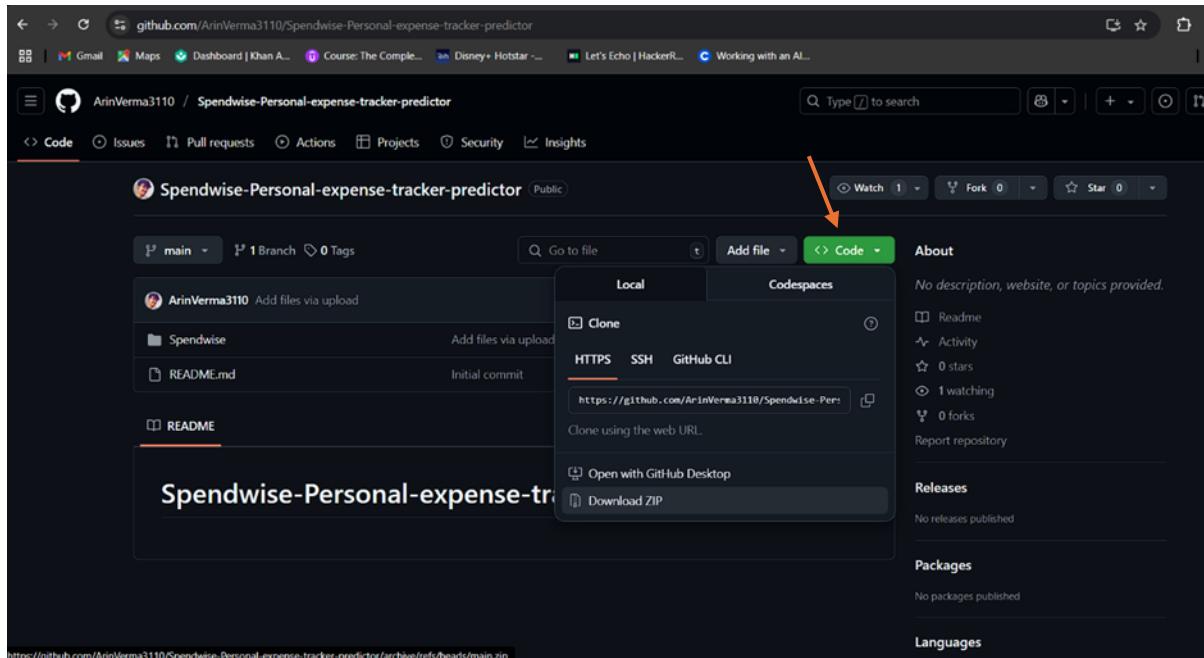


Steps to Set-up

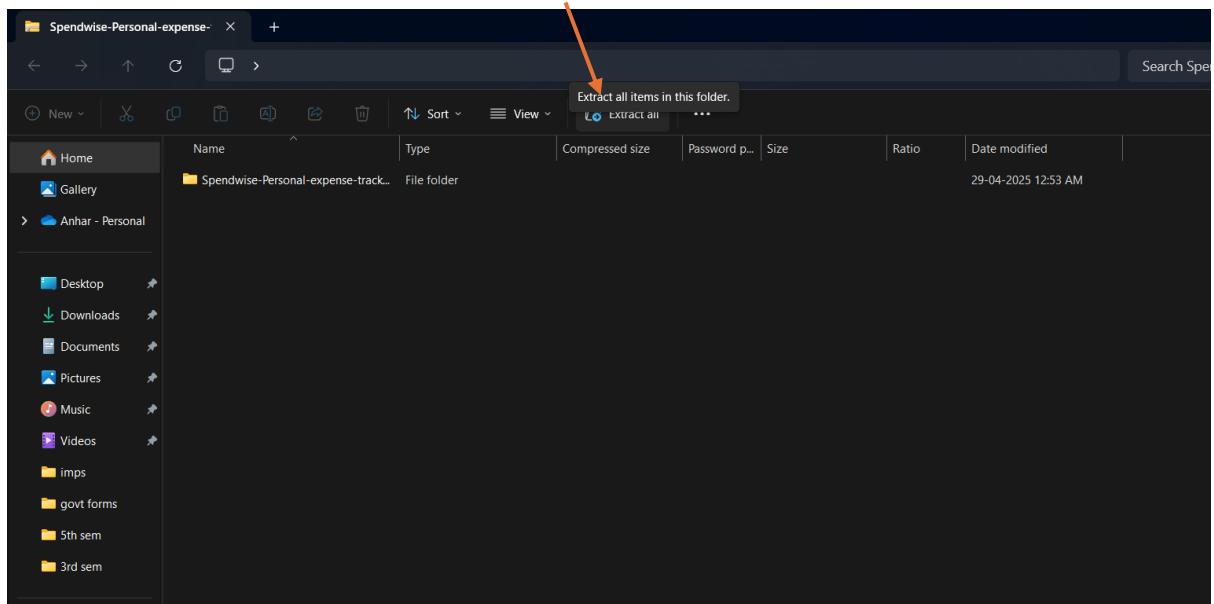
Step 1: Clone the Repository by downloading the zip file or write this command in your bash :

