

# PROJECT - CUSTOMER RELATIONSHIP MANAGEMENT (CRM)



**DONE BY,  
INDHU R V**

# INTRODUCTION :

## What is CRM ?

CRM stands for Customer Relationship Management. It refers to a strategy and a set of practices, technologies, and systems that businesses use to manage and analyze customer interactions and data throughout the customer lifecycle, with the goal of improving customer service relationships and assisting in customer retention and sales growth.

A CRM system helps businesses manage their interactions with current and potential customers. It typically involves collecting customer data from various sources, such as email, telephone calls, social media, and website visits, and consolidating it into a centralized database. This data can include contact information, purchase history, preferences, and any other relevant details.

## PURPOSE OF CRM :

The purpose of CRM (Customer Relationship Management) is multifaceted, aiming to enhance the overall customer experience and improve business operations. Here are some key purposes:

- **Customer Data Management**
- **Improved Customer Service**
- **Sales Enablement**
- **Marketing Optimization**
- **Customer Insights and Analytics**
- **Enhanced Collaboration**
- **Customer Retention and Loyalty**
- **Business Growth and Profitability**

Now , our project is to download customer data in csv format

A CSV (Comma-Separated Values) file is a plain text file format used to store tabular data, such as spreadsheets or databases. In a CSV file, each line of the file corresponds to a row of the table, and the values within each row are separated by commas (hence the name).

**Let's create a simple CRM (Customer Relationship Management) project in Python that allows users to download customer data in CSV format**

**Creating a CRM (Customer Relationship Management) project to download customer data in CSV format involves several steps. Here's a step-by-step**

### **STEP 1:**

There are numerous CRM (Customer Relationship Management) platforms available online, each offering a variety of features and tailored solutions to suit different business needs. Some of the popular CRM platforms include: Salesforce, HubSpot sales CRM , Zoho CRM, Microsoft Dynamics 365 etc

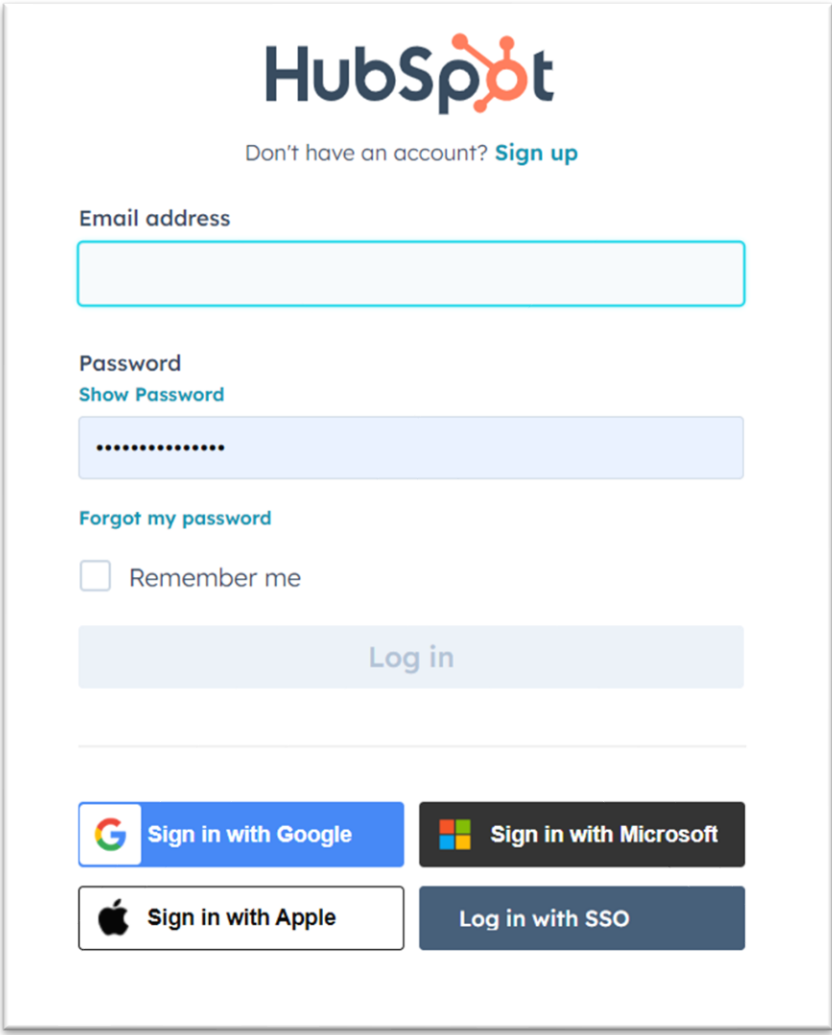
We are choosing HubSpot sales CRM platform

