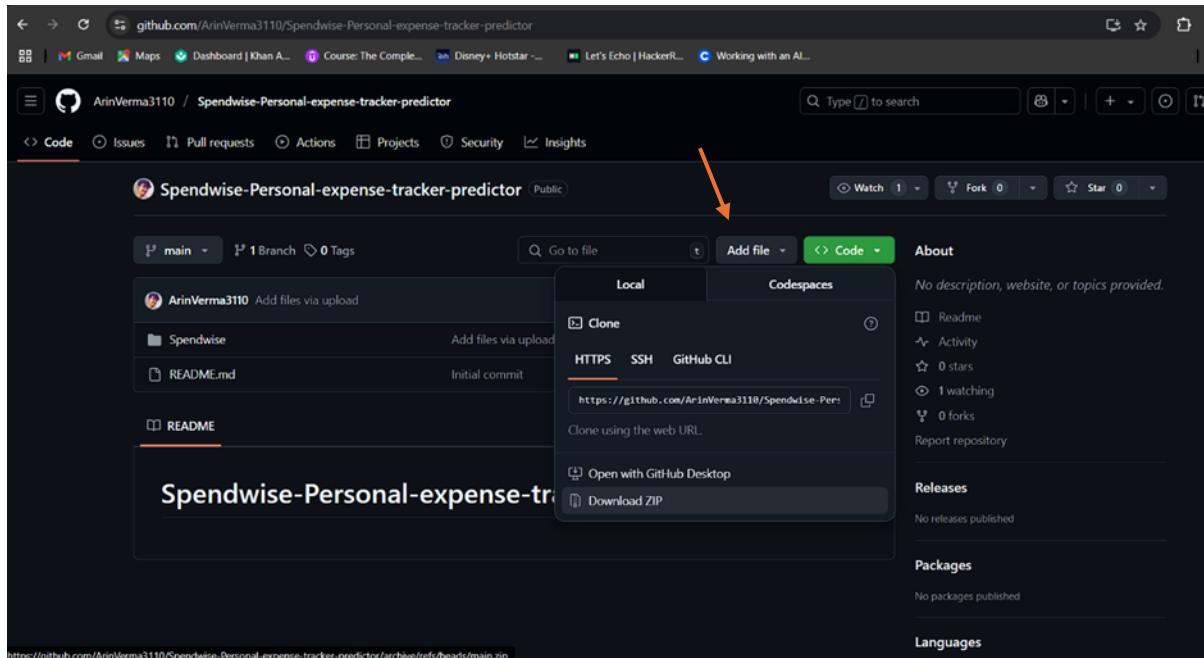


Steps to Set-up

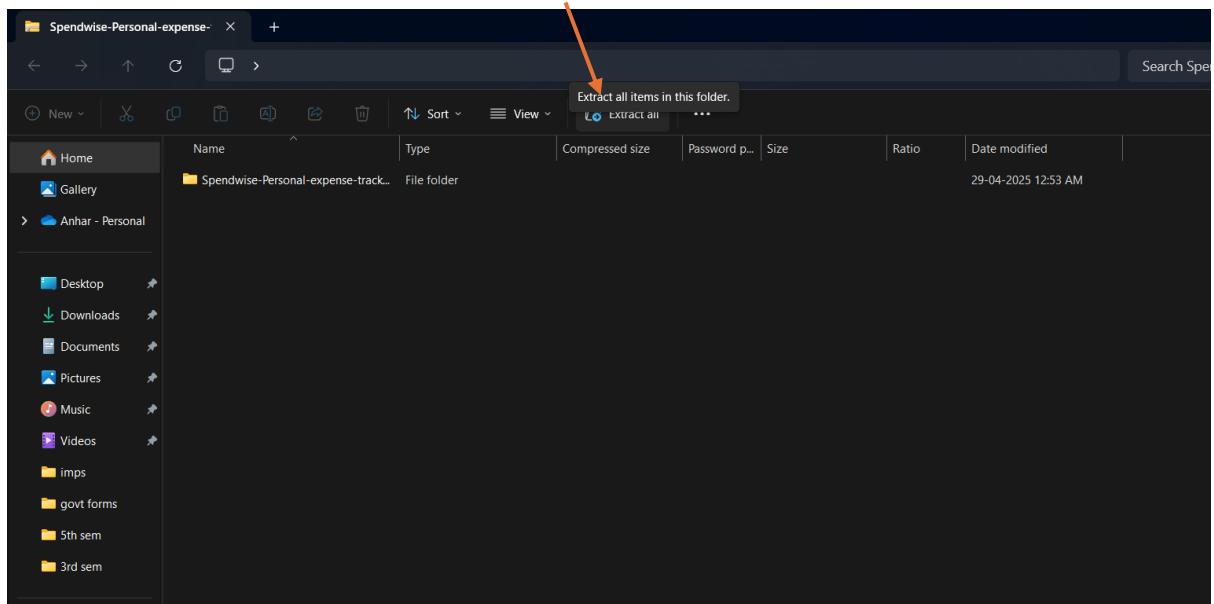
Step 1: Clone the Repository by downloading the zip file or write this command in your bash :

```
git clone https://github.com/ArinVerma3110/Spendwise-Personal-expense-tracker-predictor.git
```

```
cd Spendwise-Personal-expense-tracker-predictor
```