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

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

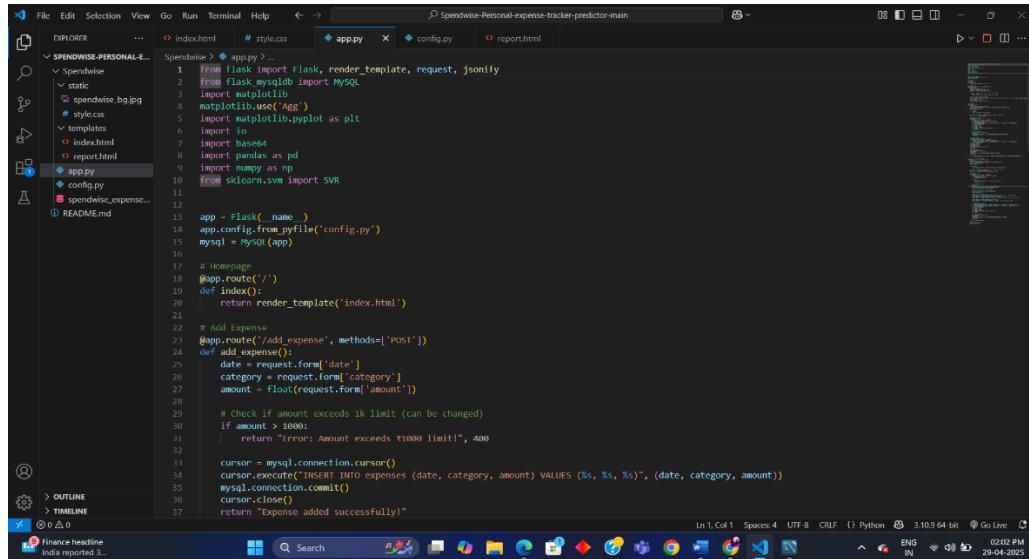


Step 2: Extract the Zip file.