Firstly, to access customer data in HubSpot Sales CRM .

You can do that by the following steps:

#### **1. Log in to HubSpot:**

- Go to the HubSpot website and log in to your HubSpot Sales CRM account using your credentials.



The image shows the HubSpot login page. At the top is the HubSpot logo. Below it is a link that says "Don't have an account? Sign up". There are two input fields: "Email address" and "Password". The password field has a "Show Password" link next to it. Below the password field is a link that says "Forgot my password". There is a checkbox labeled "Remember me". A "Log in" button is centered below the checkbox. Below the "Log in" button are four buttons for social login: "Sign in with Google", "Sign in with Microsoft", "Sign in with Apple", and "Log in with SSO".

## 2. Navigate to Contacts:

- Once logged in, navigate to the "Contacts" section in the HubSpot dashboard. You can find it in the left sidebar menu.

## 3. View Contact Records:

- In the Contacts section, you'll see a list of all your contacts. You can scroll through the list or use the search bar to find specific contacts.

Contacts

2 records

Data Quality

Actions

Import

Create contact

All contacts

X

My contacts

Unassigned contacts

+ Add view (3/5)

All views

Contact owner

Create Date

Last Activity D...



Lead Status

Advanced filters (0)

Search name, phone, emi

Export

Edit columns

	NAME	EMAIL	PHONE NUMBER	CONTACT OWNER	PRIMARY COMPANY
<input type="checkbox"/>	 Brian Halligan (Sampl...	bh@hubspot.com	--	Unassigned	--
<input type="checkbox"/>	 Maria Johnson (Samp...	emailmaria@hubspot.com	--	Unassigned	--

Click on a contact's name to view their contact record. Here, you'll find details such as name, email address, phone number, company, and any other

#### 4. Export Contacts:

- To export contacts, select the contacts you want to export by checking the checkboxes next to their names.
- Click on the "Actions" dropdown menu and select "Export". Choose the export format (CSV) and follow the prompts to download the exported file.