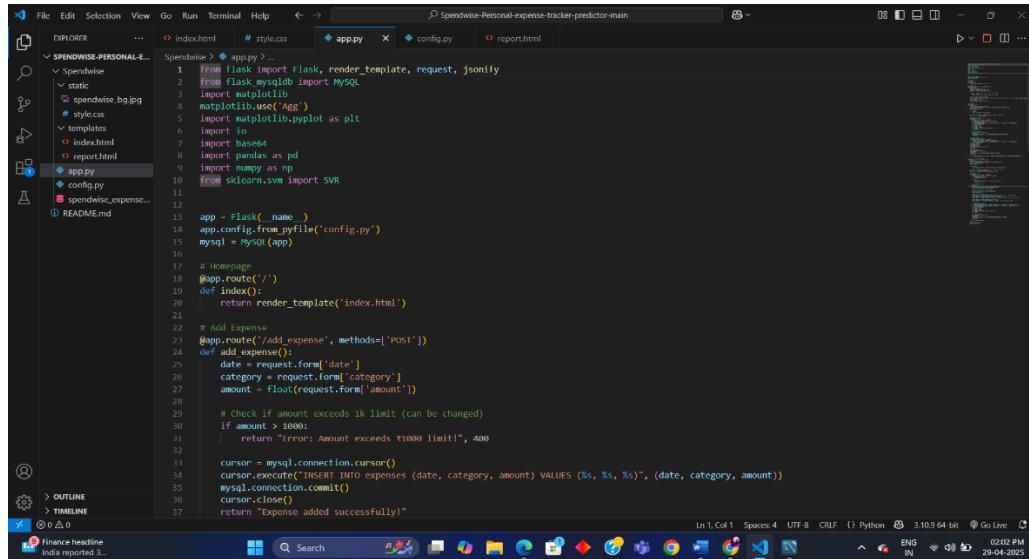


Step 2: Extract the Zip file.



Name	Status	Date modified	Type	Size
static	✓	29-04-2025 01:28 PM	File folder	
templates	✓	29-04-2025 01:28 PM	File folder	
app.py	✓	29-04-2025 01:27 PM	Python Source File	6 KB
config.py	✓	29-04-2025 02:04 PM	Python Source File	1 KB
spendwise_expenses.sql	✓	29-04-2025 01:27 PM	SQL Text File	11 KB

Step 3.1: Open the SpendWise Project Folder in your code editor and open all the files.



The screenshot shows the VS Code interface with the 'Spendwise-Personal-expense-tracker-predictor-main' project open. The Explorer sidebar shows files like index.html, style.css, app.py, config.py, and report.html. The code editor displays the contents of app.py, which imports Flask, MySQLdb, and various plotting libraries. It defines an application object and routes for the homepage and an expense addition endpoint. A cursor is visible in the code editor.

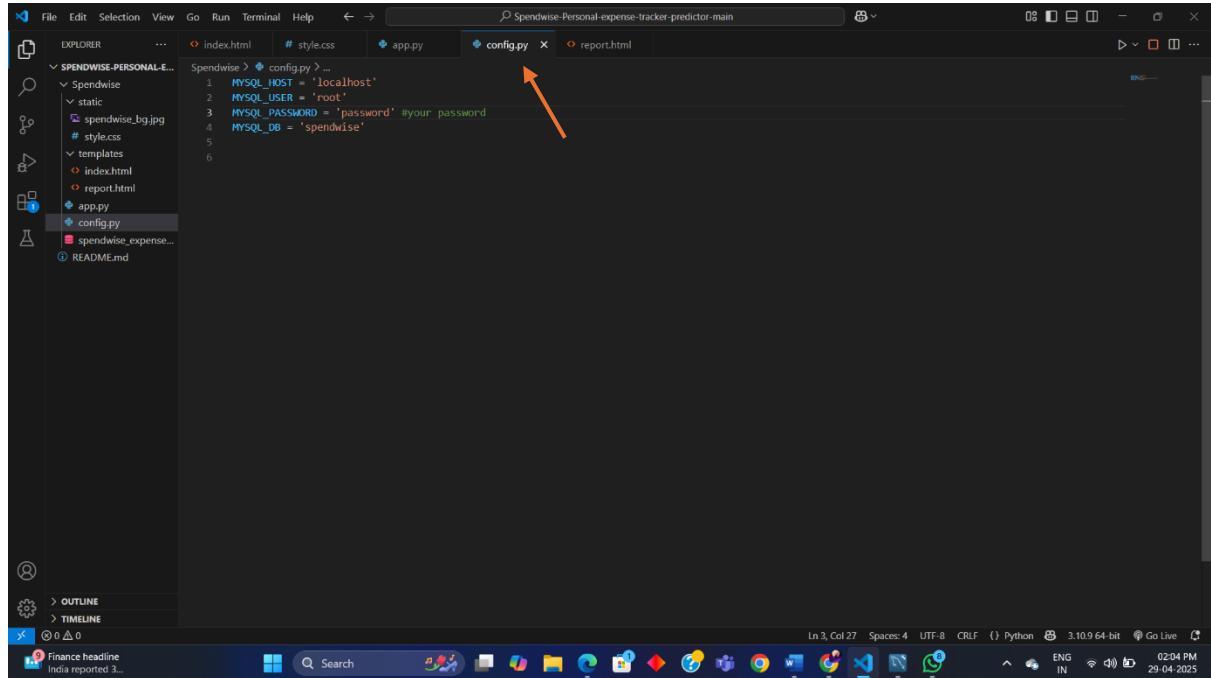
```
SPENDWISE-PERSONAL-E... Spendwise > app.py ...
```

```
1 from flask import Flask, render_template, request, jsonify
2 from flask_mysqldb import MySQL
3 import matplotlib
4 from matplotlib import Agg
5 import matplotlib.pyplot as plt
6 import io
7 import base64
8 import pandas as pd
9 import numpy as np
10 from sklearn.svm import SVR
11
12 app = Flask(__name__)
13 app.config.from_pyfile('config.py')
14 mysql = MySQL(app)
15
16 # Homepage
17 @app.route('/')
18 def index():
19     return render_template('index.html')
20
21 # Add Expense
22 @app.route('/add-expense', methods=['POST'])
23 def add_expense():
24     date = request.form['date']
25     category = request.form['category']
26     amount = float(request.form['amount'])
27
28     # Check if amount exceeds 1k limit (can be changed)
29     if amount > 1000:
30         return "Error: Amount exceeds ₹1000 limit!", 400
31
32     cursor = mysql.connection.cursor()
33     cursor.execute("INSERT INTO expenses (date, category, amount) VALUES (%s, %s, %s)", (date, category, amount))
34     mysql.connection.commit()
35     cursor.close()
36
37     return "Expense added successfully!"
```

Step 3.2: Install the python dependencies in the code editor's terminal:

pip install -r requirements.txt

Step 3.3: In config.py enter your MYSQL_PASSWORD.

