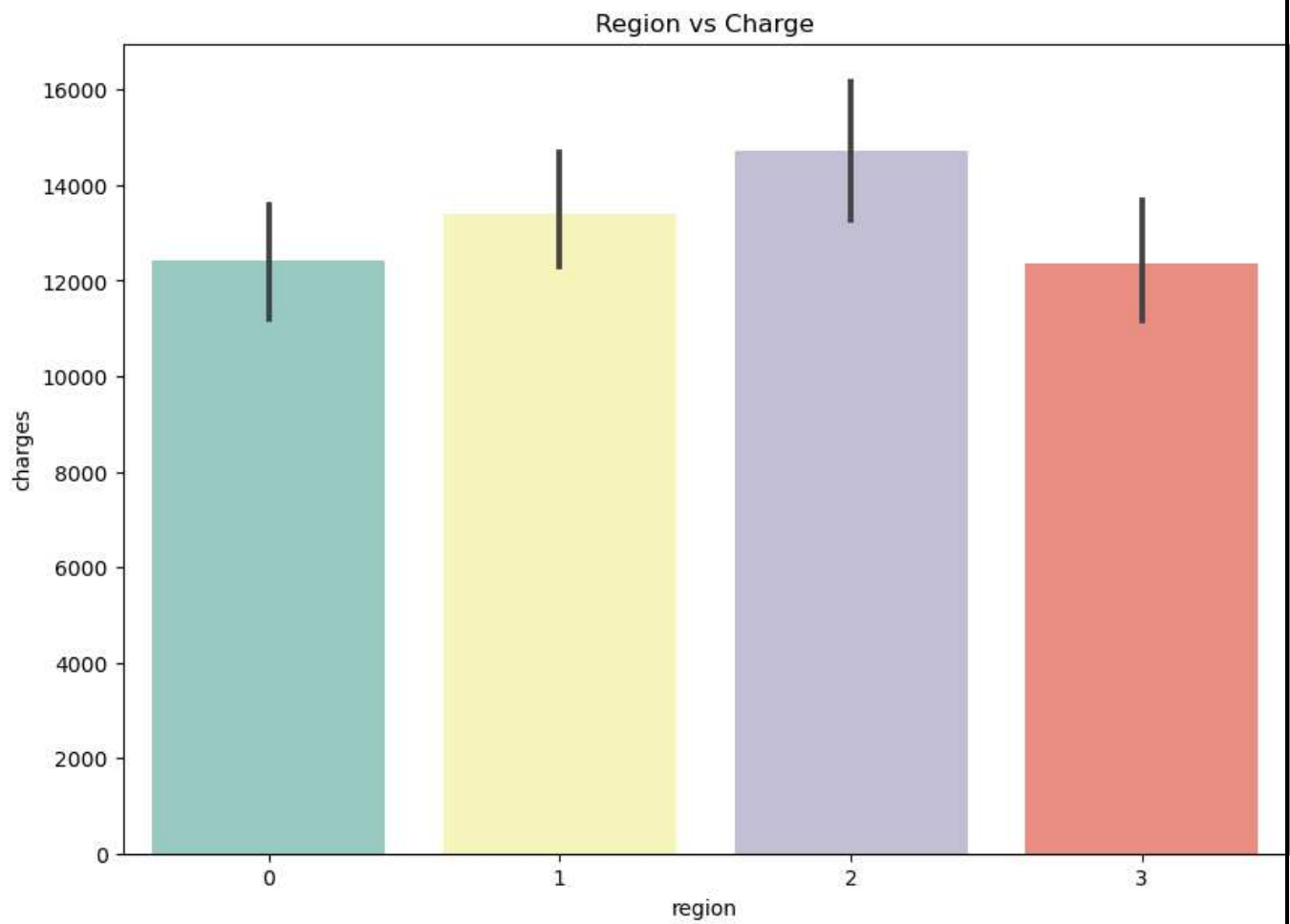
In [11]:

```
plt.figure(figsize=(10,7))
plt.title('Region vs Charge')
sns.barplot(x='region',y='charges',data=data_copy,palette='Set3')
```

Out[11]:

```
<Axes: title={'center': 'Region vs Charge'}, xlabel='region', ylabel='charges'>
```



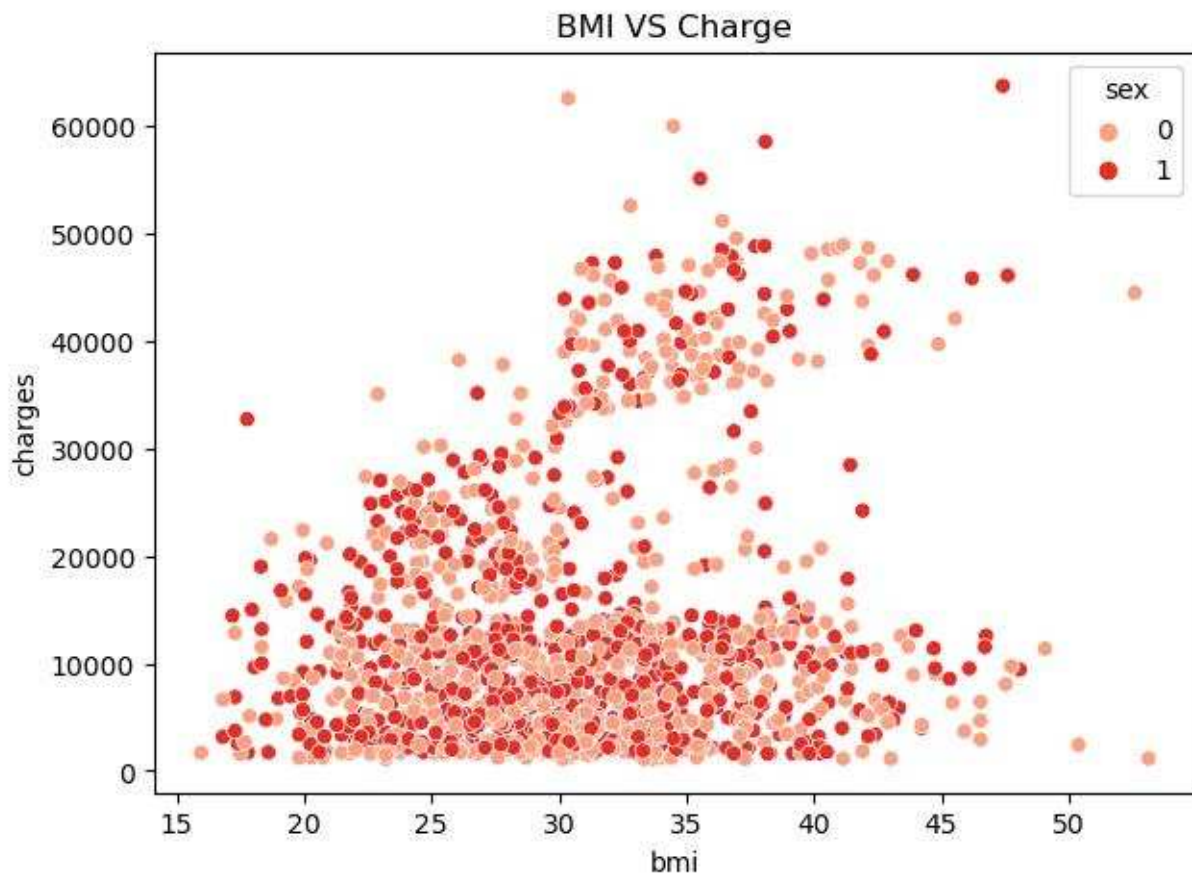


In [12]:

```
plt.figure(figsize=(7,5))  
sns.scatterplot(x='bmi',y='charges',hue='sex',data=data_copy,palette='Reds')  
plt.title('BMI VS Charge')
```

Out[12]:

```
Text(0.5, 1.0, 'BMI VS Charge')
```

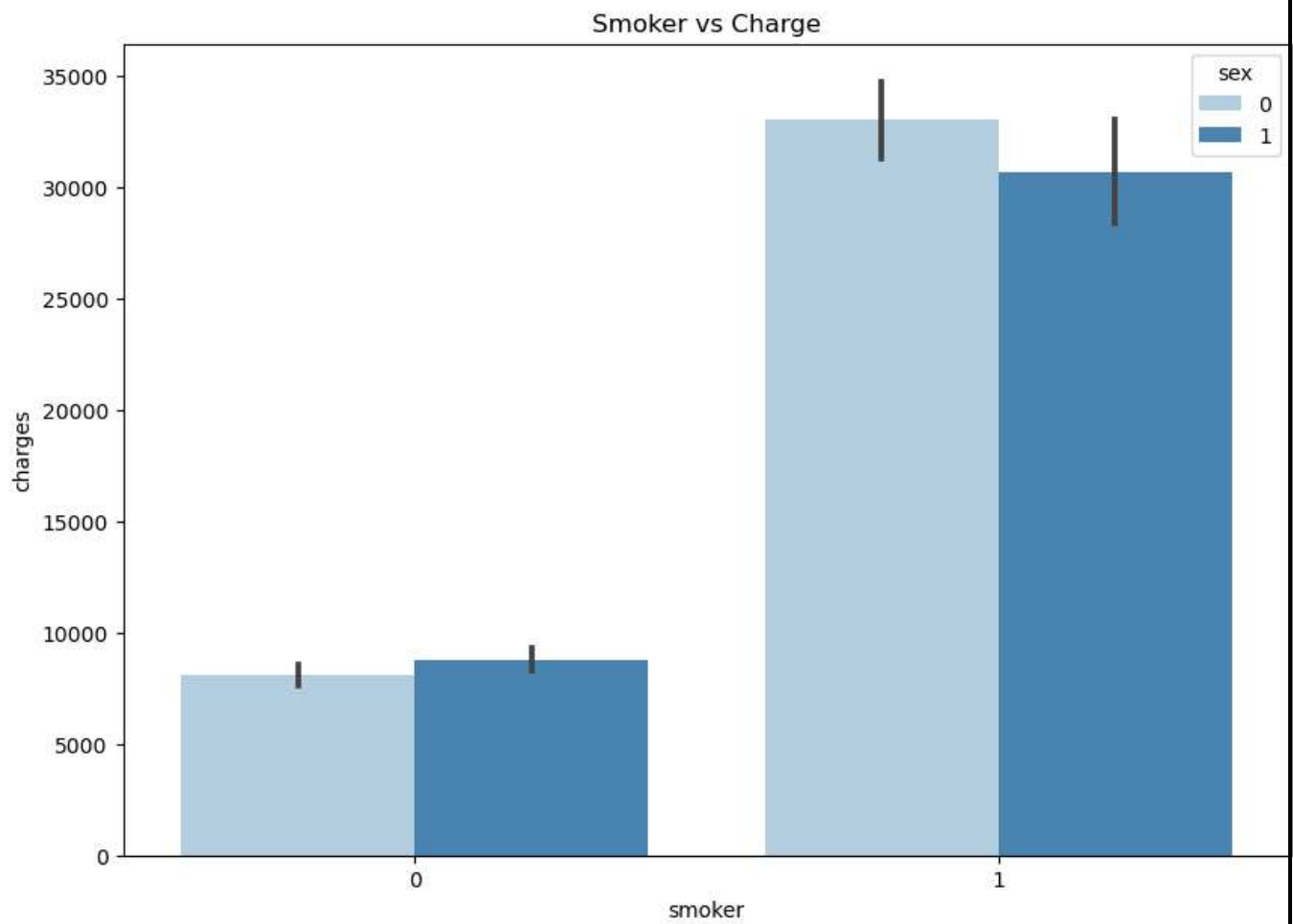


In [13]:

```
plt.figure(figsize=(10,7))  
plt.title('Smoker vs Charge')  
sns.barplot(x='smoker',y='charges',data=data_copy,palette='Blues',hue='sex')
```

Out[13]:

```
<Axes: title={'center': 'Smoker vs Charge'}, xlabel='smoker', ylabel='charges'>
```

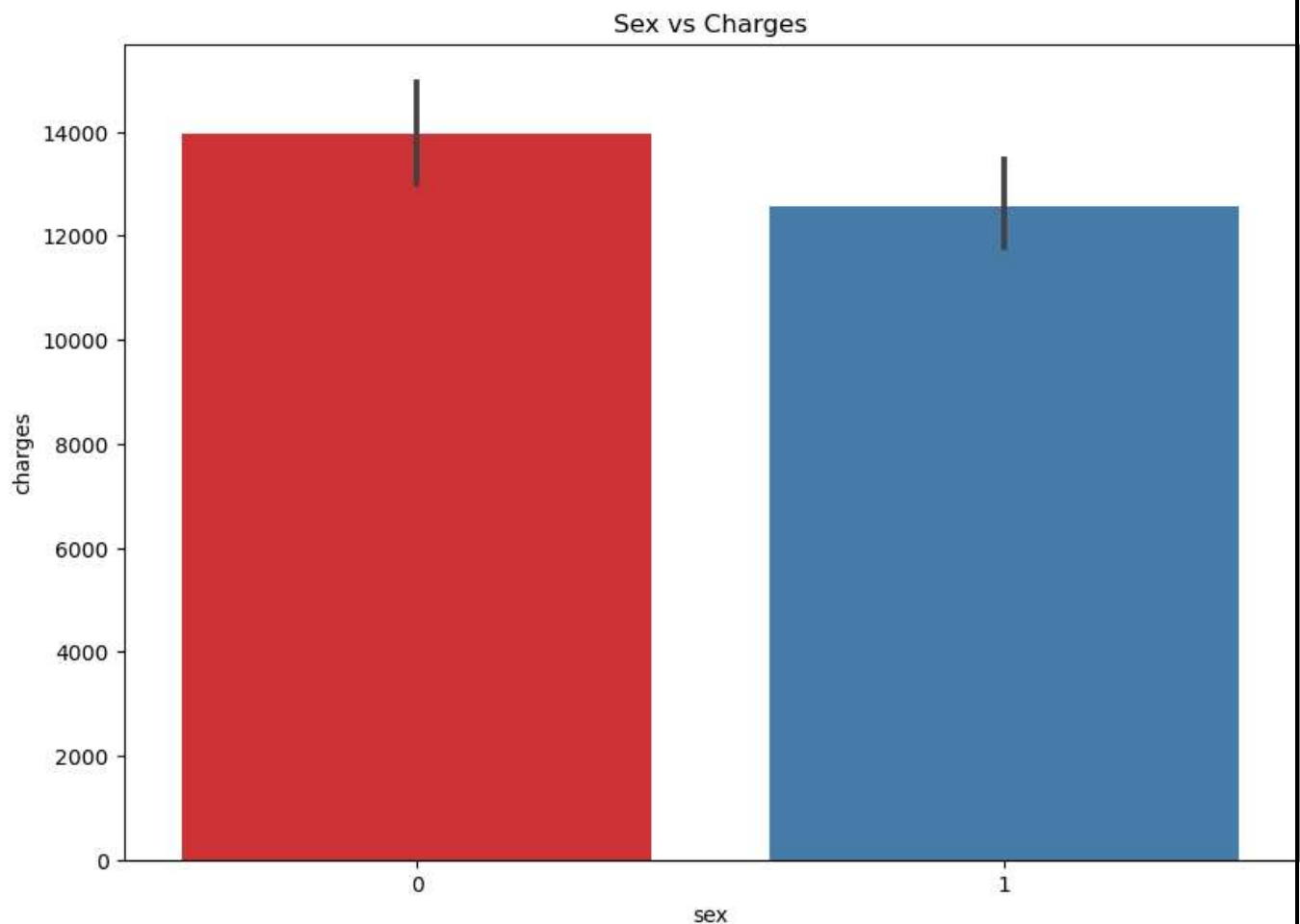


In [14]:

```
plt.figure(figsize=(10,7))  
plt.title('Sex vs Charges')  
sns.barplot(x='sex',y='charges',data=data_copy,palette='Set1')
```