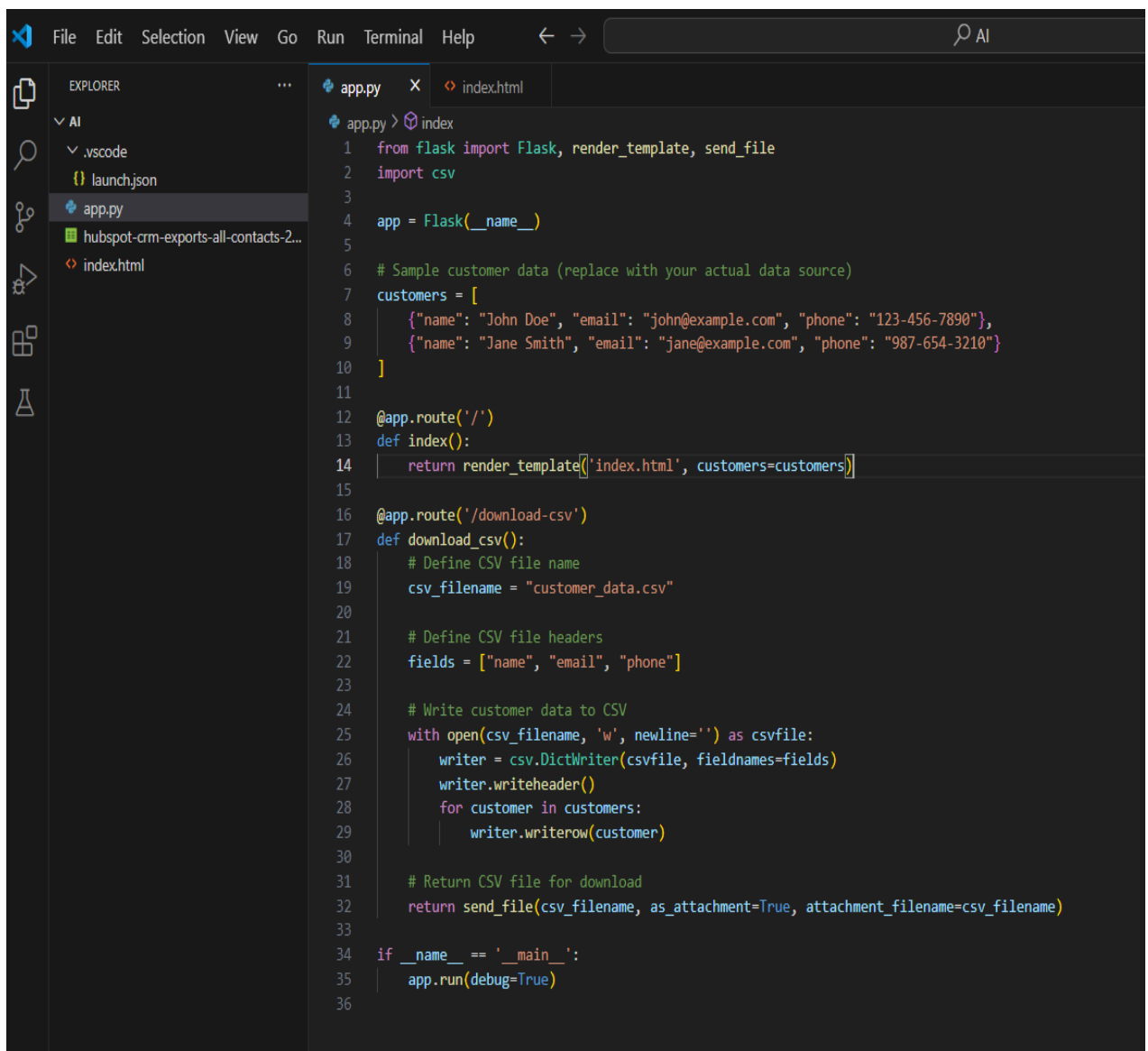
## STEP 2: Setting Up the Project Structure

Create a new directory for your project. Inside this directory, create the following files:

1. **app.py**: This will contain the main Flask application.
2. **templates/**: This directory will hold the HTML templates.

## STEP 3: Writing the Flask Application (app.py)

Open **app.py** in your text editor and write the following code:



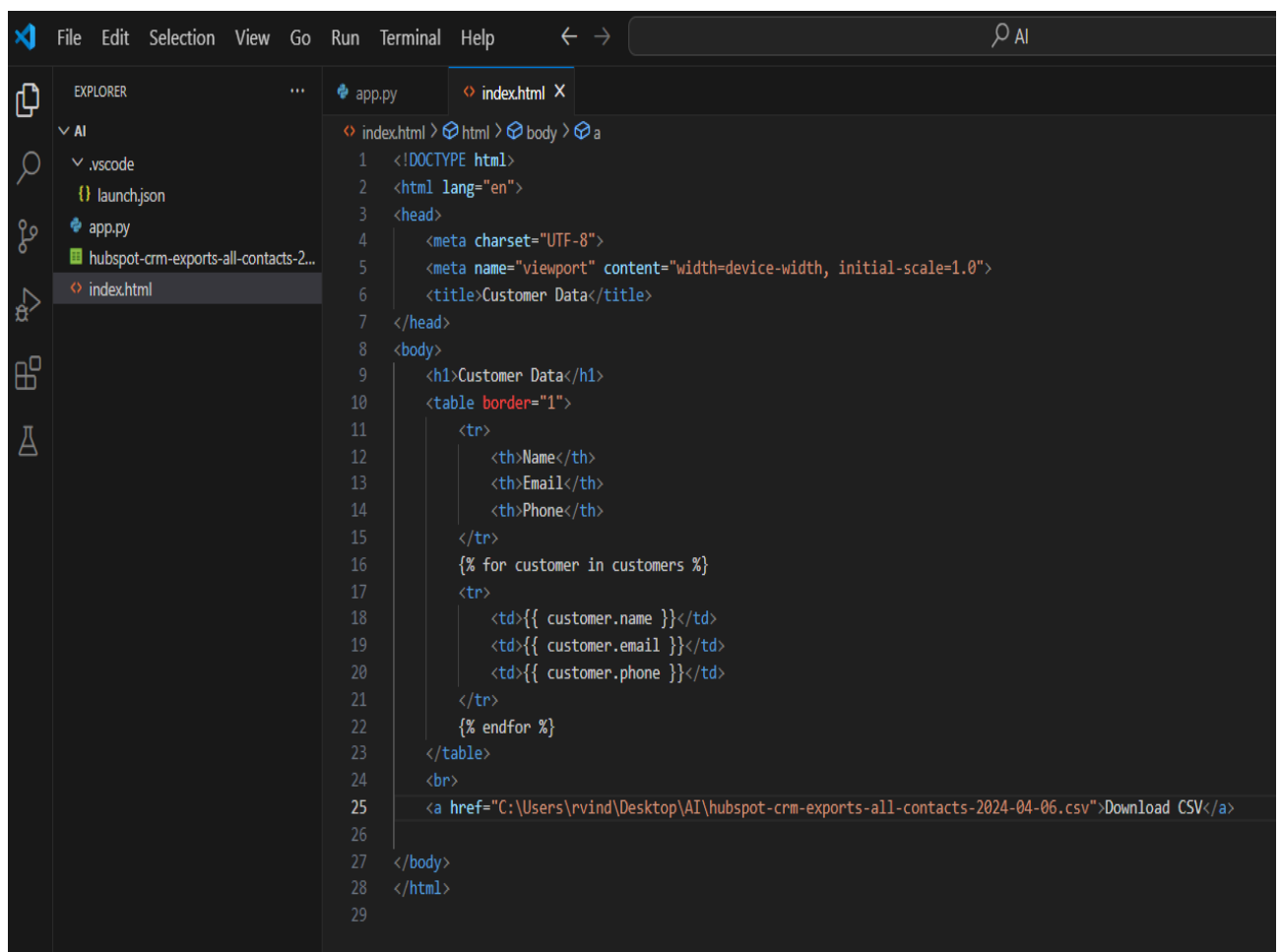
```
1 from flask import Flask, render_template, send_file
2 import csv
3
4 app = Flask(__name__)
5
6 # Sample customer data (replace with your actual data source)
7 customers = [
8     {"name": "John Doe", "email": "john@example.com", "phone": "123-456-7890"},
9     {"name": "Jane Smith", "email": "jane@example.com", "phone": "987-654-3210"}
10 ]
11
12 @app.route('/')
13 def index():
14     return render_template('index.html', customers=customers)
15
16 @app.route('/download-csv')
17 def download_csv():
18     # Define CSV file name
19     csv_filename = "customer_data.csv"
20
21     # Define CSV file headers
22     fields = ["name", "email", "phone"]
23
24     # Write customer data to CSV
25     with open(csv_filename, 'w', newline='') as csvfile:
26         writer = csv.DictWriter(csvfile, fieldnames=fields)
27         writer.writeheader()
28         for customer in customers:
29             writer.writerow(customer)
30
31     # Return CSV file for download
32     return send_file(csv_filename, as_attachment=True, attachment_filename=csv_filename)
33
34 if __name__ == '__main__':
35     app.run(debug=True)
36
```

## STEP 4: Creating HTML Template

Create a file named **index.html** inside the **templates/** directory. This file will contain the HTML template to display customer data and provide a link to download the CSV file.