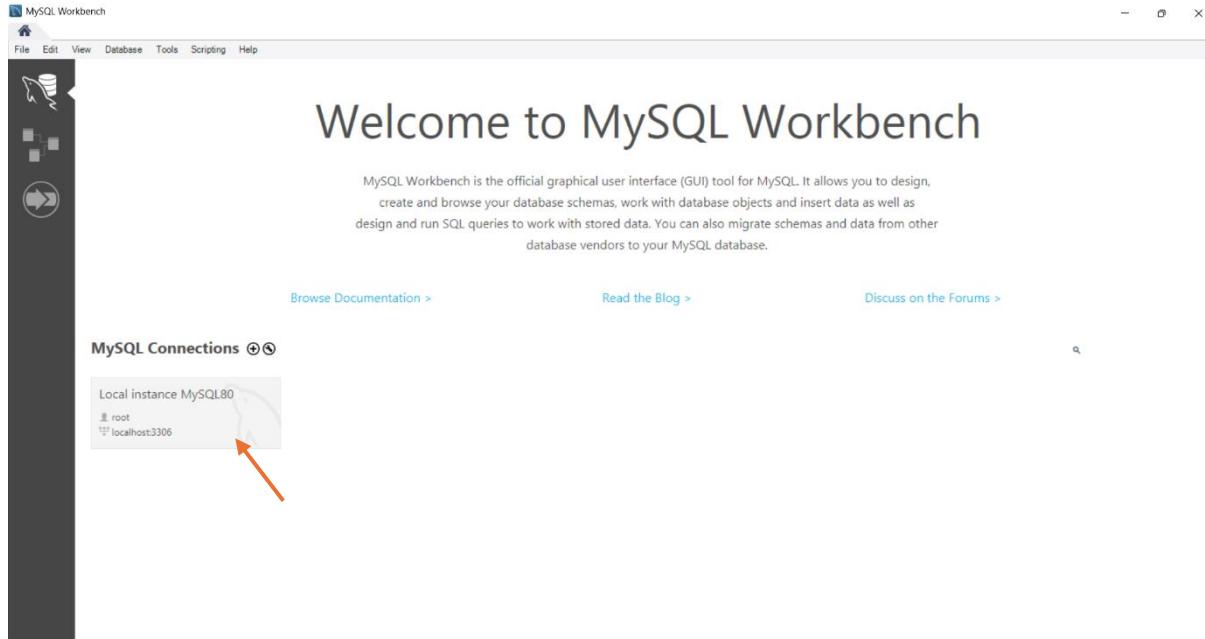


The screenshot shows the VS Code interface with the 'Spendwise-Personal-expense-tracker-predictor-main' project open. The Explorer sidebar shows files like index.html, style.css, app.py, config.py, and report.html. The code editor displays the contents of config.py, which contains MySQL connection parameters. An orange arrow points to the line 'MYSQL_PASSWORD = 'password' #your password'. The status bar at the bottom indicates the file is 3.10.9 64-bit and the current time is 02:04 PM on 29-04-2025.

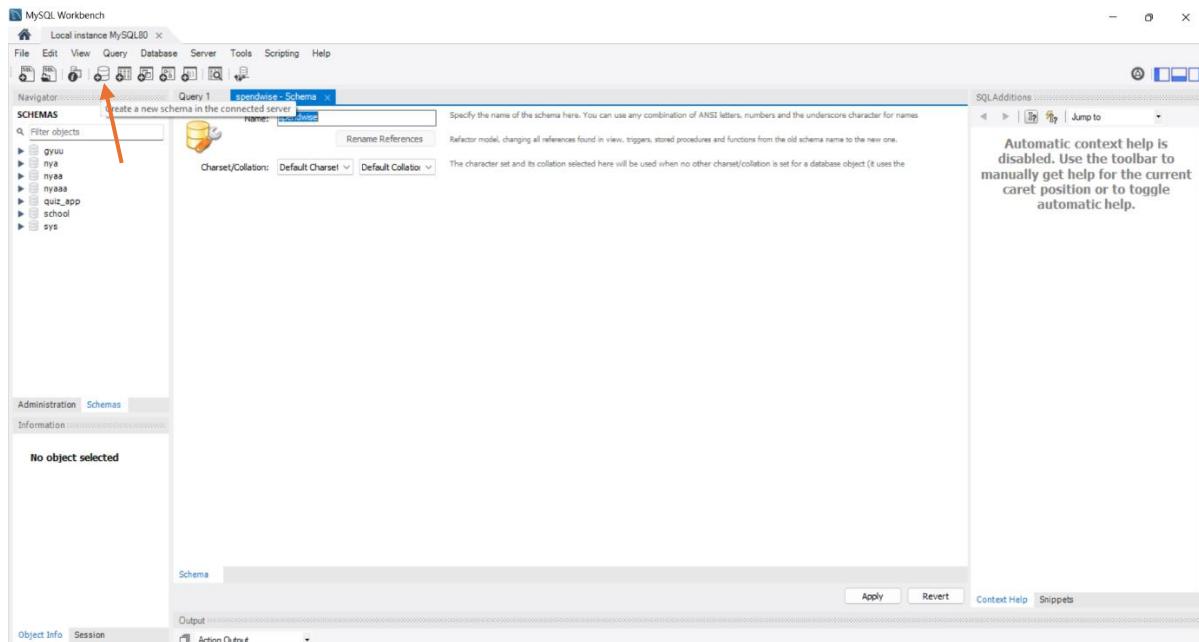
```
SPENDWISE-PERSONAL-E... config.py ...
```