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 project open. The Explorer sidebar on the left lists files and folders: index.html, style.css, app.py, config.py, report.html, README.md, and others. The code editor tab for 'app.py' is active, displaying Python code for a Flask application. The code includes imports for Flask, MySQLdb, and various plotting libraries. It defines routes for the homepage and an expense addition endpoint, which inserts data into a MySQL database. The status bar at the bottom shows the file is 3.10.9 64-bit, Python 3.10.9 64-bit, and the date is 29-04-2025.

```
from flask import Flask, render_template, request, jsonify
from flask_mysqldb import MySQL
import matplotlib
matplotlib.use('Agg')
import io
import base64
import pandas as pd
import numpy as np
from sklearn.svm import SVR

app = Flask(__name__)
app.config.from_pyfile('config.py')
mysql = MySQL(app)

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

@app.route('/add-expense', methods=['POST'])
def add_expense():
    date = request.form['date']
    category = request.form['category']
    amount = float(request.form['amount'])

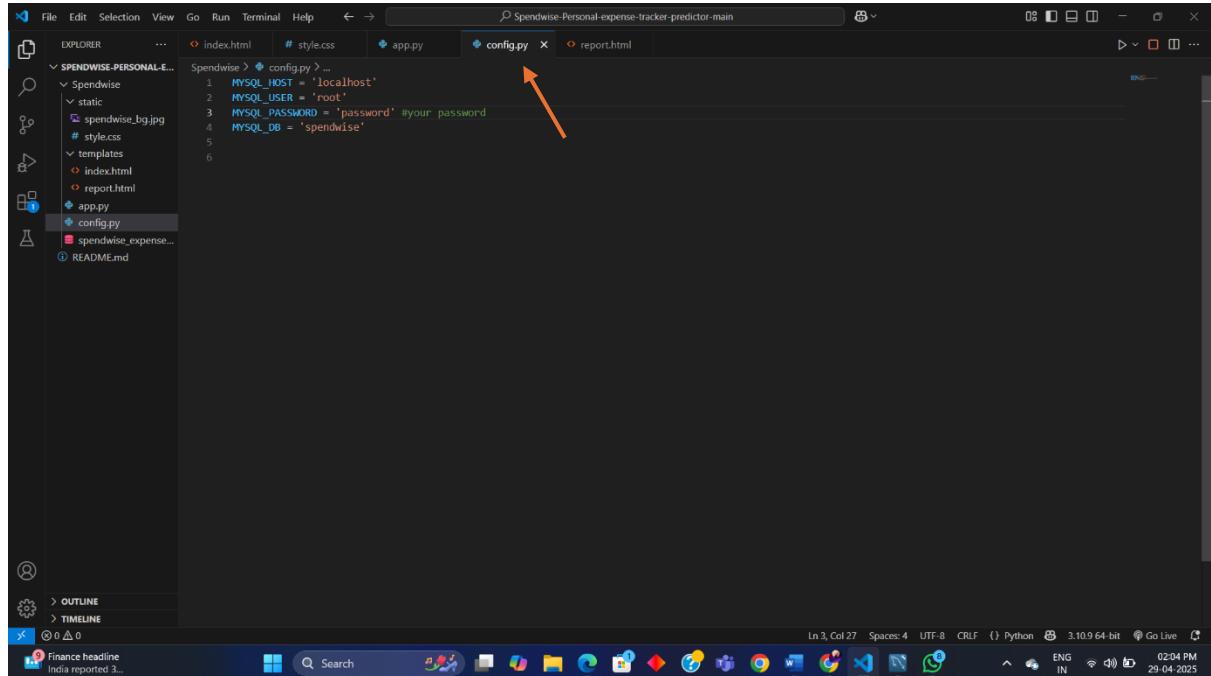
    # Check if amount exceeds 1k limit (can be changed)
    if amount > 1000:
        return "Error: Amount exceeds ₹1000 limit!", 400

    cursor = mysql.connection.cursor()
    cursor.execute("INSERT INTO expenses (date, category, amount) VALUES (%s, %s, %s)", (date, category, amount))
    mysql.connection.commit()
    cursor.close()
    return "Expense added successfully!"
```

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

pip install -r requirements.txt

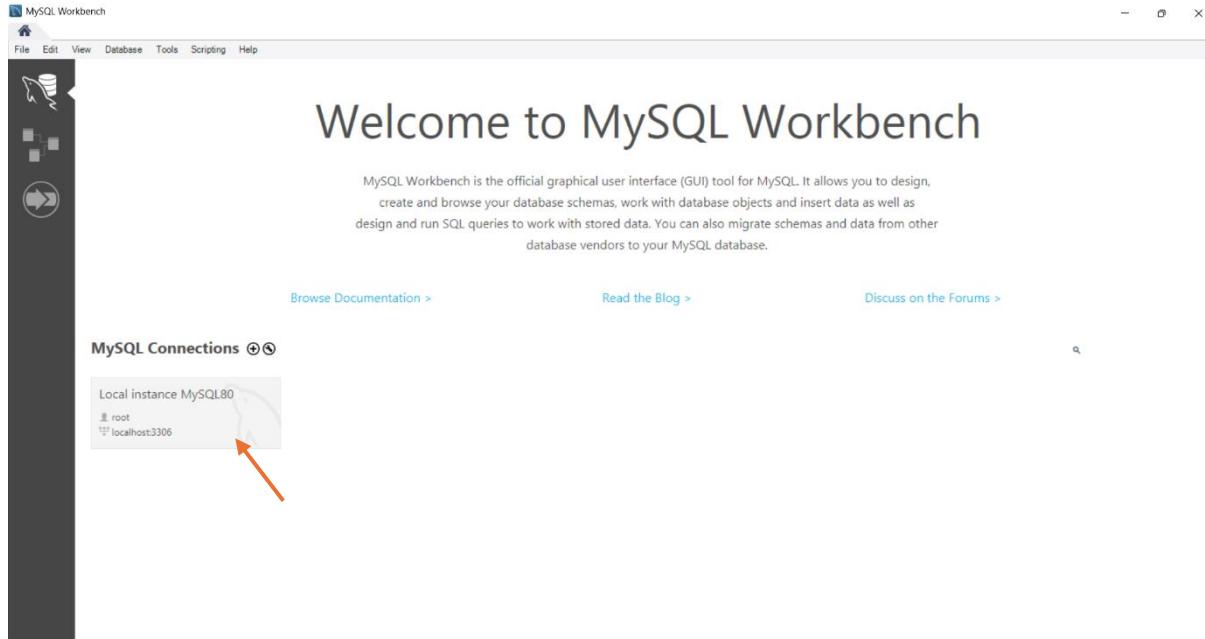
Step 3.3: In config.py enter your MYSQL_PASSWORD.