Out[14]:

```
<Axes: title={'center': 'Sex vs Charges'}, xlabel='sex', ylabel='charges'>
```



## Plotting Skew and Kurtosis

In [15]:

```
print('Printing Skewness and Kurtosis for all columns')
print()
for col in list(data_copy.columns):
    print('{0} : Skewness {1:.3f} and Kurtosis {2:.3f}'.format(col,data_copy[col].skew(),data_copy[col].kurt()))
Printing Skewness and Kurtosis for all columns
```

```
age : Skewness 0.056 and Kurtosis -1.245
sex : Skewness 0.021 and Kurtosis -2.003
bmi : Skewness 0.284 and Kurtosis -0.051
children : Skewness 0.938 and Kurtosis 0.202
smoker : Skewness 1.465 and Kurtosis 0.146
region : Skewness -0.038 and Kurtosis -1.329
charges : Skewness 1.516 and Kurtosis 1.606
```

In [16]:

```
plt.figure(figsize=(10,7))
sns.distplot(data_copy['age'])
plt.title('Plot for Age')
plt.xlabel('Age')
```

C:\Users\Vinod Kumar\AppData\Local\Temp\ipykernel\_18164\3356510352.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

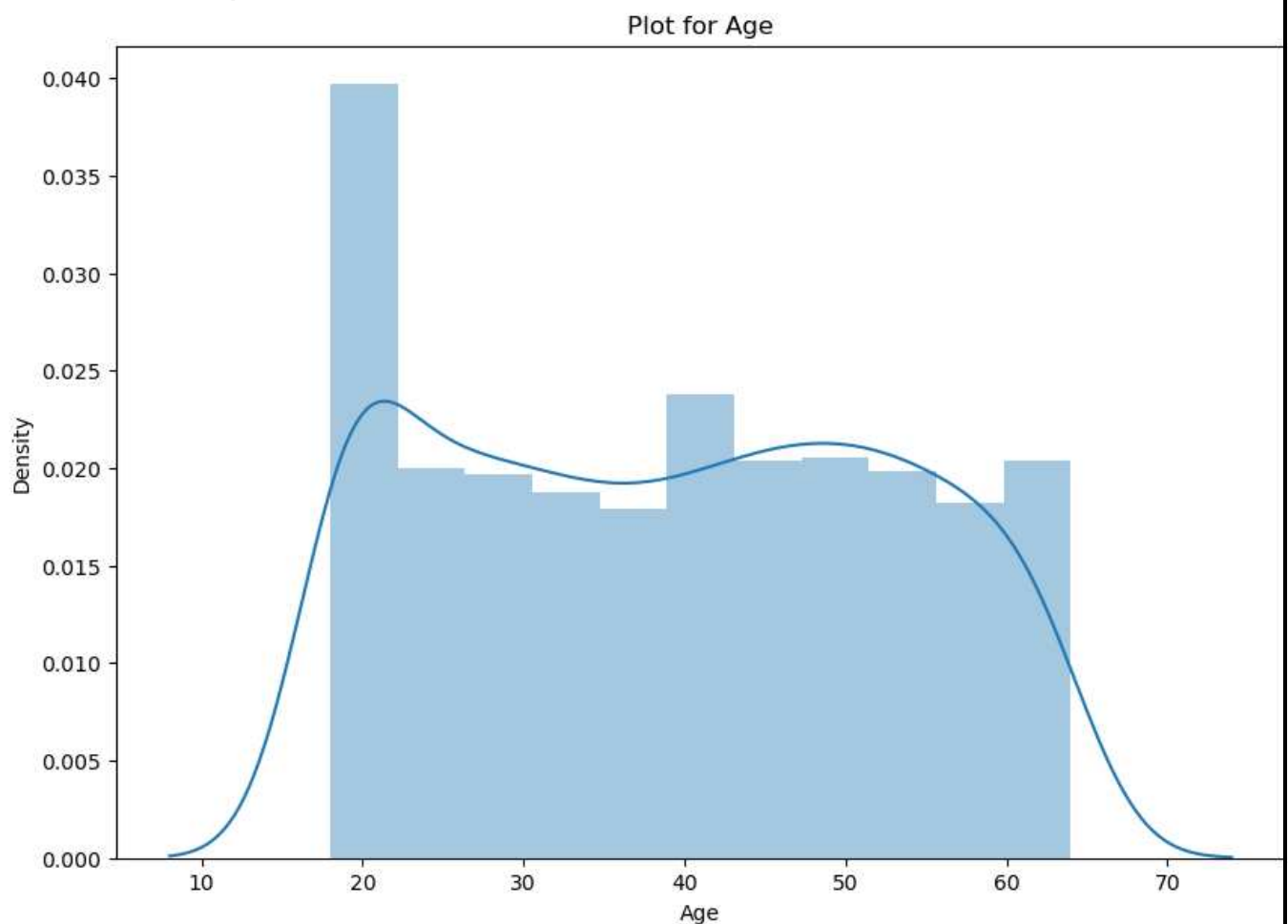
Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data_copy['age'])
```

Out[16]:

```
Text(0.5, 0, 'Age')
```



In [ ]:

In [ ]:

**There might be few outliers in Charges but then we**

**cannot say that the value is an outlier as there might be cases in which Charge for medical was very les actually!**

## **Prepating data - We can scale BMI and Charges Column before proceeding with Prediction**

In [17]:

```
from sklearn.preprocessing import StandardScaler
data_pre = data_copy.copy()

tempBmi = data_pre.bmi
tempBmi = tempBmi.values.reshape(-1,1)
data_pre['bmi'] = StandardScaler().fit_transform(tempBmi)

tempAge = data_pre.age
tempAge = tempAge.values.reshape(-1,1)
data_pre['age'] = StandardScaler().fit_transform(tempAge)

tempCharges = data_pre.charges
tempCharges = tempCharges.values.reshape(-1,1)
data_pre['charges'] = StandardScaler().fit_transform(tempCharges)

data_pre.head()
```

Out[17]:

	age	Sex	bmi	children	smoker	region	charges
0	-1.438764	1	-0.453320	0	1	3	0.298584
1	-1.509965	0	0.509621	1	0	2	-0.953689
2	-0.797954	0	0.383307	3	0	2	-0.728675
3	-0.441948	0	-1.305531	0	0	0	0.719843
4	-0.513149	0	-0.292556	0	0	0	-0.776802

In [18]:

```
X = data_pre.drop('charges',axis=1).values
y = data_pre['charges'].values.reshape(-1,1)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_state=42)

print('Size of X_train : ', X_train.shape)
print('Size of y_train : ', y_train.shape)
print('Size of X_test : ', X_test.shape)
```

```
print('Size of Y_test:', y_test.shape)
Size of X_train : (1070, 6)
Size of y_train : (1070, 1)
Size of X_test : (268, 6)
Size of Y_test : (268, 1)
```

## Importing Libraries

In [19]:

```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
import xgboost as xgb
```

```
from sklearn.metrics import r2_score, mean_squared_error, accuracy_score, confusion_matrix
from sklearn.model_selection import cross_val_score, RandomizedSearchCV, GridSearchCV
```

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
Cell In[19], line 5
      3 from sklearn.tree import DecisionTreeRegressor
      4 from sklearn.svm import SVR
----> 5 import xgboost as xgb
      7 from sklearn.metrics import r2_score, mean_squared_error, accuracy_score
, confusion_matrix
      8 from sklearn.model_selection import cross_val_score, RandomizedSearchCV,
GridSearchCV
```

```
ModuleNotFoundError: No module named 'xgboost'
```

## Linear Regression

In [ ]:

```
%%time
linear_reg = LinearRegression()
linear_reg.fit(X_train, y_train)
```

In [ ]:

```
cv_linear_reg = cross_val_score(estimator = linear_reg, X = X, y = y, cv = 10)
```

```
y_pred_linear_reg_train = linear_reg.predict(X_train)
r2_score_linear_reg_train = r2_score(y_train, y_pred_linear_reg_train)
```

```
y_pred_linear_reg_test = linear_reg.predict(X_test)
r2_score_linear_reg_test = r2_score(y_test, y_pred_linear_reg_test)
```

```
rmse_linear = (np.sqrt(mean_squared_error(y_test, y_pred_linear_reg_test)))
```

```
print('CV Linear Regression : {0:.3f}'.format(cv_linear_reg.mean()))
print('R2_score (train) : {0:.3f}'.format(r2_score_linear_reg_train))
print('R2_score (test) : {0:.3f}'.format(r2_score_linear_reg_test))
```

```
print('RMSE : {0:.3f}'.format(rmse_linear))
```

## **Support Vector Machine** **(Regression)**

In [ ]:

```
X_c = data_copy.drop('charges',axis=1).values  
y_c = data_copy['charges'].values.reshape(-1,1)
```

```
X_train_c, X_test_c, y_train_c, y_test_c = train_test_split(X_c,y_c,test_size=0.2, random_state=42)
```

```
X_train_scaled = StandardScaler().fit_transform(X_train_c)  
y_train_scaled = StandardScaler().fit_transform(y_train_c)  
X_test_scaled = StandardScaler().fit_transform(X_test_c)  
y_test_scaled = StandardScaler().fit_transform(y_test_c)
```

```
svr = SVR()  
#svr.fit(X_train_scaled,y_train_scaled.ravel())
```

In [ ]:

```
parameters = { 'kernel' : ['rbf', 'sigmoid'],  
               'gamma' : [0.001, 0.01, 0.1, 1, 'scale'],  
               'tol' : [0.0001],  
               'C': [0.001, 0.01, 0.1, 1, 10, 100] }  
svr_grid = GridSearchCV(estimator=svr, param_grid=parameters, cv=10, verbose=4, n_jobs=-1)  
svr_grid.fit(X_train_scaled, y_train_scaled.ravel())
```

In [ ]:

```
svr = SVR(C=10, gamma=0.1, tol=0.0001)  
svr.fit(X_train_scaled, y_train_scaled.ravel())  
print(svr_grid.best_estimator_)  
print(svr_grid.best_score_)
```

In [ ]:

```
cv_svr = svr_grid.best_score_  
  
y_pred_svr_train = svr.predict(X_train_scaled)  
r2_score_svr_train = r2_score(y_train_scaled, y_pred_svr_train)  
  
y_pred_svr_test = svr.predict(X_test_scaled)  
r2_score_svr_test = r2_score(y_test_scaled, y_pred_svr_test)  
  
rmse_svr = (np.sqrt(mean_squared_error(y_test_scaled, y_pred_svr_test)))  
  
print('CV : {0:.3f}'.format(cv_svr.mean()))  
print('R2_score (train) : {0:.3f}'.format(r2_score_svr_train))  
print('R2 score (test) : {0:.3f}'.format(r2_score_svr_test))  
print('RMSE : {0:.3f}'.format(rmse_svr))
```

## **Ridge Regressor**

In [ ]:

```
from sklearn.preprocessing import PolynomialFeatures, StandardScaler
```



```

from sklearn.pipeline import Pipeline
from sklearn.linear_model import Ridge

steps = [ ('scalar', StandardScaler()),
          ('poly', PolynomialFeatures(degree=2)),
          ('model', Ridge())]

ridge_pipe = Pipeline(steps)

parameters = { 'model__alpha': [1e-15, 1e-10, 1e-8, 1e-3, 1e-2,1,2,5,10,20,25,35, 43,55,100], 'model__random_state': [42]}
reg_ridge = GridSearchCV(ridge_pipe, parameters, cv=10)
reg_ridge = reg_ridge.fit(X_train, y_train.ravel())

reg_ridge.best_estimator_, reg_ridge.best_score_

ridge = Ridge(alpha=20, random_state=42)
ridge.fit(X_train_scaled, y_train_scaled.ravel())
cv_ridge = reg_ridge.best_score_

y_pred_ridge_train = ridge.predict(X_train_scaled)
r2_score_ridge_train = r2_score(y_train_scaled, y_pred_ridge_train)

y_pred_ridge_test = ridge.predict(X_test_scaled)
r2_score_ridge_test = r2_score(y_test_scaled, y_pred_ridge_test)

rmse_ridge = (np.sqrt(mean_squared_error(y_test_scaled, y_pred_linear_reg_test)))
print('CV : {0:.3f}'.format(cv_ridge.mean()))
print('R2 score (train) : {0:.3f}'.format(r2_score_ridge_train))
print('R2 score (test) : {0:.3f}'.format(r2_score_ridge_test))
print('RMSE : {0:.3f}'.format(rmse_ridge))

models = [('Linear Regression', rmse_linear, r2_score_linear_reg_train, r2_score_linear_reg_test, cv_linear_reg.mean()),
          ('Ridge Regression', rmse_ridge, r2_score_ridge_train, r2_score_ridge_test, cv_ridge.mean()),
          ('Support Vector Regression', rmse_svr, r2_score_svr_train, r2_score_svr_test, cv_svr.mean())
        ]

predict = pd.DataFrame(data = models, columns=['Model', 'RMSE', 'R2_Score(training)', 'R2_Score(test)', 'Cross-Validation'])
predict

plt.figure(figsize=(12,7))
predict.sort_values(by=['Cross-Validation'], ascending=False, inplace=True)

sns.barplot(x='Cross-Validation', y='Model', data = predict, palette='Reds')
plt.xlabel('Cross Validation Score')
plt.ylabel('Model')