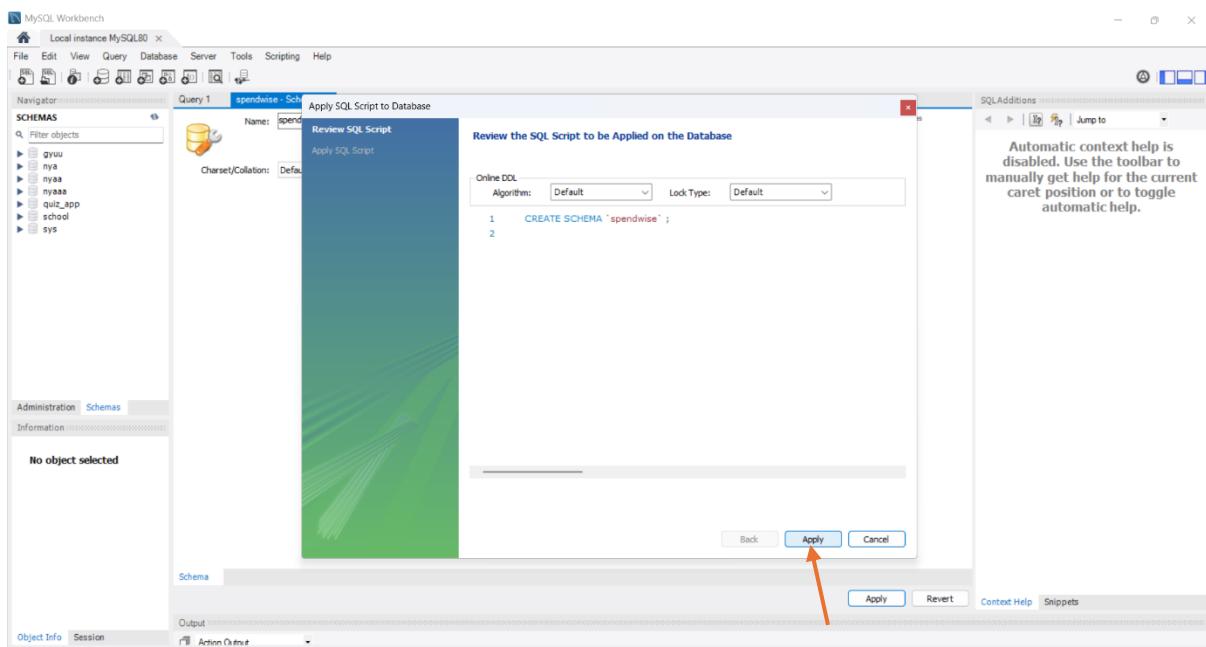
```
1 MYSQL_HOST = 'localhost'
2 MYSQL_USER = 'root'
3 MYSQL_PASSWORD = 'password' #your password
4 MYSQL_DB = 'spendwise'
5
6
```

Step 4.1: Open MySQL Workbench. Under MySQL Connections go to Local Instance.

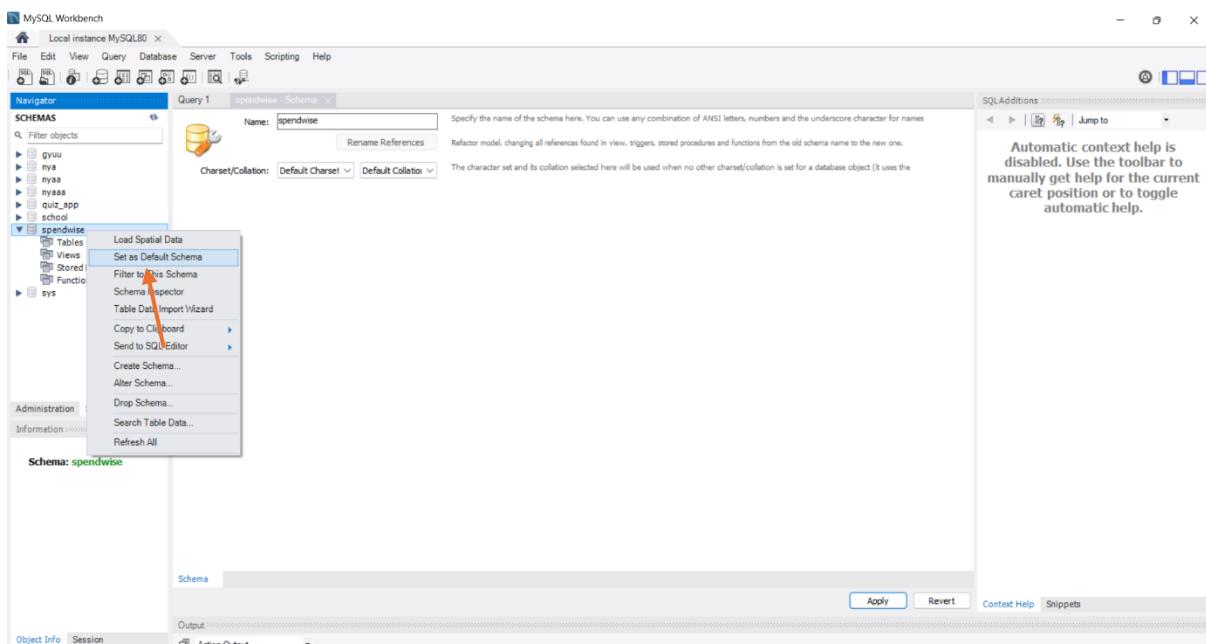


Step 4.2: Create a new database named spendwise.