### Download CSV Route:

- When accessing the **/download-csv** route, the application dynamically generates a CSV file named **customer\_data.csv**



The screenshot shows the Visual Studio Code editor with the `index.html` file open. The Explorer sidebar on the left shows the project structure with files like `launch.json`, `app.py`, and `index.html`. The main editor area displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Customer Data</title>
7 </head>
8 <body>
9   <h1>Customer Data</h1>
10  <table border="1">
11    <tr>
12      <th>Name</th>
13      <th>Email</th>
14      <th>Phone</th>
15    </tr>
16    {% for customer in customers %}
17    <tr>
18      <td>{{ customer.name }}</td>
19      <td>{{ customer.email }}</td>
20      <td>{{ customer.phone }}</td>
21    </tr>
22    {% endfor %}
23  </table>
24  <br>
25  <a href="C:\Users\rvind\Desktop\AI\hubspot-crm-exports-all-contacts-2024-04-06.csv">Download CSV</a>
26
27 </body>
28 </html>
29
```

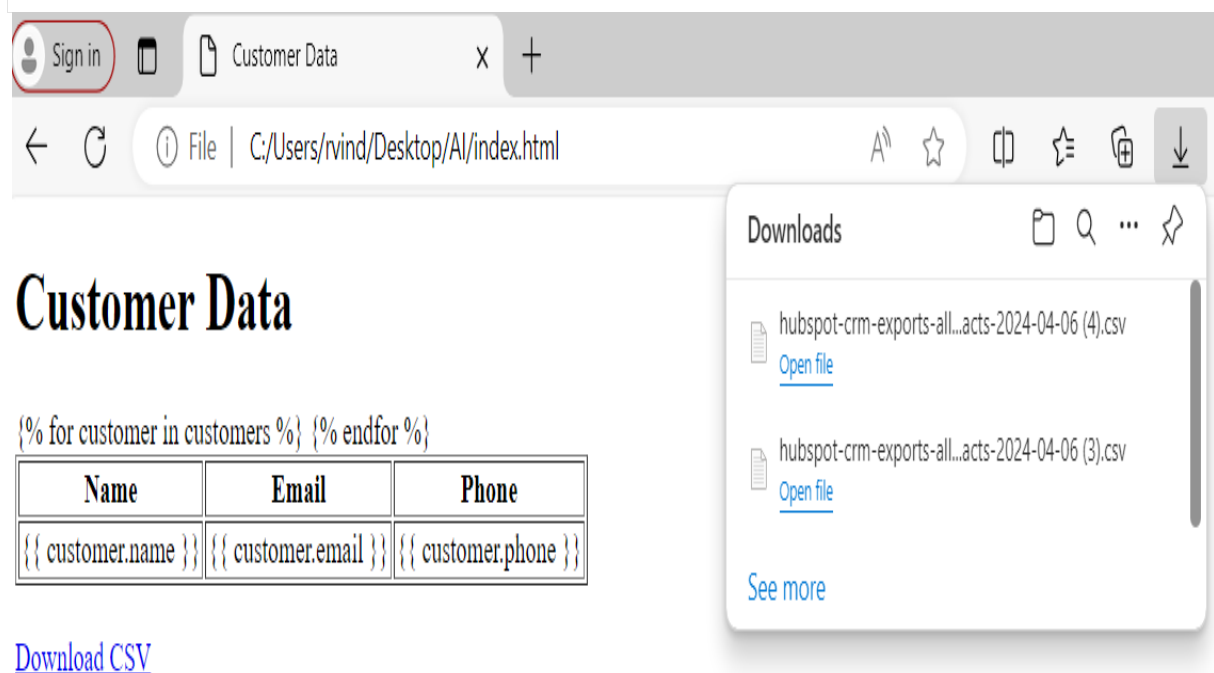


## STEP 5: Running the Application

Open a terminal, navigate to your project directory, and run the Flask application by executing `python app.py`.

## STEP 6 : Accessing the Application

Open a web browser and go to `http://localhost:5000`. You should see a list of customer data along with a link to download the CSV file



The screenshot shows a web browser window with a single tab titled 'Customer Data'. The address bar shows the file path 'C:/Users/rvind/Desktop/AI/index.html'. The page content includes a heading 'Customer Data', a Jinja2 template snippet, a table with three columns (Name, Email, Phone), and a link 'Download CSV'. A 'Downloads' panel is open on the right, displaying two CSV files: 'hubspot-crm-exports-all...acts-2024-04-06 (4).csv' and 'hubspot-crm-exports-all...acts-2024-04-06 (3).csv', each with an 'Open file' link.