```

```
plt.show()
```

## Training Data without Scaling for RandomClassifier

```
data_copy.head()
```

```
X_ = data_copy.drop('charges',axis=1).values  
y_ = data_copy['charges'].values.reshape(-1,1)
```

```
from sklearn.model_selection import train_test_split  
X_train_, X_test_, y_train_, y_test_ = train_test_split(X_,y_,test_size=0.2, random_state=42)
```

```
print('Size of X_train_ : ', X_train_.shape)  
print('Size of y_train_ : ', y_train_.shape)  
print('Size of X_test_ : ', X_test_.shape)  
print('Size of Y_test_ : ', y_test_.shape)
```

```
import pickle
```

```
Pkl_Filename = "svr_tuned.pkl"
```

```
with open(Pkl_Filename, 'wb') as file:  
    pickle.dump(svr, file)
```

```
# Load the Model back from file  
with open(Pkl_Filename, 'rb') as file:  
    rf_tuned_loaded = pickle.load(file)
```

```
rf_tuned_loaded
```

```
pred=rf_tuned_loaded.predict(np.array([20,1,28,0,1,3]).reshape(1,6))[0]
```

```
print('{0:.3f}'.format(pred))
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```

from flask import Flask, request, url_for, redirect,
render_template, session
import pickle
import numpy as np
import csv

app = Flask(__name__, template_folder='./templates',
static_folder='./static')
app.secret_key = "123456" # Set a secret key for session management

Pkl_Filename = "svr_tuned.pkl"
with open(Pkl_Filename, 'rb') as file:
    model = pickle.load(file)

@app.route('/')
def home():
    return render_template('login.html')

# Helper function to check if a user exists in the CSV file
def is_user_registered(username, password):
    with open('users.csv', mode='r') as users_file:
        reader = csv.reader(users_file)
        for row in reader:
            if row[0] == username and row[1] == password:
                return True
    return False

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']

```

```

        if is_user_registered(username, password):
            session['username'] = username # Store the username in
the session
            return redirect('/home')
        else:
            return "Login failed. Please try again or register."

return render_template('login.html')

```

```

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        with open('users.csv', mode='a', newline='\n') as
users_file:
            writer = csv.writer(users_file)
            writer.writerow([username, password])
            return "Registration successful. You can now login."

return render_template('register.html')

```

```

@app.route('/home')
def hello_world():
    return render_template('home.html')

```

```

@app.route('/predict', methods=['POST', 'GET'])
def predict():
    features = [int(x) for x in request.form.values()]

    print(features)
    final = np.array(features).reshape((1,6))
    print(final)

```

```
pred = model.predict(final)[0]
print(pred)

if pred < 0:
    return render_template('op.html', pred='Error calculating
Amount!')
else:
    return render_template('op.html', pred='Expected amount is
{0:.3f}'.format(pred))

if __name__ == '__main__':
    app.run(debug=True)
```

## Register

Username:

Password:

Register

[Login](#)

## Medical Insurance

Username:

Password:

Login

[Register](#)

# Medical Cost Prediction

Predict the cost for your Medical Insurance!

Age	Gender	BMI
<input type="text" value="21"/>	<input type="text" value="1"/>	<input type="text" value="98"/>
Childrens	Do you Smoke?	Which Region?
<input type="text" value="2"/>	<input type="text" value="0"/>	<input type="text" value="2"/>

## Predicted Amount :

Expected amount is 0.443

# CONCLUSION

We've looked at the fundamentals of the linear regression model, how to use it to forecast charges, and how to compare anticipated and real outcomes. I hope you found this post helpful and that you now have a basic understanding of how a linear regression model works. For estimating medical expenditures, we suggested a machine learning approach.. We applied regression techniques Linear Regression and observed that age, BMI are features that decide the dependent variable. Out of all experiments, this model gave a better result.

## Advantages

- **Improve Accuracy:** analyze large volumes of data, including historical insurance claims and market trends, to make more accurate predictions. This accuracy can help insurance providers offer competitive and precise pricing.
- **Cost Reduction:** By accurately predicting policies. This can lead to cost savings and improved profitability.
- **Customization:** personalize insurance pricing based on individual risk factors, allowing insurer to offer tailored customer satisfaction.
- **Faster Decisions:** quickly process and analyze data, which can lead to faster decision making.
- **Fraud Detection:** also help in detecting fraudulent insurance claims.



## Disadvantages

- **Data Privacy concerns:** medical insurance pricing relies on a significant amount of personal and sensitive data. Protecting this data and ensuring compliance with privacy regulations can be challenging.
- **Bias and Fairness Issues:** inherit biases the data they are trained on, leading to discriminatory pricing. This results in some individuals or groups paying unfairly high premiums or being denied coverage.
- **Transparency:** complex and difficult to interpret.
- **Regulatory Challenges:** pricing may raise regulatory concerns and require oversight to ensure that insurance companies are not engaging in unfair or discriminatory practices.
- **Unforeseen risk Factor:** account for all relevant risk factors or overlook emerging risk factors.

# BIBLIOGRAPHY

<https://www.geeksforgeeks.org>

<https://youtu.be/ntBa7YKc9XM?si=sVFf08JzoXJLicS7>

[https://youtu.be/iS\\_iI7btRXk?si=ZPejnnpIcuPYIoDD](https://youtu.be/iS_iI7btRXk?si=ZPejnnpIcuPYIoDD)

Project Report On

**“VIRTUAL ASSISTANT”**

Submitted to **HOPE FOUNDATION**



In partial fulfillment for the award of degree of  
**Bachelor of Computer Application (BCA)**

Submitted by

**A.Valluvan**

**SL.NO 55**

Guided by  
**Ms. AMALI SUNITHA P**

**St. Aloysius degree College - Bengaluru**  
**2022**

Project Report On

**“VIRTUAL ASSISTANT”**

Submitted to **HOPE FOUNDATION**



This is to Certify that the Student **A. VALLUVAN (SL.NO 55)** has satisfactorily completed his project

**“VIRTUAL ASSISTANT”**

during the year 2021 – 2022 in the partial fulfilment of Bachelor of Computer Application [BCA].

**Name & Signature of project Guide:**

## **INDEX**

SL.NO	CONTENT	
1	ABSTRACT	
2	LANGUAGES	
3	SYSTEM DOMAIN	
4	DATAFLOW DIAGRAMS	
5	CODING AND SCREENSHOTS	
6	CONCLUSION	
7	BIBLIOGRAPHY	

## **ABSTRACT**

“Virtual assistant” is an Artificial Intelligence project to facilitate text into speech or converting audio into text.

This software is not that secure but it mainly focuses on AI prediction which is very much needed for voice recognition while handling the users speech on the desktop.

Voice assistants are software agents that can interpret human speech and respond via synthesized voices. Apple's Siri, Amazon's Alexa, Microsoft's Cortana, and Google's Assistant are the most popular voice assistants and are embedded in smartphones or dedicated home speakers

Personal Assistant with Voice Recognition Intelligence, which takes the user input in form of voice or text and process it and returns the output in various forms like action to be performed or the search result is dictated to the end user. In addition, this proposed system can change the way of interactions between end user and the mobile devices. The system is being designed in such a way that all the services provided by the mobile devices are accessible by the end user on the user's voice commands..

## **1.Languages used for the creation of the project:**

1. Front end: Visual Studio Code
2. Back end: Python

## **2. SYSTEM DOMAIN**

### **Hardware specifications**

- Processor-2.30 GHz
- RAM – 4GB
- System Type – 64 bit OS

### **Platform specification:**

- Windows operating system
- Visual Studio code
- Python 3.8.1

## **3. APPLICATION DOMAIN**

- This project is aimed at AI Prediction & smooth functioning of voice assistant audio files into text.
- Maintaining a user-friendly s/w is deemed of highest priority in order to hold the various command comes built-in python.
- Used for conversion of text to speech recognize the voice and gives the output according to users need.

## **EXPECTED OUTCOMES**

**S**

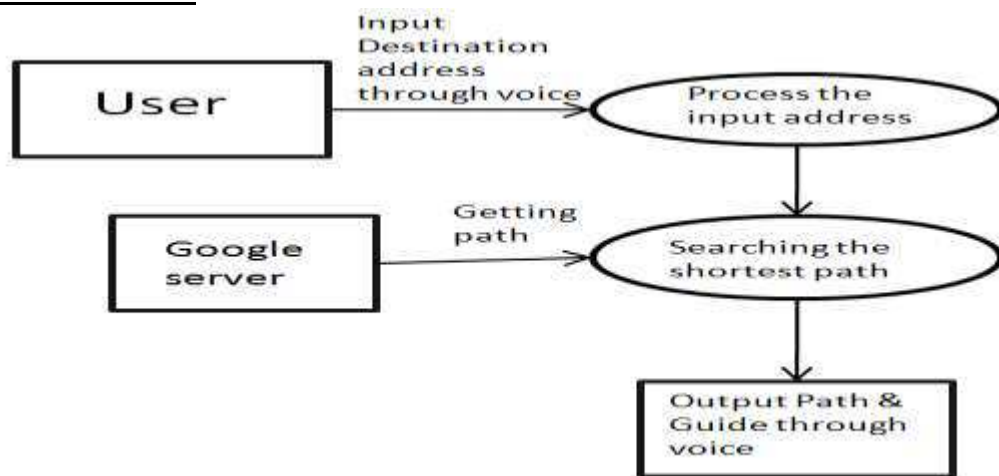
The outcomes expected are as follows:

- Smooth and effective functioning of variety of actions more robust.
- AI prediction for any kind of information provided as input.
- Easy to scrap information from webpages.
- Voice assistants are technology based in both consumer and business environments.

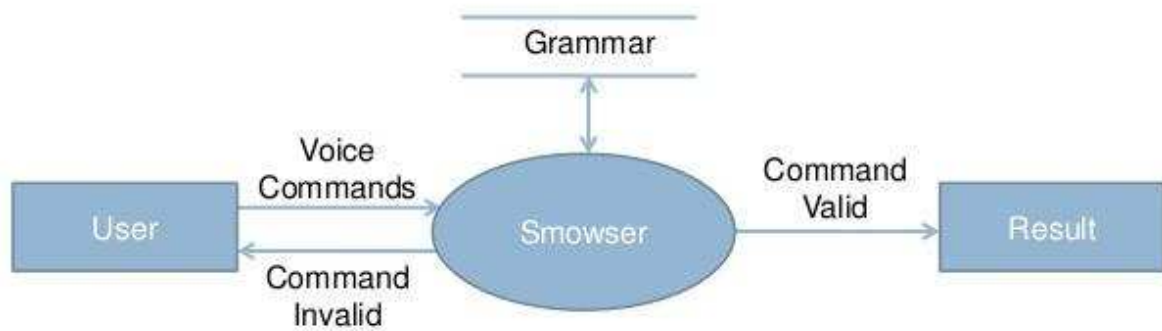


# DATAFLOW DIAGRAMS

## Representation:



## Level 0:



# CODING AND SCREENSHOTS

## VALLUVAN.PY

```
import calendar
import os
import sched
import socket
import time
import turtle
from urllib.request import urlopen
import speech_recognition as sr
import datetime
import wikipedia
import pyttsx3
import webbrowser
from turtle import *
from tkinter import * #digital timeokok
from tkinter.ttk import *
from covid import * # for covid updates
import pyjokes
import phonenumbers # phone number details
from phonenumbers import carrier
from phonenumbers import geocoder
from bs4 import BeautifulSoup as soup #trending news
from urllib.request import urlopen
import socket #Ip address
import speedtest #internet speed

#Text To Speech
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
#print(voices)
engine.setProperty('voice',voices[0].id)
now_time=datetime.datetime.now().strftime("%H:%M:%S") #stores the exact time
in h:m:s(hrs, min, sec) format.
date_now=datetime.date.today() #stores the exact date in y:d:m(yr, date,
month) format.
def speak(audio): #here audio is var which contain text
```

```

engine.say(audio)
engine.runAndWait()

def wish():
    hour = int(datetime.datetime.now().hour)
    if hour >= 0 and hour<12:
        print("Good morning, I am your virtual assistant VALLU, how can i help you?")
        speak("Good morning, I am your virtual assistant VALLU how can i help you")
    elif hour>=12 and hour<18:
        print("Good afternoon, I am your virtual assistant VALLU, how can i help you?")
        speak("Good afternoon, I am your virtual assistant VALLU, how can i help you")
    else:
        print("Good evening, I am your virtual assistant VALLU, how can i help you?")
        speak("Good evening, I am your virtual assistant VALLU, how can i help you")

#now convert audio to text
#
def takecommand():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        audio = r.listen(source)
    try:
        print("Recognising...")
        text = r.recognize_google(audio,language='en-in')
        print(text)
    except Exception:
        #For Error handling
        speak("error , Say that again please")
        print("Network connection error")
        return "none"
    return text

#for main function
if __name__ == "__main__":
    wish()
    while True:
        query = takecommand().lower()
        if "wikipedia" in query:
            speak("searching details....Wait")
            query.replace("wikipedia","")
            results = wikipedia.summary(query,sentences=2)
            print(results)