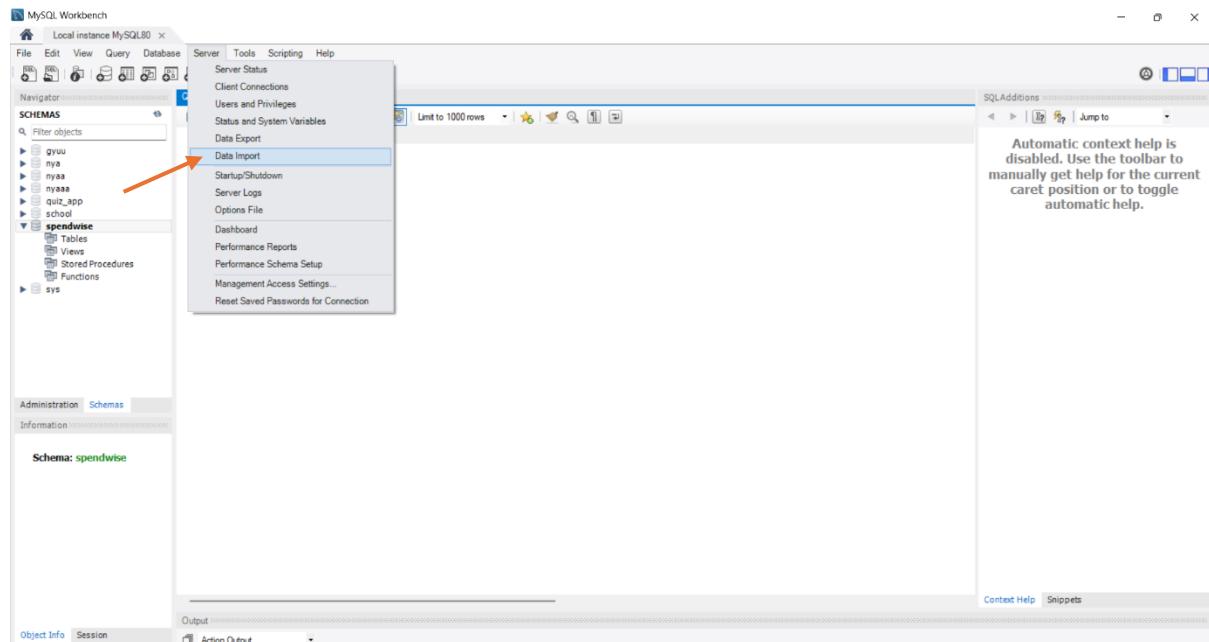




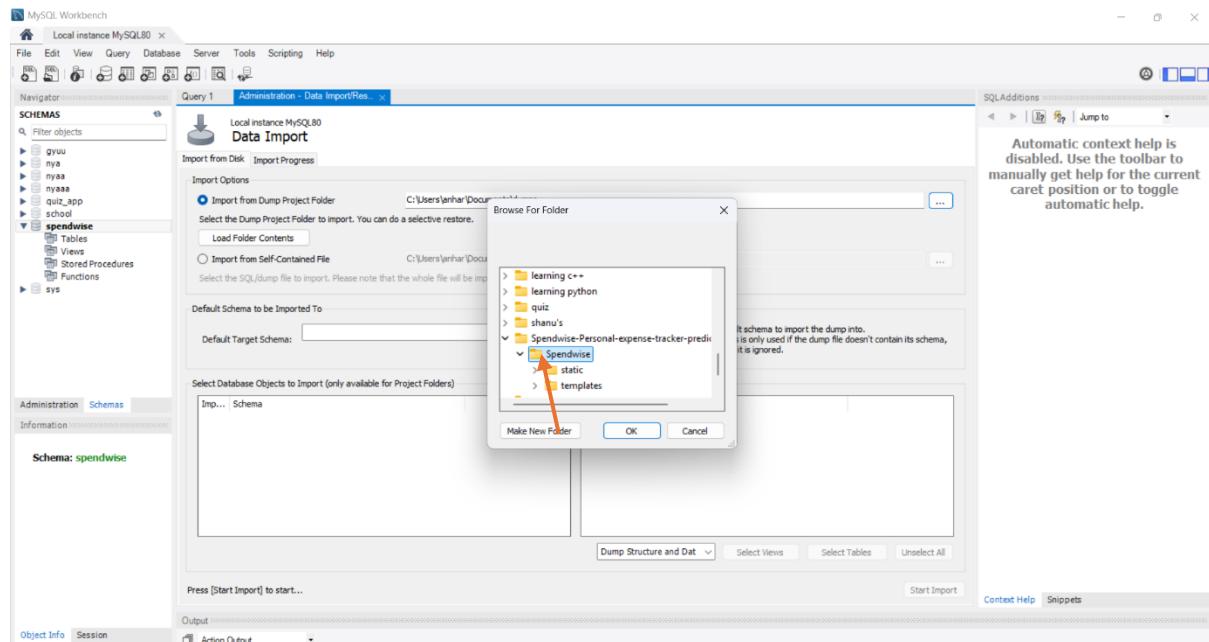
Step 4.3: Set “spendwise” as Default Schema.



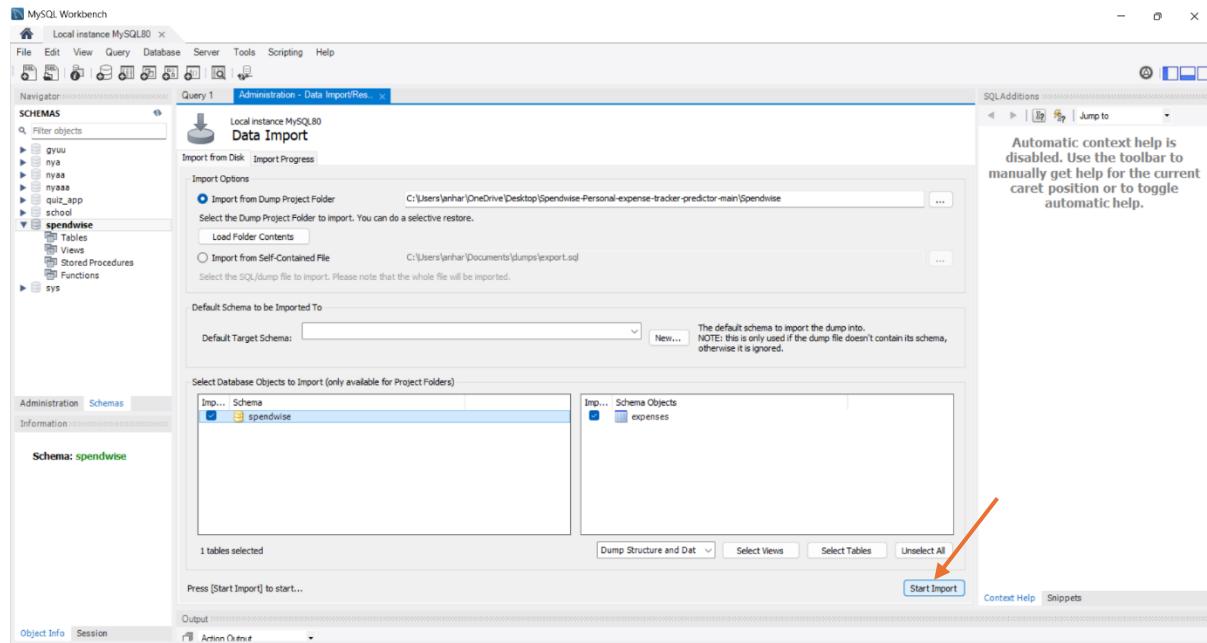
Step 4.4 : Click server and select Data Import



Step 4.5 : Select Spendwise, click “OK”.