Sign in

Customer Data

File | C:/Users/rvind/Desktop/AI/index.html

## Customer Data

{% for customer in customers %} {% endfor %}

Name	Email	Phone
{{ customer.name }}	{{ customer.email }}	{{ customer.phone }}

[Download CSV](#)

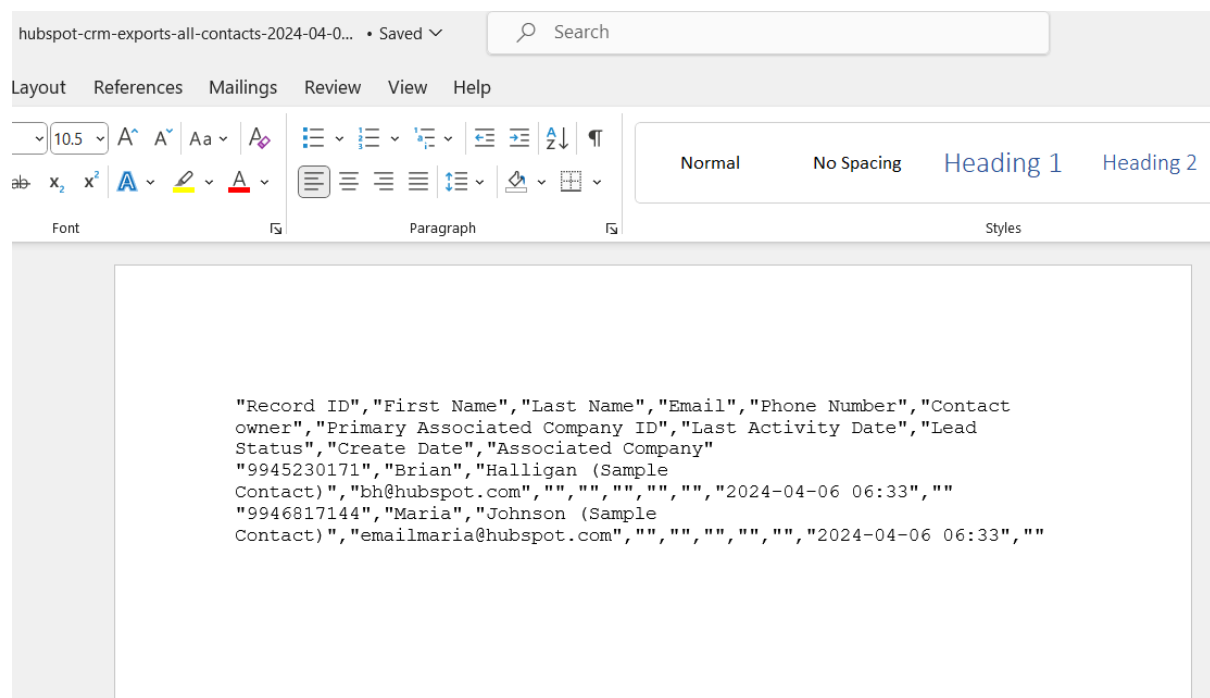
Downloads

- hubspot-crm-exports-all...acts-2024-04-06 (4).csv  
[Open file](#)
- hubspot-crm-exports-all...acts-2024-04-06 (3).csv  
[Open file](#)

[See more](#)

## STEP 7 : Output

Opening the downloaded CSV file should display the customer data in CSV format, with headers for name, email, and phone, and each row representing a customer's information.



**FILE:** I am Adding the Github repository link for this project here :

## **CONCLUSION:**

Hence, we have successfully created a simple CRM (Customer Relationship Management) project in Python that allows users to download customer data in CSV format using visual code idle .

In conclusion, the CRM project we've developed leverages Flask, a lightweight web framework in Python, to efficiently manage and present customer data. By providing a seamless interface for users to access and download data in CSV format, it enhances user experience and facilitates data utilization. With clear routes for data access and download, the application ensures simplicity and effectiveness. By continually refining and expanding the functionality, this CRM project can evolve to meet the evolving needs of businesses.