```

```
        speak(results)
        takecommand()

    elif 'open youtube' in query or "open video online" in query:
        webbrowser.open("www.youtube.com")
        speak("okay, opening youtube")
        takecommand()

    elif 'open facebook' in query:
        webbrowser.open("https://www.facebook.com")
        speak("okay, opening facebook")
        takecommand()

    elif 'open instagram' in query:
        webbrowser.open("https://www.instagram.com")
        speak("okay, opening instagram")
        takecommand()

    elif 'open google' in query:
        webbrowser.open("https://www.google.com")
        speak("okay, opening google")
        takecommand()

    elif 'open yahoo' in query:
        webbrowser.open("https://www.yahoo.com")
        speak("okay, opening yahoo")
        takecommand()

    elif 'open gmail' in query:
        webbrowser.open("https://mail.google.com")
        speak("okay, opening google mail")
        takecommand()

    elif 'open snapdeal' in query:
        webbrowser.open("https://www.snapdeal.com")
        speak("opening snapdeal")
        takecommand()

    elif 'open amazon' in query or 'shop online' in query:
        webbrowser.open("https://www.amazon.com")
        speak("opening amazon")
        takecommand()

    elif 'open flipkart' in query:
        webbrowser.open("https://www.flipkart.com")
        speak("opening flipkart")
        takecommand()
```

```

elif 'open ebay' in query:
    webbrowser.open("https://www.ebay.com")
    speak("opening ebay")
    takecommand()

elif 'open stack overflow' in query:
    print("okay, opening stackoverflow")
    speak("okay, opening stackoverflow")
    webbrowser.open('www.stackoverflow.com')
    takecommand()

elif 'open reddit' in query:
    print("okay, opening reddit")
    speak("okay, opening reddit")
    webbrowser.open('www.reddit.com')
    takecommand()

elif "open instructables" in query:
    print("Okay, opening instructables")
    speak("Okay, opening instructables")
    webbrowser.open("https://www.google.com/search?xsrf=ALeKk010KURlO
wfKChWyWOLdJBuPvIjmaQ%3A1600095782674&ei=JoZfX6PlKKXDpgfm5YqwCg&q=instructable
s+projects&oq=ins&gs_lcp=CgZwc3ktYWIQAxgBMgQIIxAnMgQIIxAnMgQIIxAnMgUIABCRAjILC
AAQsQMgWEQkQIYcAgAELEDEIMBMgQIABBDMgoIABcXAxCDARBDMgQIABBDMgQIABBD0gcIIXCwAhA
nOgcIIXDqAhAnOgUIABcXA1D5H1iZNWDqSwgBcAB4BIABkQKIAZwMkgEFMC44LjGYAQCgAQGqAQdnd
3Mtd2l6sAEKwAEB&scclient=psy-ab")
    takecommand()

    #url="amazon.com"
    #chrome_path='C:/Program Files
(x86)/Google/Chrome/Application/chrome.exe%s'
    #webbrowser.get(chrome_path).open(url)

    #open_new_tab("https://mail.google.com/mail/u/0/?ogbl#inbox")

elif "open my inbox" in query:
    print("Okay, opening your inbox")
    speak("Okay, opening your inbox")
    webbrowser.open("https://mail.google.com/mail/u/0/?ogbl#inbox")
    takecommand()

elif "starred emails" in query or "open my starred emails" in query or
"open starred emails" in query or "open starred messages" in query:
    print("Okay, opening your starred emails")
    speak("Okay, opening your starred emails")
    webbrowser.open("https://mail.google.com/mail/u/0/#starred")

```

```

        takecommand()

        elif "sent emails" in query or "emails I have sent" in query or "show
the emails the I've sent" in query or "show the emails that I have sent" in
query or "open sent emails" in query:
            print("Okay, opening emails that you have sent")
            speak("Okay, opening emails that you have sent")
            webbrowser.open("https://mail.google.com/mail/u/0/?tab=rm&ogbl#sen
t")
            takecommand()

        elif "open chats in my email" in query or "open chats in email" in
query:
            print("Okay, opening the chats in your email")
            speak("Okay, opening the chats in your email")
            webbrowser.open("https://mail.google.com/mail/u/0/?tab=rm&ogbl#cha
ts")
            takecommand()

        elif "open all the mails" in query:
            print("Okay, opening all the mails")
            speak("Okay, opening all the mails")
            webbrowser.open("https://mail.google.com/mail/u/0/?tab=rm&ogbl#all
")
            takecommand()

        elif "open spam emails" in query or "open the emails that are in the
spam section" in query:
            print("Okay, opening spam emails")
            speak("Okay, opening spam emails")
            webbrowser.open("https://mail.google.com/mail/u/0/?tab=rm&ogbl#spa
m")
            takecommand()

        elif "time" in query or "what's the time" in query or "what is the
time" in query:
            print(f"The time is {now_time}")# using an 'f' string says the
time and then tells the time, and the same is with the #date
            speak(f"The time is {now_time}")
            takecommand()

        elif "date" in query or "what is the date today" in query or "what is
the date" in query or "what is today's date" in query or "what's the date" in
query:
            print(f"Today is {date_now}")
            speak(f"Today is {date_now}")

```

```

        takecommand()

    elif "favourite robot" in query:
        print("My favourite robot is R2D2 from star wars, transformers is
also good but he ain't got that mojo")
        speak("My favourite robot is R2D2 from star wars, transformers is
also good but he ain't got that mojo")
        takecommand()

    elif 'play music' in query.lower():
        speak("Alright palying music for you")
        songs_dir = "C:\\Users\\ANTHONY\\Desktop\\Favorite"
        songs = os.listdir(songs_dir)
        print(songs)
        os.startfile(os.path.join(songs_dir, songs[0]))
        takecommand()

    elif 'IP address' in query.lower():
        hostname = socket.gethostname()
        IPAddr = socket.gethostbyname(hostname)
        print("your Computer name is" + hostname)
        speak("your Computer name is" + hostname)
        print("your IP address is" + IPAddr)
        speak("your IP address is" + IPAddr)
        takecommand()

    elif 'draw star' in query.lower():
        speak("okay, drawing star")
        pen = turtle.Turtle()
        n = 20
        pen.color("red")
        for i in range(n):
            pen.forward(i * 10)
            pen.right(144)
        turtle.done()
        takecommand()

    elif 'phone number' in query.lower():
        speak("okay, enter your phone number:")
        Phonenumber=input("Enter your phone number:")
        number=phonenumbers.parse(Phonenumber)
        print(geocoder.description_for_number(number, 'en'))
        speak(geocoder.description_for_number(number, 'en'))
        print(carrier.name_for_number(number, 'en'))
        speak(carrier.name_for_number(number, 'en'))
        takecommand()

```

```

elif 'joke' in query.lower():
    joke = pyjokes.get_joke()
    print(joke)
    speak(joke)
    takecommand()

elif 'take screenshot' in query.lower():
    print("Okay, taking screenshot")
    speak("okay, taking screenshot")
    time.sleep(3)
    e = sched.grab()
    e.save('New-Screenshot.png')
    e.show()
    takecommand()

elif 'calculate my age' in query.lower():
    print("Are you ready to know your age:")
    speak("Are you ready to know your age")

    def age_calculator(age):
        age_s = age * 365 * 24 * 60 * 60
        age_m = age * 365 * 24 * 60
        age_d = age * 365
        return F"You are {age_s} seconds only old\nYou are {age_m}
minutes old \nYou are {age_d} days old"

    speak("enter your age")
    Age = int(input("Enter your age in year:"))

    print(age_calculator(Age))
    speak(age_calculator(Age))
    takecommand()

elif 'trending news' in query.lower():
    speak("okay, here are the trending news")
    news_url="https://news.google.com/news/rss"
    Client=urlopen(news_url)
    xml_page=Client.read()
    Client.close()
    soup_page=soup(xml_page, "xml")
    news_list=soup_page.findAll("item")
    for news in news_list:
        print(news.title.text)
        speak(news.title.text)
        print(news.pubDate.text)