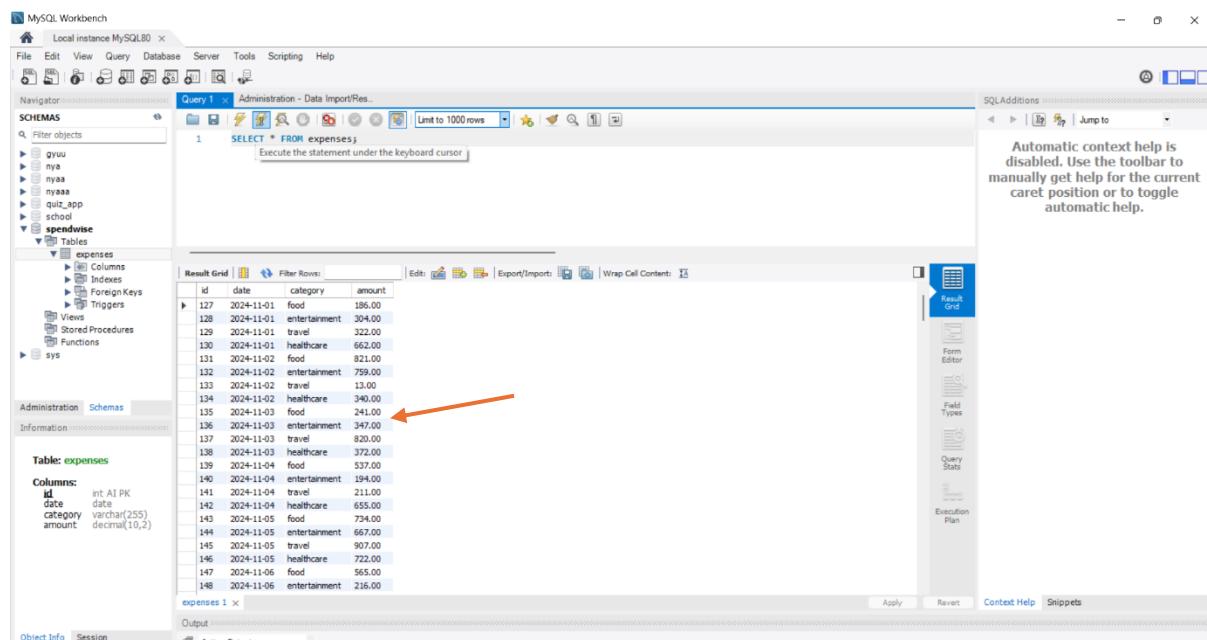


Step 4.6 : Click “Start Import”.



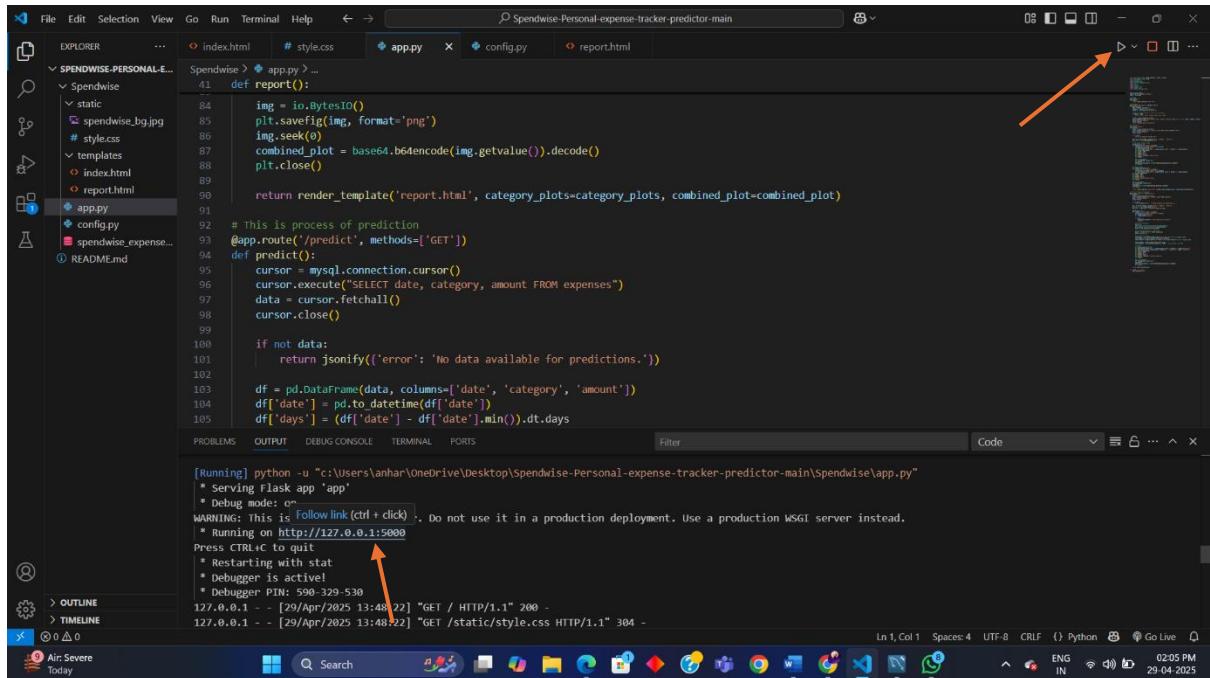
Step 4.7 : We have already provided some demo data.
Check it by running the following query :

SELECT * FROM expenses;



Step 4.8 : Minimize MySQL Workbench.

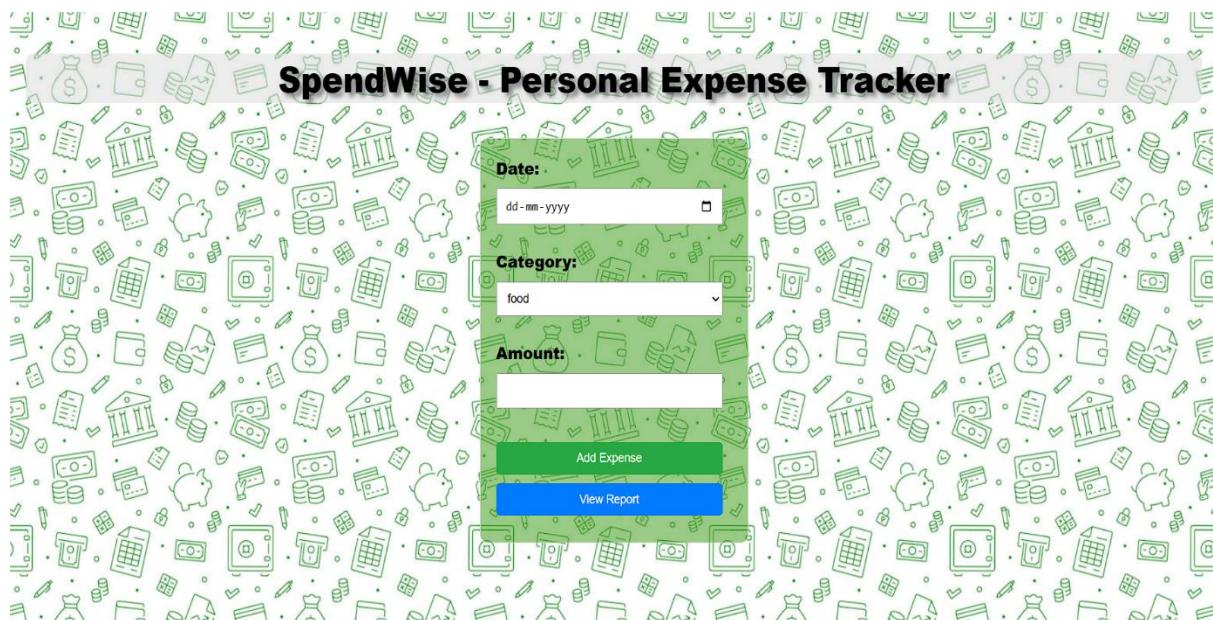
Step 5 : Run app.py in your code editor and open the link – <https://127.0.0.1:5000> – in your terminal



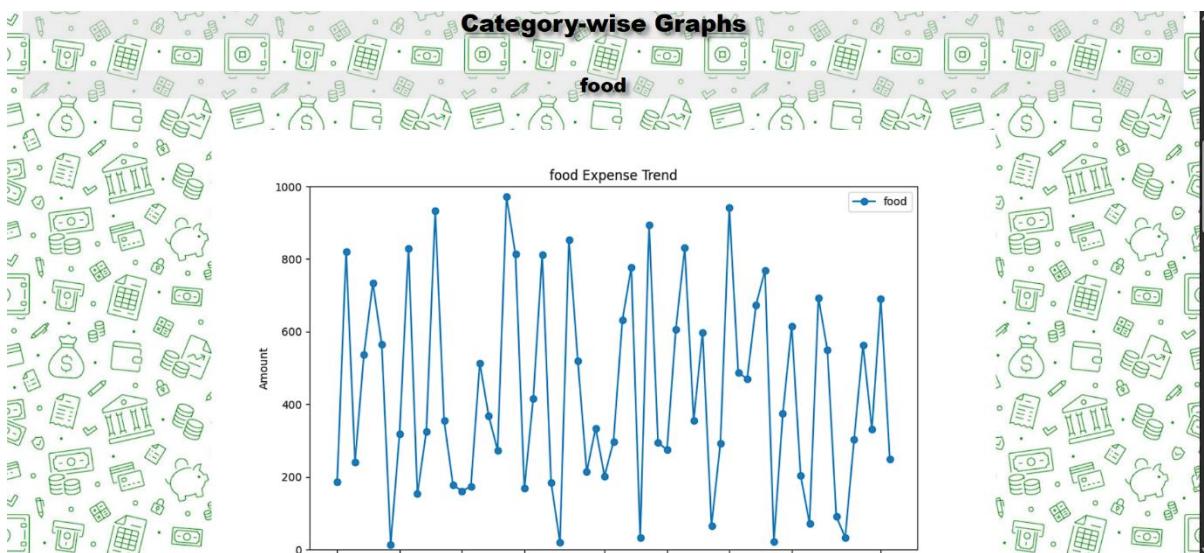
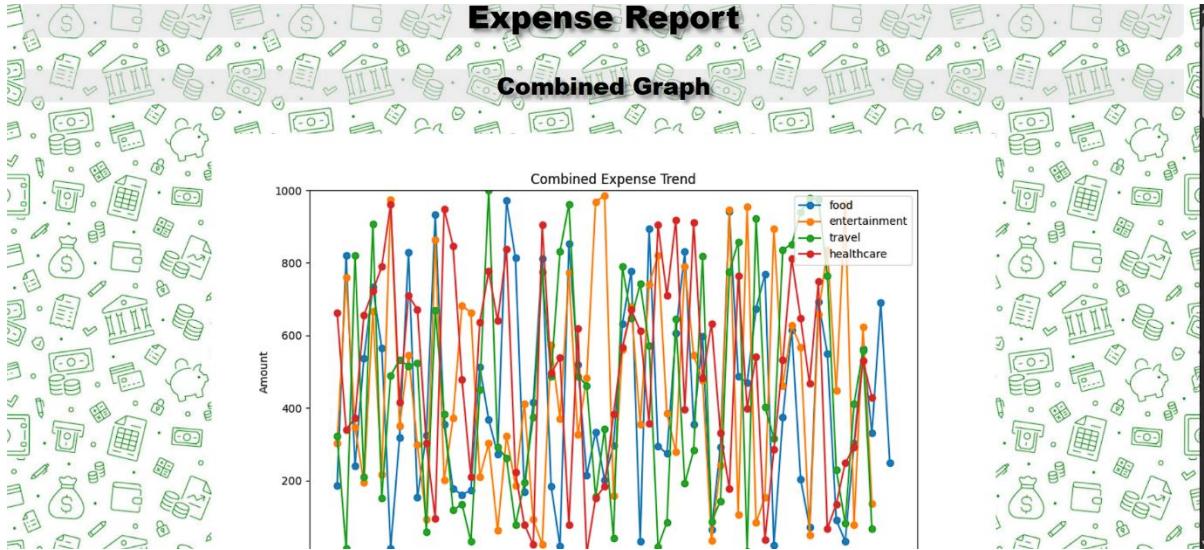
```
[Running] python -u "c:\users\anhar\onedrive\desktop\spendwise-personal-expense-tracker-predictor-main\spendwise\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is Follow link \(ctrl + click\). Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 590-329-530
127.0.0.1 - - [29/Apr/2025 13:48:22] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [29/Apr/2025 13:48:22] "GET /static/style.css HTTP/1.1" 304 -
```