```



```

        print("-"*60)
        takecommand()

    elif 'calendar' in query.lower():
        speak("okay, enter the month and year")
        year = int(input("Enter the year:"))
        month = int(input("Enter the month:"))
        print(calendar.month(year,month))
        speak("Here is the calendar for you")
        takecommand()

    elif 'internet speed' in query.lower():
        speak("okay,showing the speed of internet")
        speed=speedtest.Speedtest()
        download=speed.download()
        upload=speed.upload()
        print(f"Download Speed : {download}")
        speak(f"Download Speed : {download}")
        print(f"Upload Speed: {upload}")
        speak(f"Upload Speed: {upload}")
        takecommand()

    elif 'thank you' in query.lower():
        speak("ALways pleasure to help you")
        takecommand()

    elif 'sorry' in query.lower():
        speak('Its ok')
        takecommand()

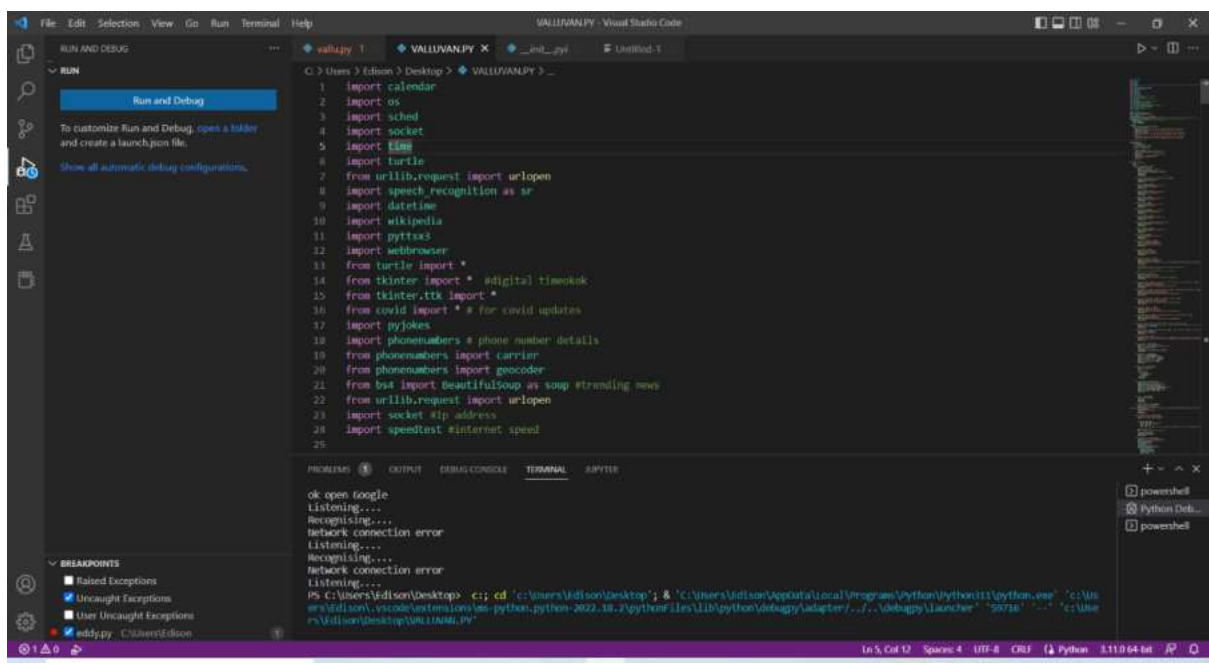
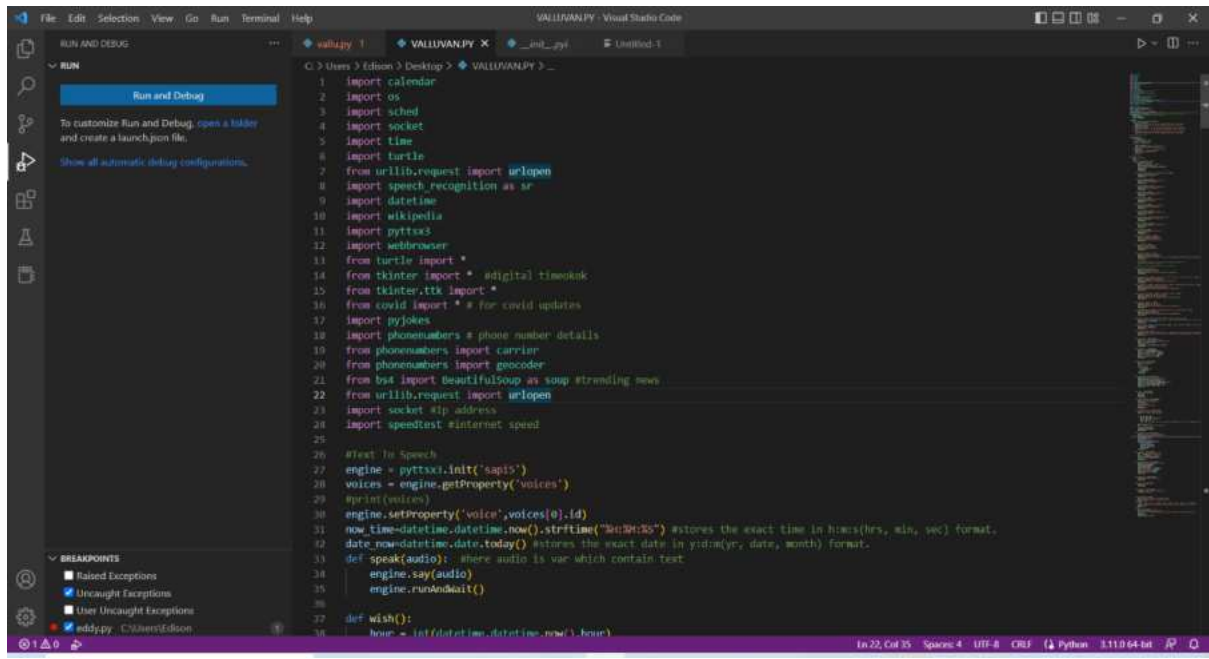
    elif 'shutdown' in query.lower():
        speak("Okay, your system will shutdown now")
        os.system('shutdown /s')

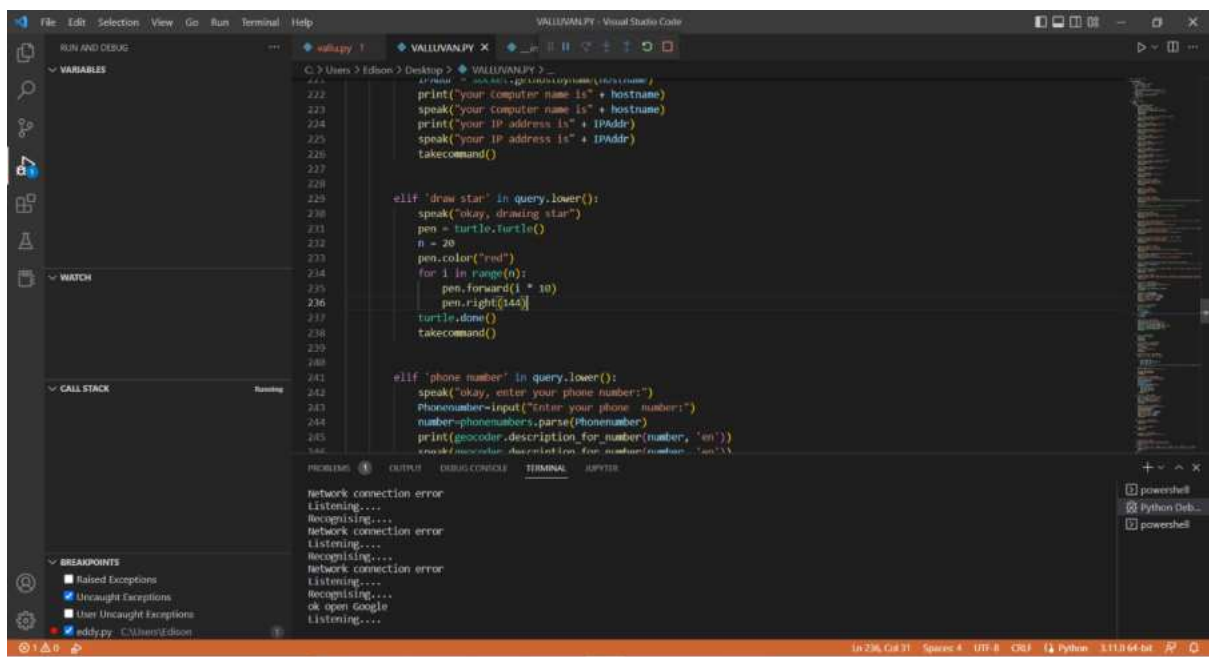
    elif "you feeling" in query:
        print("feeling Very sweet after meeting with you")
        speak("feeling Very sweet after meeting with you")
    elif query == 'none':
        continue
    elif 'exit' in query or 'abort' in query or 'stop' in query or 'bye'
in query or 'quit' in query :
        ex_exit = 'I am feeling very sweet after meeting with you but you
are going! i am very sad'

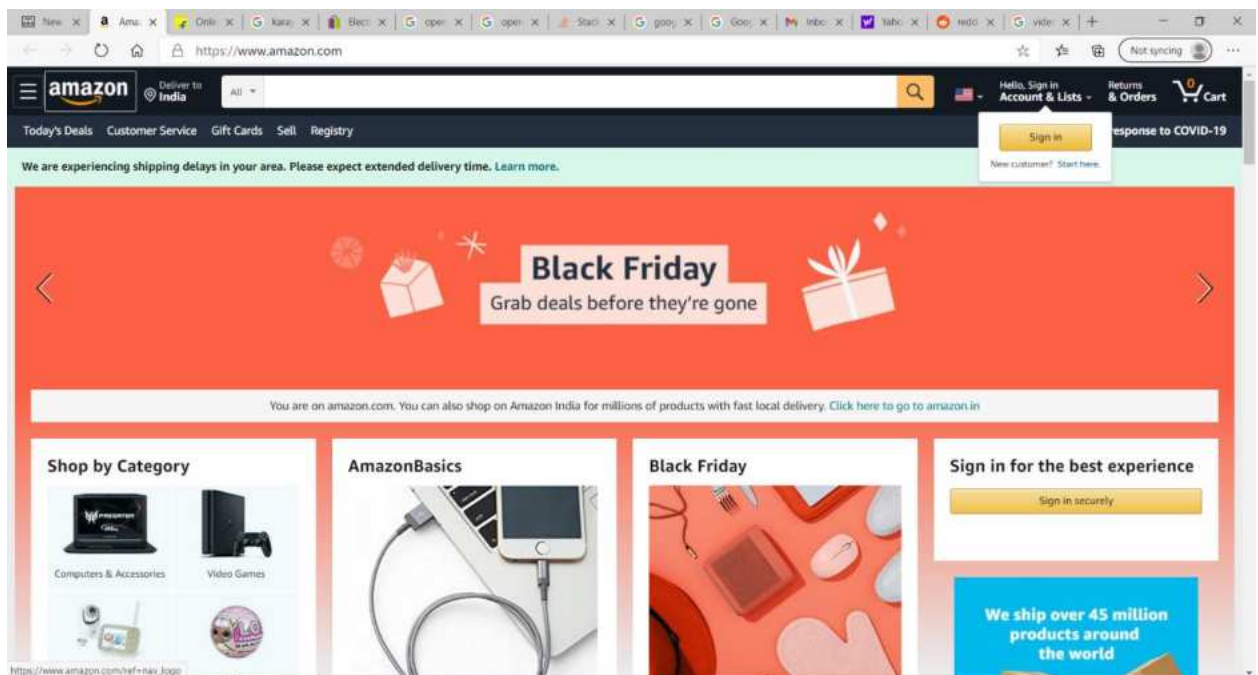
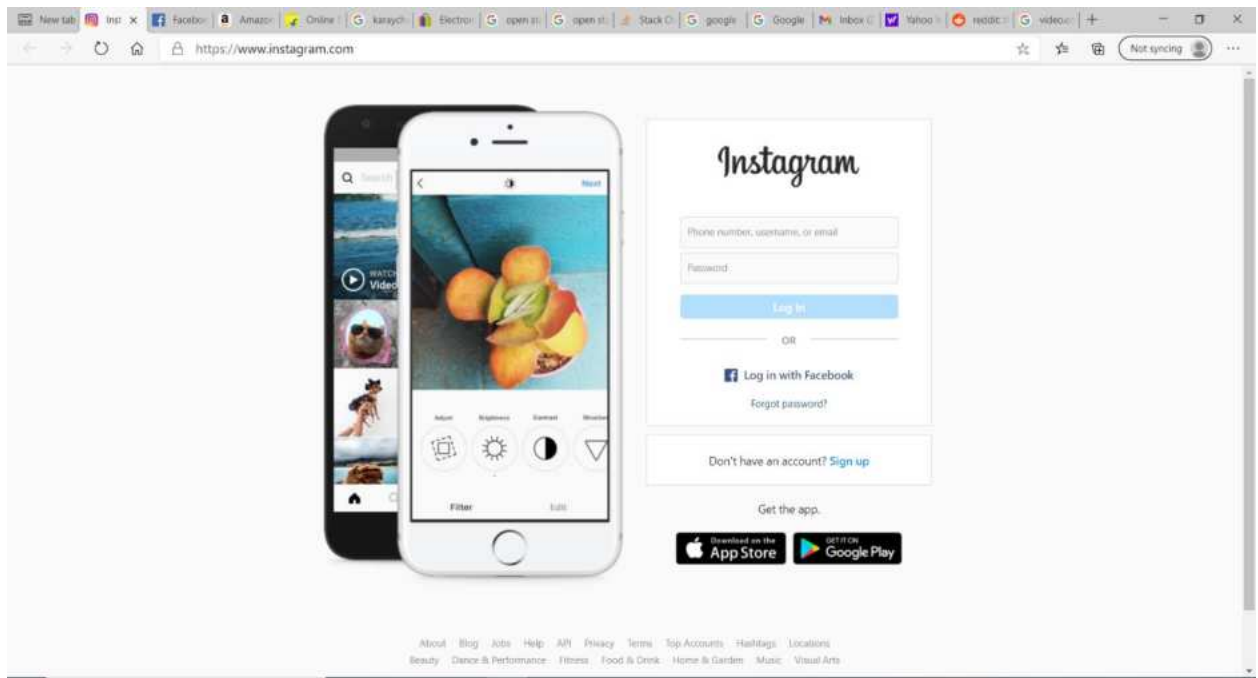
```

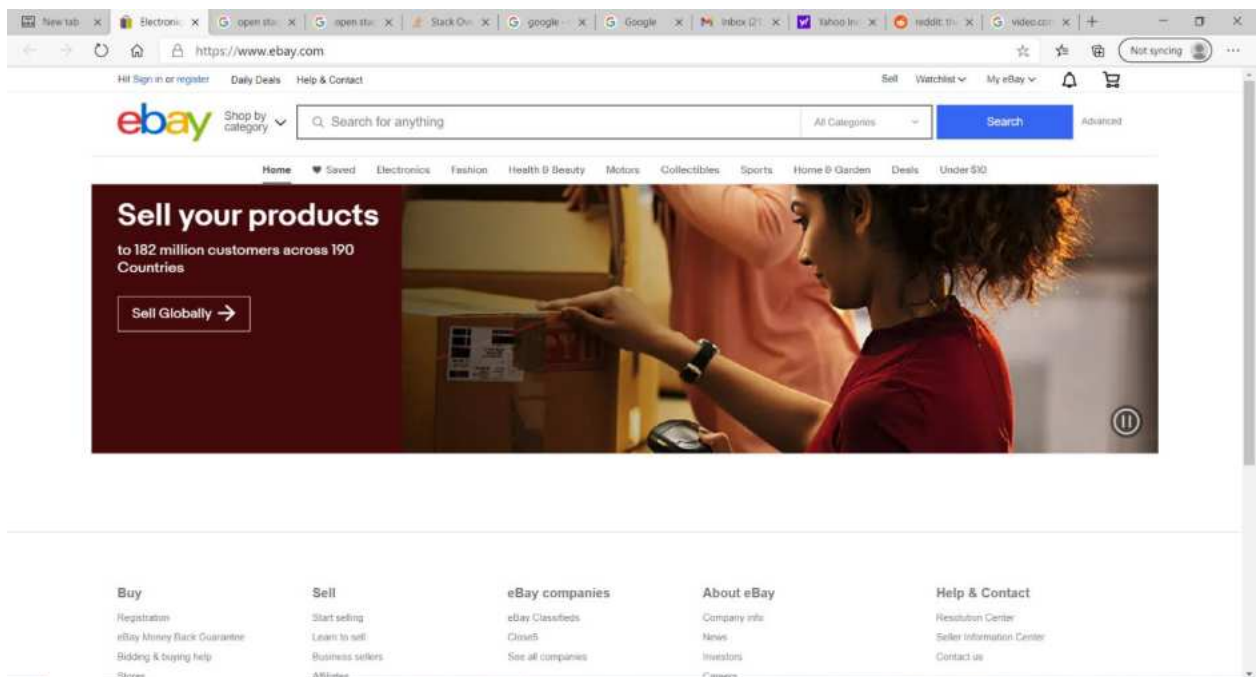
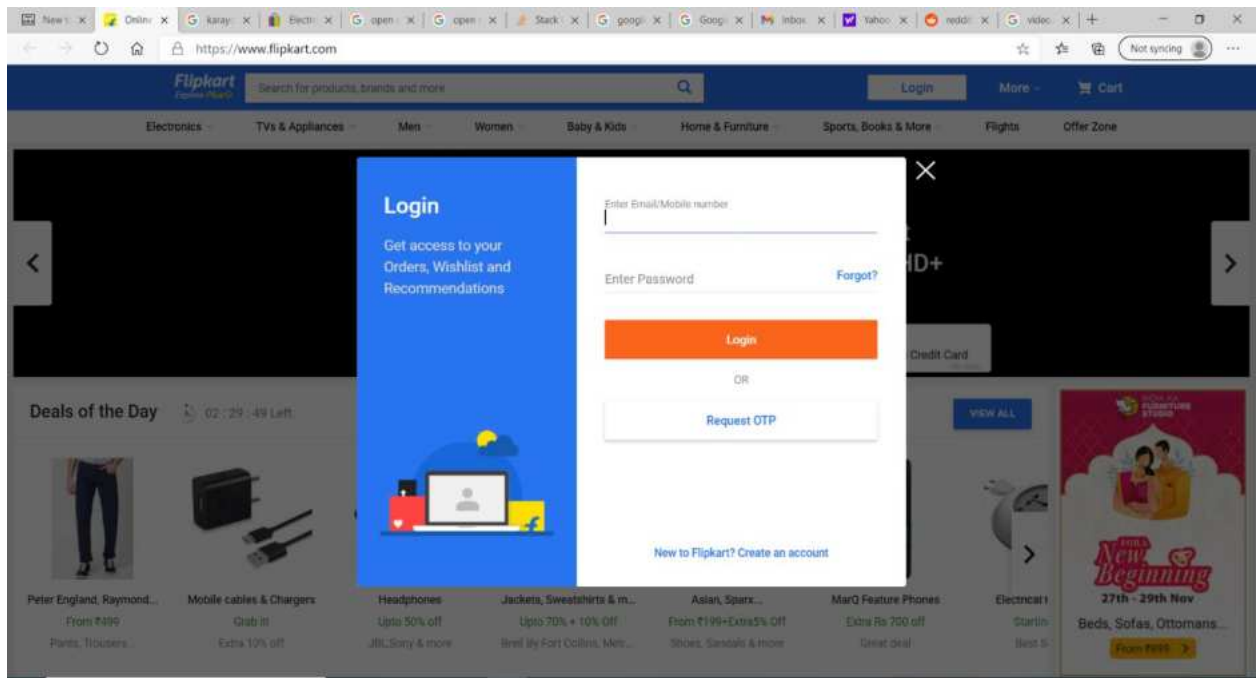
```
        speak(ex_exit)
        exit()
    else:
        temp = query.replace(' ', '+')
        g_url="https://www.google.com/search?q="
        res_g = 'sorry! i cant understand but i can search from internet
to give your answer ! okay'
        print(res_g)
        speak(res_g)
        webbrowser.open(g_url+temp)
```