Run the Project

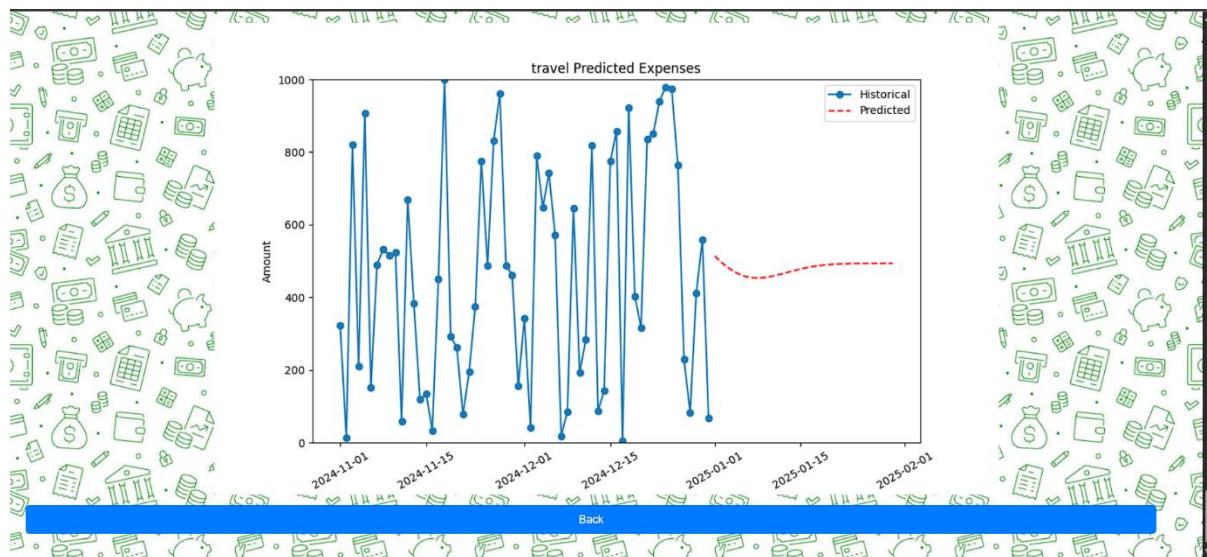
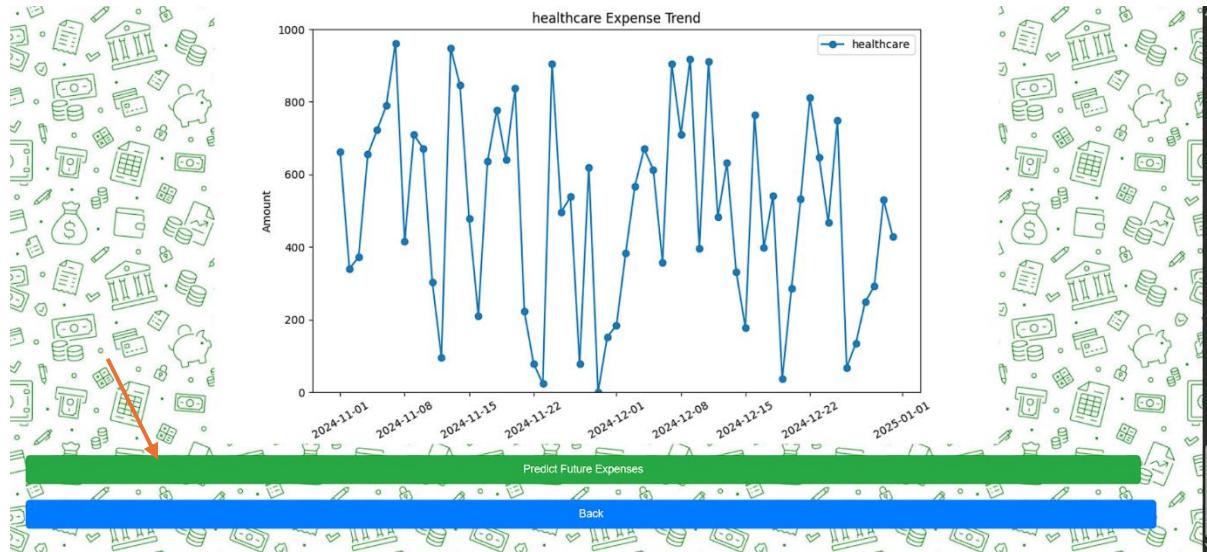
Step 1: This is the home page of our website. Fill the form and click “Add Expense” button to store your data. After which you can click on “View Report” button to view your reports.



Step 2: This is the report page of our website. Here you can access your combined report of expenditures also including the individual reports.



Step 3: Click the “Predict Future Expenses” button to generate the predictions up to the next month. The red dotted line indicates the predicted expenditures.



Step 4: Manage your expenses with the help of predicted budget of the upcoming month and keep a track of your expenditures on a daily basis using our website 😊