## SCREENSHOTS

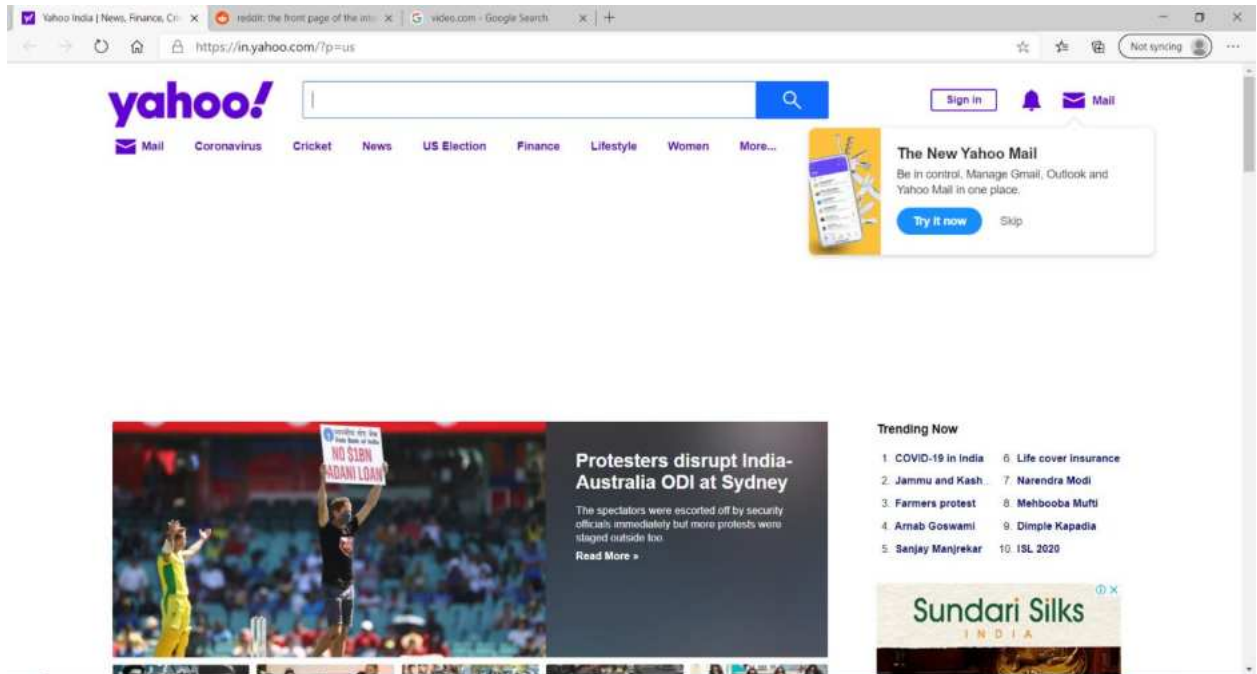
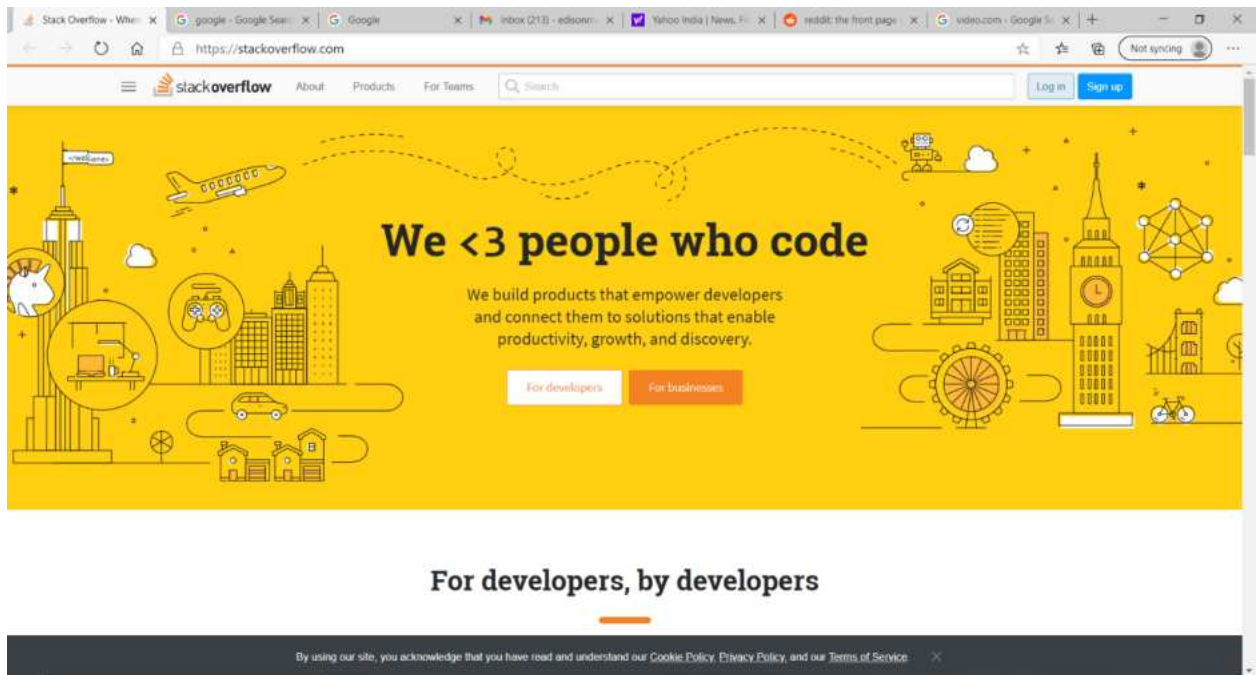


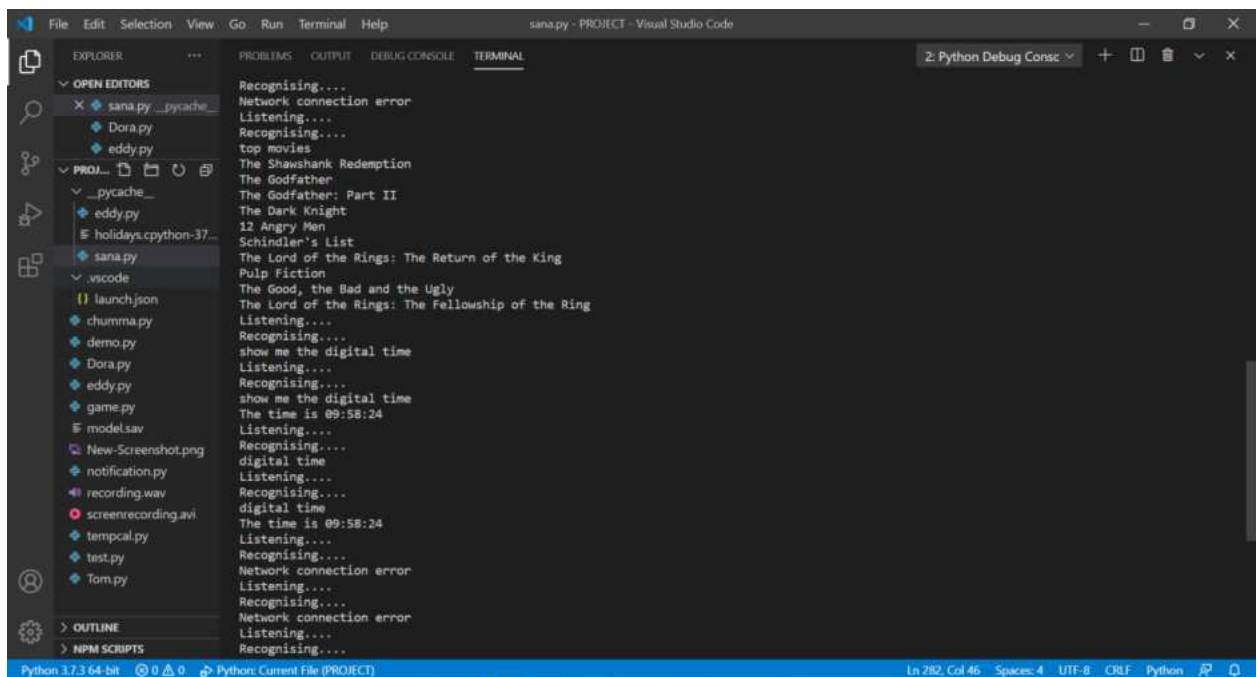
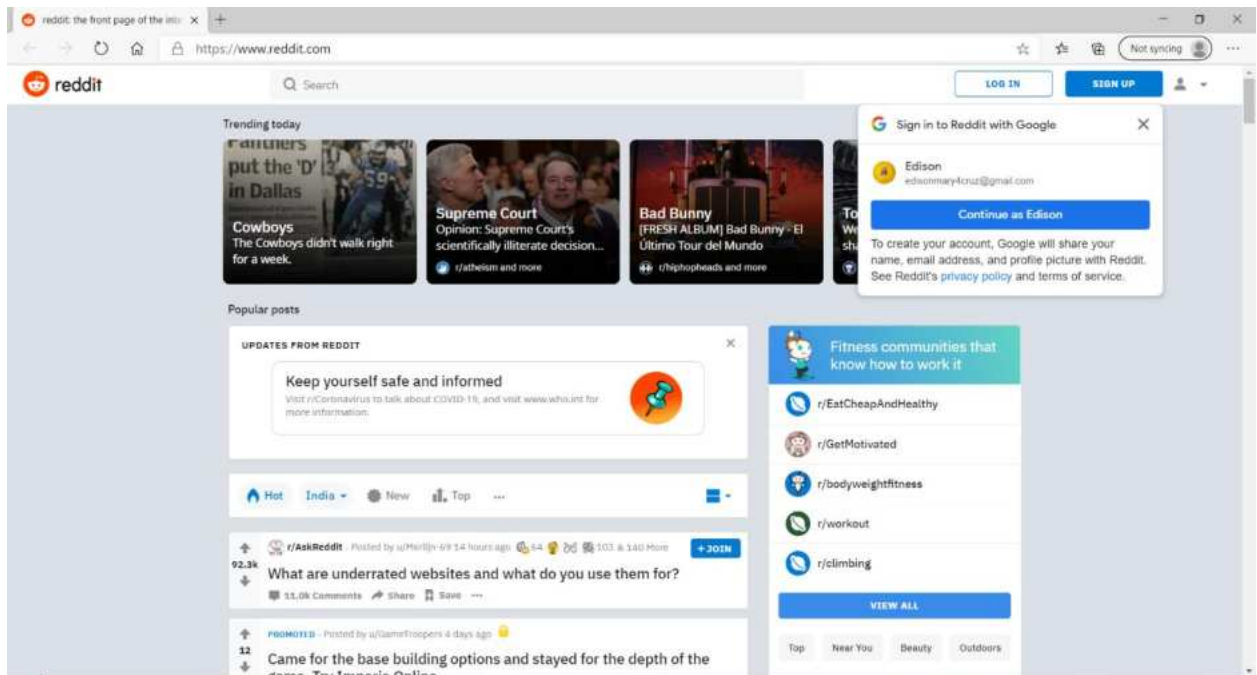




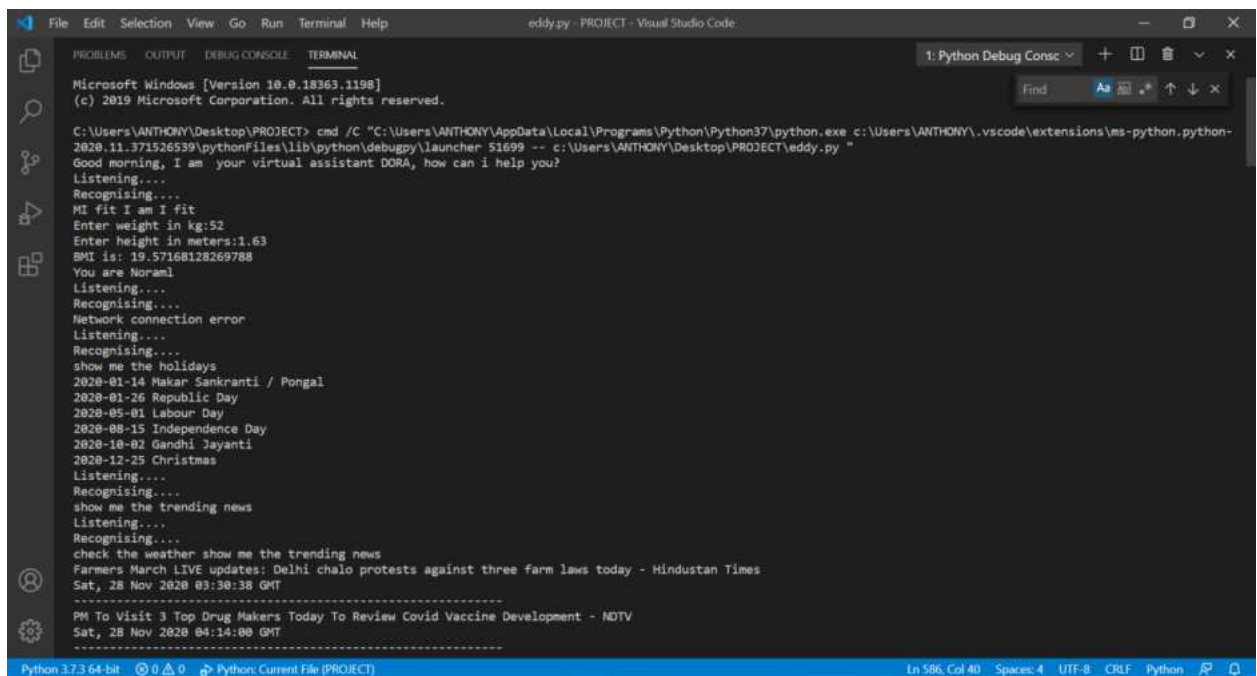
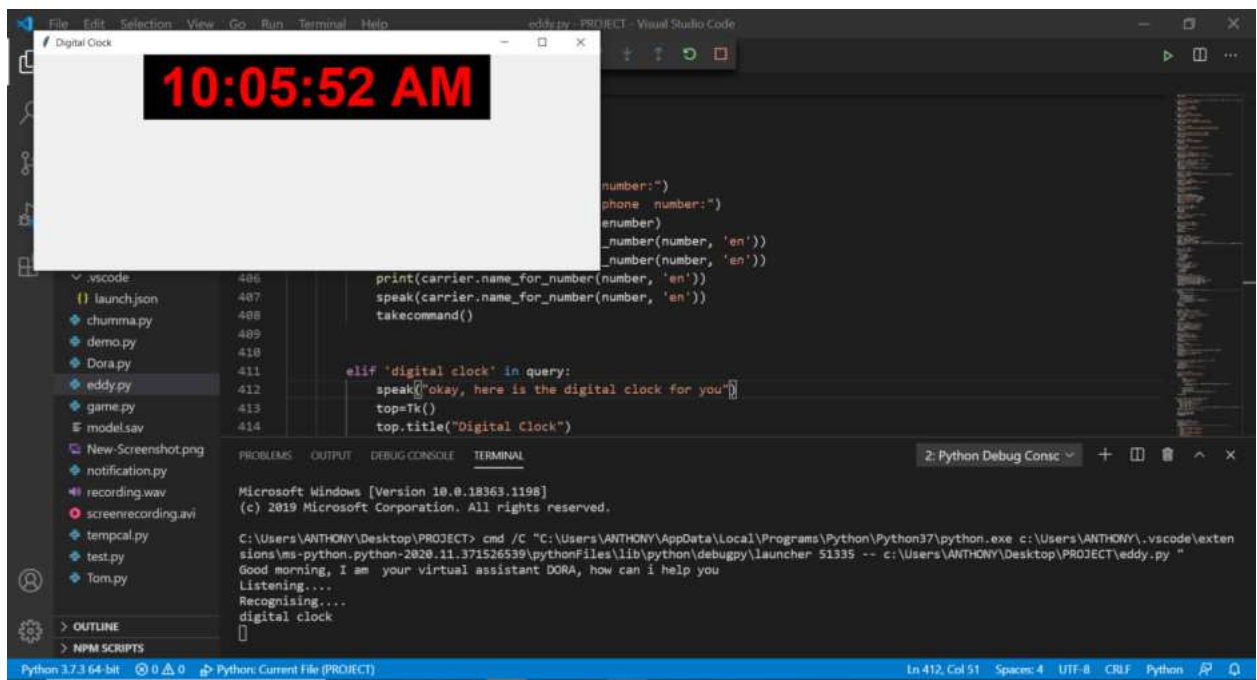










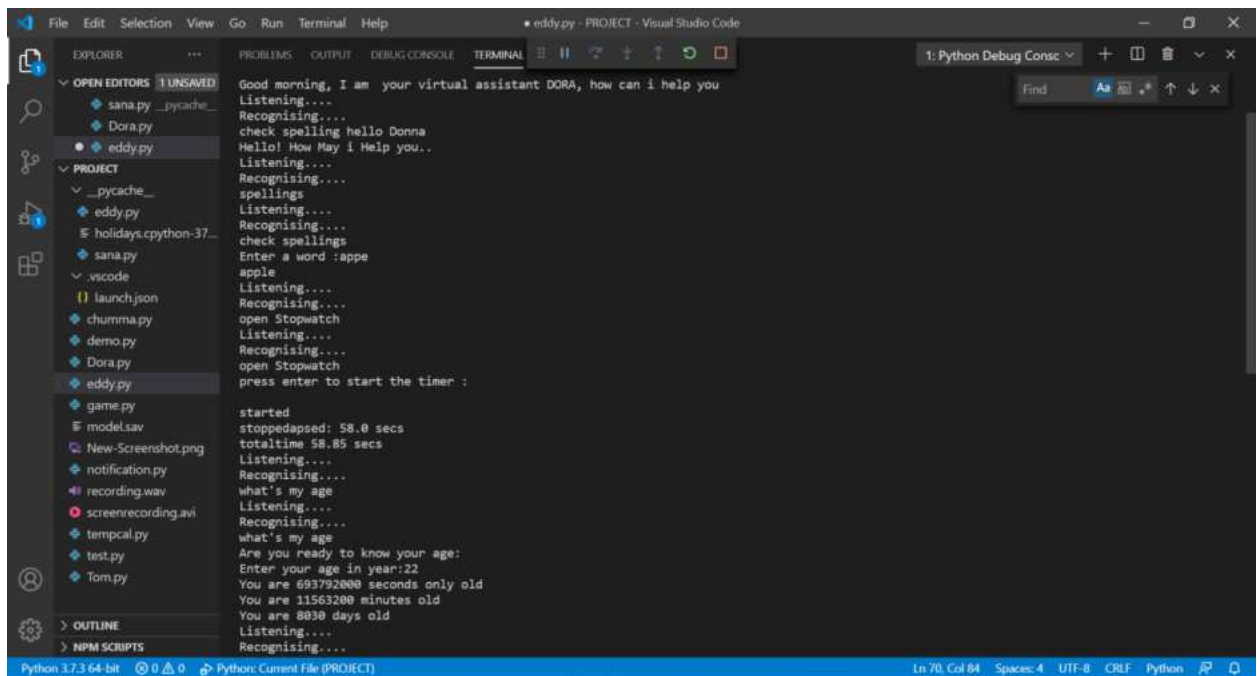
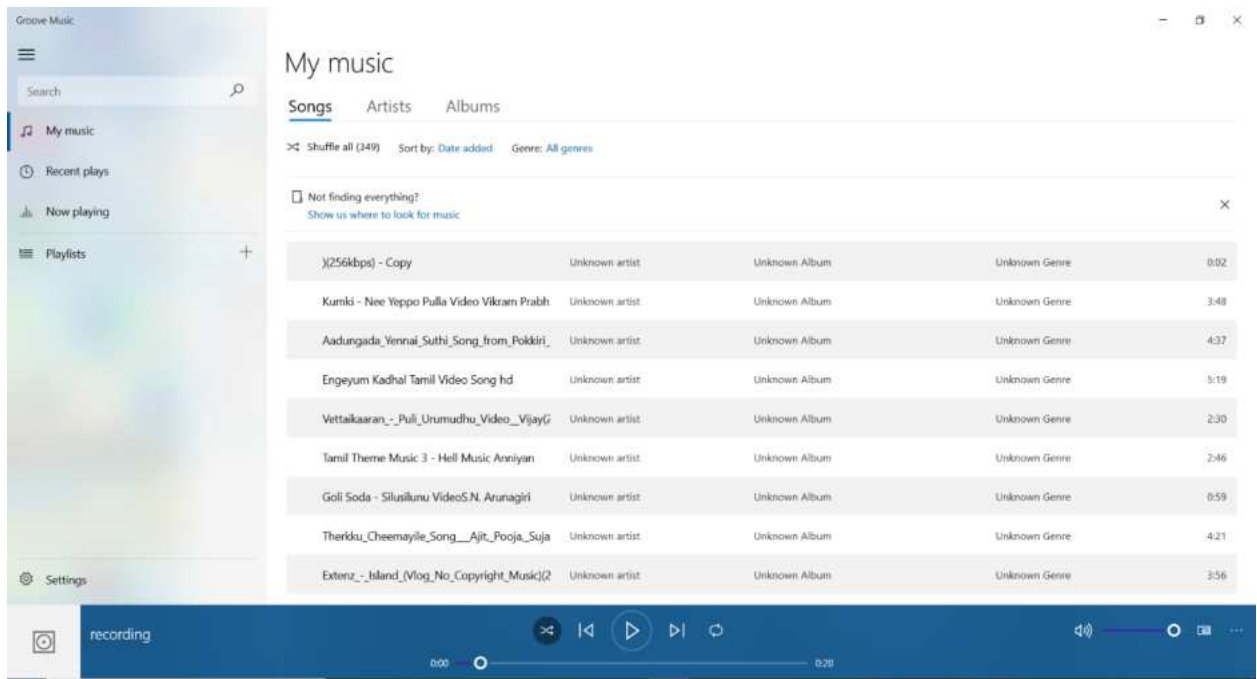


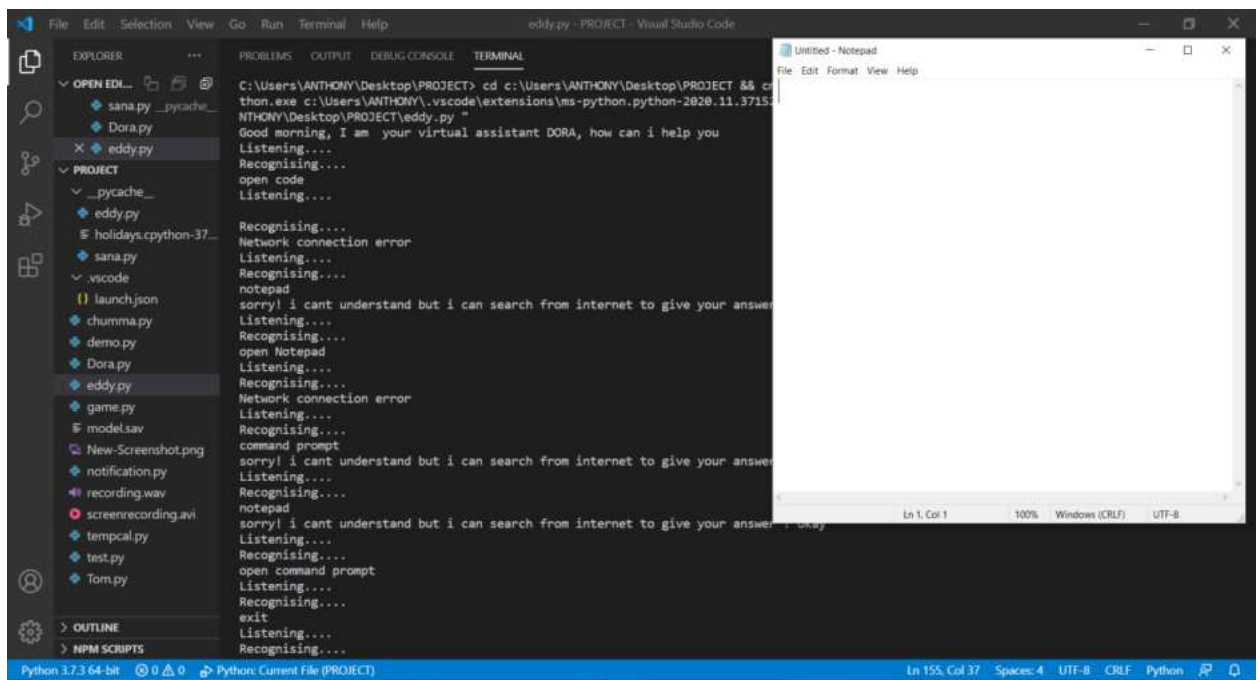
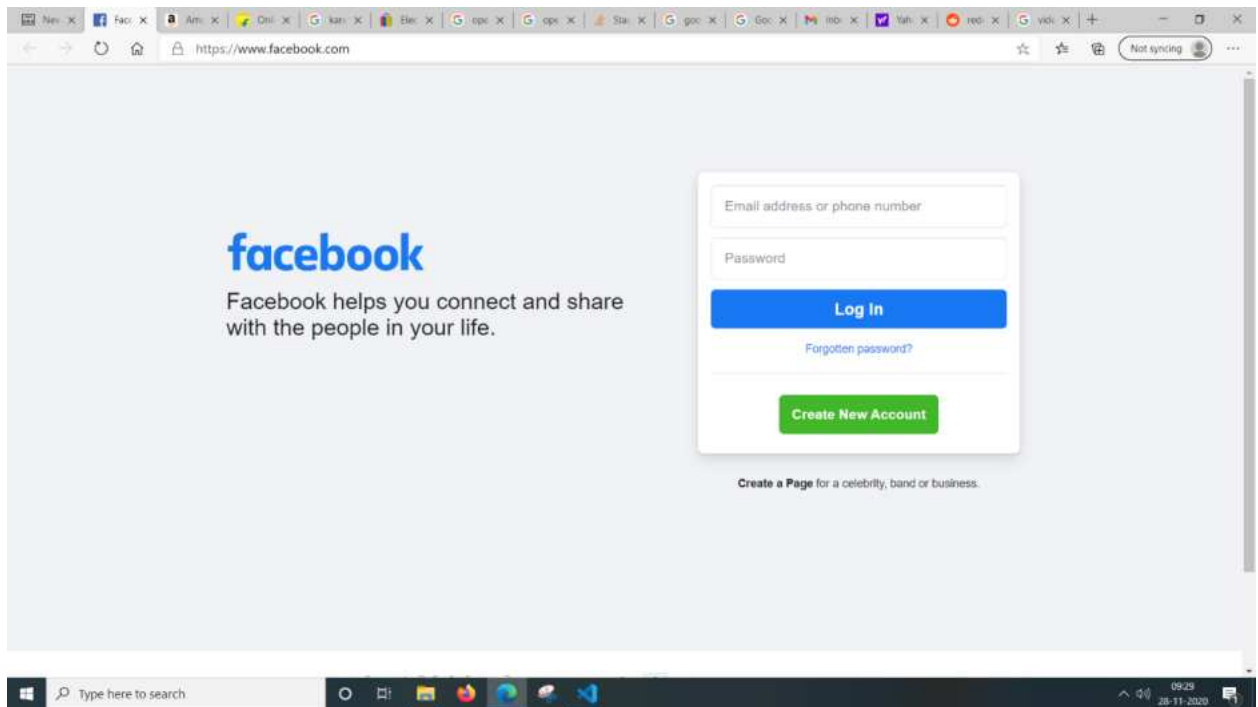
The screenshot shows the Visual Studio Code interface with the 'TERMINAL' tab active. The terminal displays a series of news headlines, each preceded by a date and time in GMT. The headlines are separated by dashed lines. The status bar at the bottom indicates 'Python 3.7.3 64-bit' and 'Python: Current File (PROJECT)'.

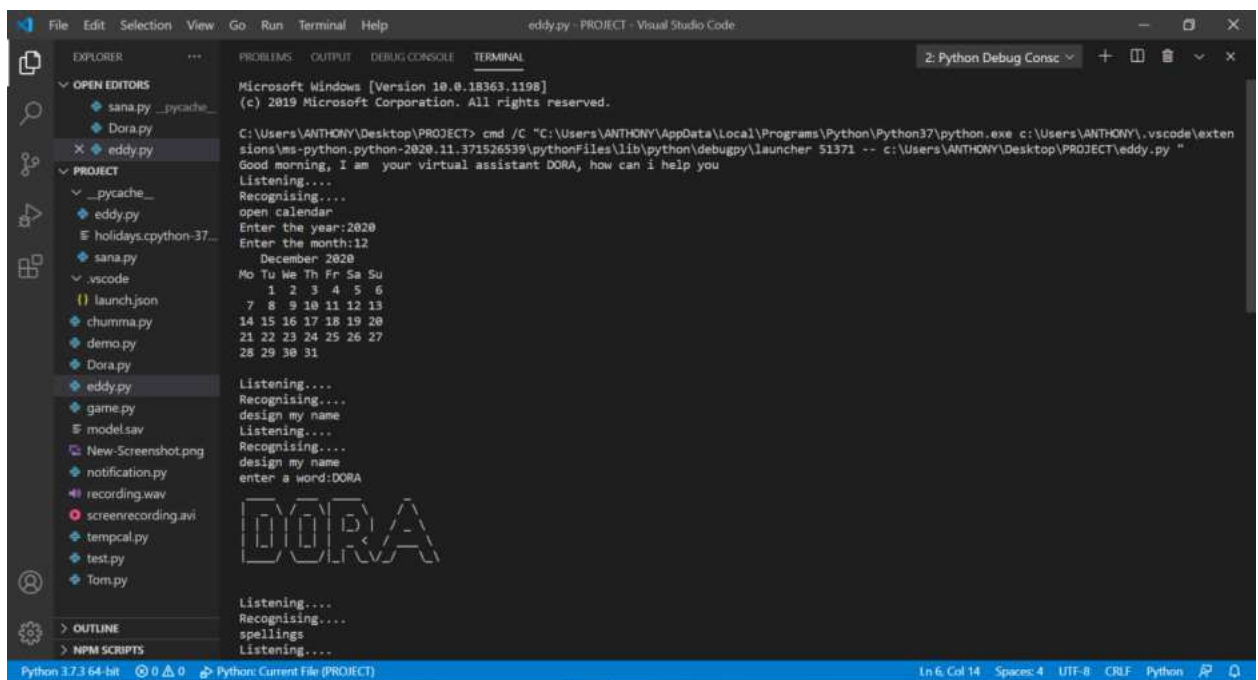
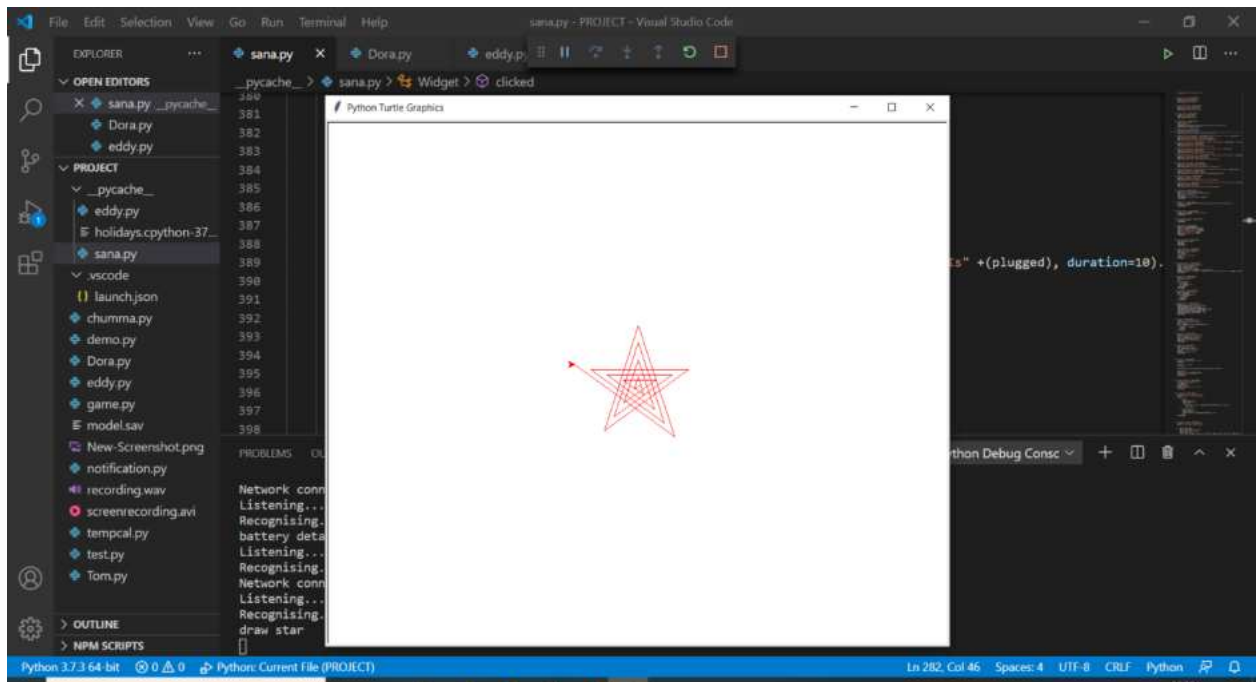
```
-----
Rahul Gandhi seeks opinion from Bengal leaders on alliance with Left against TMC, BJP - Hindustan Times
Fri, 27 Nov 2020 18:25:18 GMT
-----
It's time to reclaim love from the Hindutva hate jihad - Scroll.in
Fri, 27 Nov 2020 01:00:00 GMT
-----
Ten Covid-19 vaccines seen by mid-year, head of global pharma group says - Times of India
Fri, 27 Nov 2020 16:38:00 GMT
-----
Joe Biden will be a president who represents the best in us: Kamala Harris - Moneycontrol
Sat, 28 Nov 2020 03:31:00 GMT
-----
'Don't Ever Talk to The President that Way': Donald Trump Snaps at Reporter Over Election Question | Watch - India.com
Fri, 27 Nov 2020 11:10:35 GMT
-----
Explained: Takeaways from Oxford University Covid-19 vaccine error - The Indian Express
Thu, 26 Nov 2020 21:51:21 GMT
-----
GDP shrinks 7.5% in Q2; manufacturing picks up, services sector worst hit - The Indian Express
Sat, 28 Nov 2020 03:25:44 GMT
-----
LVB-DBS amalgamation: Madras HC's interim order gives some relief to LVB shareholders - Moneycontrol.com
Sat, 28 Nov 2020 03:38:00 GMT
-----
Eight core industries' output contracts 2.5 per cent in October, 8th contraction in a row - Economic Times
Fri, 27 Nov 2020 14:51:00 GMT
-----
"10 Covid Vaccines Likely By 2021 Summer": Global Pharma Group Head - NDTV
Fri, 27 Nov 2020 15:46:17 GMT
-----
"I wasn't supposed to tell that": PUBG Mobile India pro Maxtern posts clarification on deleted viral tweet - Sportskeeda
Fri, 27 Nov 2020 18:50:00 GMT
-----
One for the future: Samsung Display teases a tri-fold foldable phone - gizmochina
Fri, 27 Nov 2020 15:29:00 GMT
-----
```

The screenshot shows the Visual Studio Code interface with the 'EXPLORER' and 'TERMINAL' tabs active. The 'EXPLORER' shows a file tree with 'PROJECT' expanded, listing files like 'sanapy', 'Dorapy', 'eddy.py', and 'game.py'. The 'TERMINAL' shows a series of news headlines, each preceded by a date and time in GMT. The status bar at the bottom indicates 'Python 3.7.3 64-bit' and 'Python: Current File (PROJECT)'.

```
-----
Russia hails launch of China's lunar probe - TASS
Tue, 24 Nov 2020 11:16:54 GMT
-----
6 easy hacks to reduce excess belly fat - Times of India
Sat, 28 Nov 2020 01:30:00 GMT
-----
India holds key to global fight against coronavirus pandemic - WION
Fri, 27 Nov 2020 18:16:05 GMT
-----
New UNAIDS report suggests getting HIV response back on track - The Indian Express
Fri, 27 Nov 2020 18:21:01 GMT
-----
What is The Difference Between Covid Vaccine's Efficacy Vs Effectiveness? | NewsMo - India Today
Fri, 27 Nov 2020 16:34:05 GMT
-----
Listening....
Recognising....
check the weather
Listening....
Recognising....
set the weather: check the weather
Enter city name : Bangalore
Temperature (in kelvin unit) = 297.27
atmospheric pressure (in hPa unit) = 1018
humidity (in percentage) = 64
description = scattered clouds
Listening....
Recognising....
how are you how are you
Listening....
Recognising....
how are you
Listening....
Recognising....
ok
Listening....
```







```
File Edit Selection View Go Run Terminal Help eddy.py - PROJECT - Visual Studio Code
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
OPEN EDITORS
  sanapy _pycache_
  Dorapy
  x eddy.py
  _pycache_
    eddy.py
    holidays.cpython-37...
  sanapy
  .vscode
  launch.json
  chumma.py
  demo.py
  Dorapy
  eddy.py
  game.py
  modelsav
  New-Screenshot.png
  notification.py
  recording.wav
  screenrecording.avi
  tempcal.py
  test.py
  Tom.py
  > OUTLINE
  > NPM SCRIPTS
  Recognising....
  how are you feeling
  Listening....
  Recognising....
  Network connection error
  Listening....
  Recognising....
  new feeling
  sorry! i cant understand but i can search from internet to give your answer ! okay
  Listening....
  Recognising....
  love you feeling
  feeling Very sweet after meeting with you
  Listening....
  Recognising....
  take screenshot
  Okay, taking screenshot
  C:\Users\ANTHONY\AppData\Local\Temp\tmpu2onua\j.PNG
  The process cannot access the file because it is being used by another process.
  Listening....
  Recognising....
  Network connection error
  Listening....
  Recognising....
  record
  Listening....
  Recognising....
  video screen recording
  Listening....
  Recognising....
  video screen recording
  Listening....
  Recognising....
  bye shut down
  Listening....
  Recognising....
  exit
```

# CONCLUSION

## Advantages

Users can talk on the phone without putting it near to the ears:  
Voice user interface enables the users to speak on the phone without having to put it near to their ears. If phone rings when you are in the middle of inconvenient circumstances like bathing,

Help you save time:

If you know how VUI device works, it can be like an extra hand for you and helps you save a lot of time. When you are driving, you can communicate with Google Maps and ask it for direction to the place you want to go. This will reduce your chances of getting lost. It prevents you from having to take out your phone and navigate to the map to find the direction.

Assist in phone operation tasks:

It can take over a lot of tedious jobs related to phone operation so that your employees can be positioned in better places. It can handle incoming calls and SMS text messages. It can also route your phone calls to another phone number. You can specify it to take down notes as it can turn your spoken words into text typed in the document. It is the perfect tool for administrators who want to spend most of their time in front of the computers and not with notes strewn around their desks.

Help users that suffer from Physical Impairment:

It can be a great tool for people that suffer from impairments.



The VUI device can read to the visually impaired users the message in the emails or articles on the website or documents. For users with hearing impairment, it can convert audiobooks and messages into text for reading on the computer screen. It is troublesome for disabled people to use their hands to interact with the products. Incorporating technology into your product will allow you to have a competitive advantage for competitors with less accessible products.

Streamline communication with foreigners:

Voice recognition device can streamline the communication between people that speak different languages. This is because it has a built-in translation tool that can automatically translate what is said in a foreign language into a native language that you can understand. It can help travelers to overcome the language barrier so that they can access the necessary information with ease.

Can get things done fast:

Talking is faster compared to typing - it is faster to give a voice command than have to walk to another room in the house to get a task completed. It is easier to tell Google to turn on the thermostat instead of coming out of bed to turn it up yourself. The technology is making life easier as it is being built into more and more devices. It is fast enough to be used for business purposes to complete tasks.

Improve your Speech Recognition Skills:



**You are training yourself on how to talk to a computer by using speech recognition devices. Many businesses are now incorporating speech recognition into their operations. If you have the device at home, you have the skill set to communicate with the speech recognition device. This can be helpful when you are looking for jobs that require this skill set or you have to face with robot job interviewers.**

VUI Technology is Evolving:

VUI technology is continuously improving. A few years back, it was hard to get the VUI right. Many people have experienced this problem before when communicating with an automated customer care center. Now, communication with the machine is becoming more and more like human conversation and lesser and lesser like robotic. This demonstrates that the researchers are doing their best to improve the VUI devices in a way that will fit into human society.

Can control multiple products efficiently

Virtual assistants like Google Home allow users to control over 1K smart home products. These products include kettles, microwave oven, and thermostats. Some devices like HomePod also allow the users to turn on several smart home devices with a single phrase like Good Morning.

## **Disadvantages**

### **Smart Home Devices are Expensive:**

Smart home devices which are controlled by the virtual assistants are expensive, and not everyone can afford them. They also consume a lot of electricity. So, if you have a lot of smart home devices, you can expect to pay a higher electricity bill. When there is no electricity, you cannot use them.

### **Leads to lost concentration on the task in hand:**

Using a virtual assistant with voice recognition can cause you to lose concentration on the tasks in hand. You are giving attention to the virtual assistant when providing instructions and trying to accomplish the task in hand at the same time. When the attention is divided, you are prone to making a mistake. So, before giving instructions to a virtual assistant, make sure to stop whatever you are doing.

### **Privacy Concerns:**

If you use a VUI device, your data can get tracked by the manufacturer. There have been complaints about this, for example, the Cambridge Analytica scandal claimed that Google Home and Amazon Alexa are eavesdropping on private conversations. This causes people to have worries about how their private data are being used by these manufacturers. Aware of the issue, the manufacturers are now working on offering better privacy controls to the consumers so that they can use the technology in peace.

# BIBLIOGRAPHY

- [www.google.com](http://www.google.com)
- [Wikipedia](#)

## PYTHON PROJECT ASSIGNMENT STRUCTURE

### Python Course Assessment Structure:

Python Programming Level	Chapter	Practice Assignments (Non Graded)	Programming Assessments (Graded)	Quizzes (Graded)
<b>BASIC PYTHON</b>	1. Introduction & Installation	2	0	1
	2. Data Types & Data Structures	10	2	1
	3. Control Structures, File Exception Handling & Functions	33	3	1
	4. Introduction to Database & Web Servers	12	1	1
	<i>Mini Project &amp; Quiz</i>	<i>0</i>	<i>1</i>	<i>0</i>
<b>ADVANCE PYTHON</b>	5. Object Oriented Programming	10	2	1
	6. Algorithms	10	2	1
	7. Advance Functional Programming, Data Structures, Regular Expressions, Modules & Directories	12	2	1
	8. Libraries	12	2	1
	<i>Mini Project &amp; Quiz</i>	<i>0</i>	<i>1</i>	<i>0</i>
<b>EXPERT PYTHON</b>	9. Introduction to GUI & Database Apps	0	0	0
	10. Data Analysis	6	3	1
	11. Web Application Using Django	2	1	1
	12. REST Framework API's Using Python & Django	3	2	1
	<i>Mini Project &amp; Quiz</i>	<i>0</i>	<i>1</i>	<i>0</i>
<i>Final Project &amp; Quiz</i>		<i>0</i>	<i>1</i>	<i>0</i>

<i>Resume Writing &amp; Self Promotion Video</i>	<i>0</i>	<i>1</i>	<i>0</i>
<i>Mock Interview</i>	<i>0</i>	<i>1</i>	<i>0</i>
<i>Aptitude Test</i>	<i>0</i>	<i>0</i>	<i>1</i>
<b>Total Course Assignments</b>	<b>112</b>	<b>26</b>	<b>12</b>