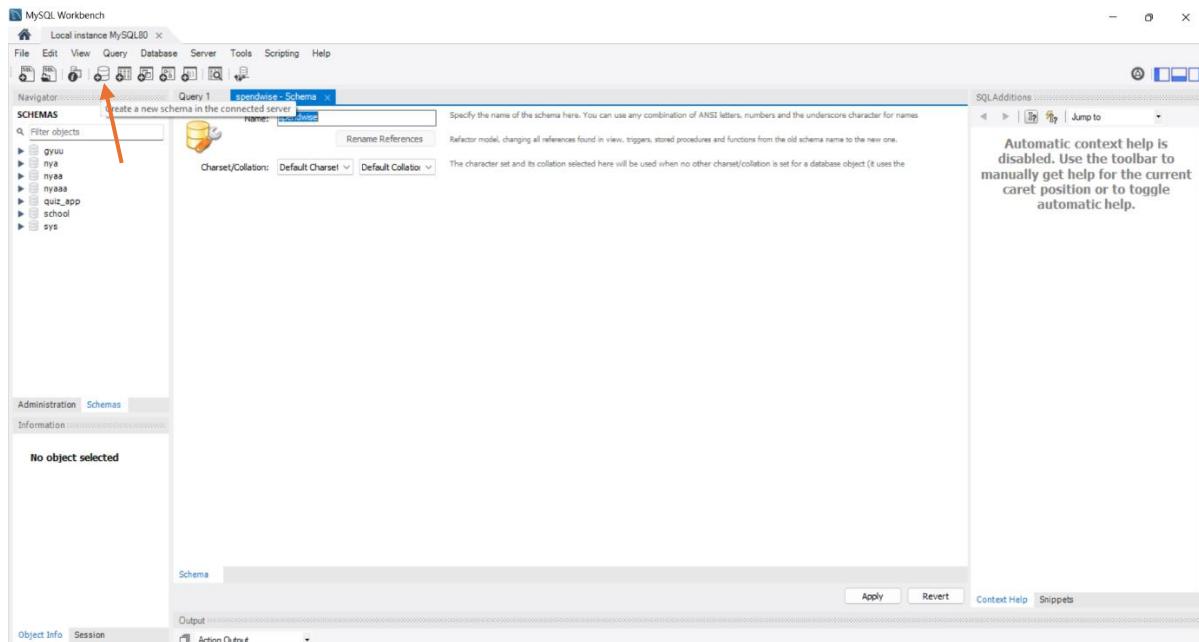
The screenshot shows the VS Code interface with the SpendWise project open. The Explorer sidebar on the left lists files and folders: index.html, style.css, app.py, config.py, report.html, README.md, and others. The code editor tab for 'config.py' is active, displaying Python code that defines MySQL connection parameters. An orange arrow points to the line 'MYSQL_PASSWORD = 'password' #your password'. The status bar at the bottom shows the file is 3.10.9 64-bit, Python 3.10.9 64-bit, and the date is 29-04-2025.

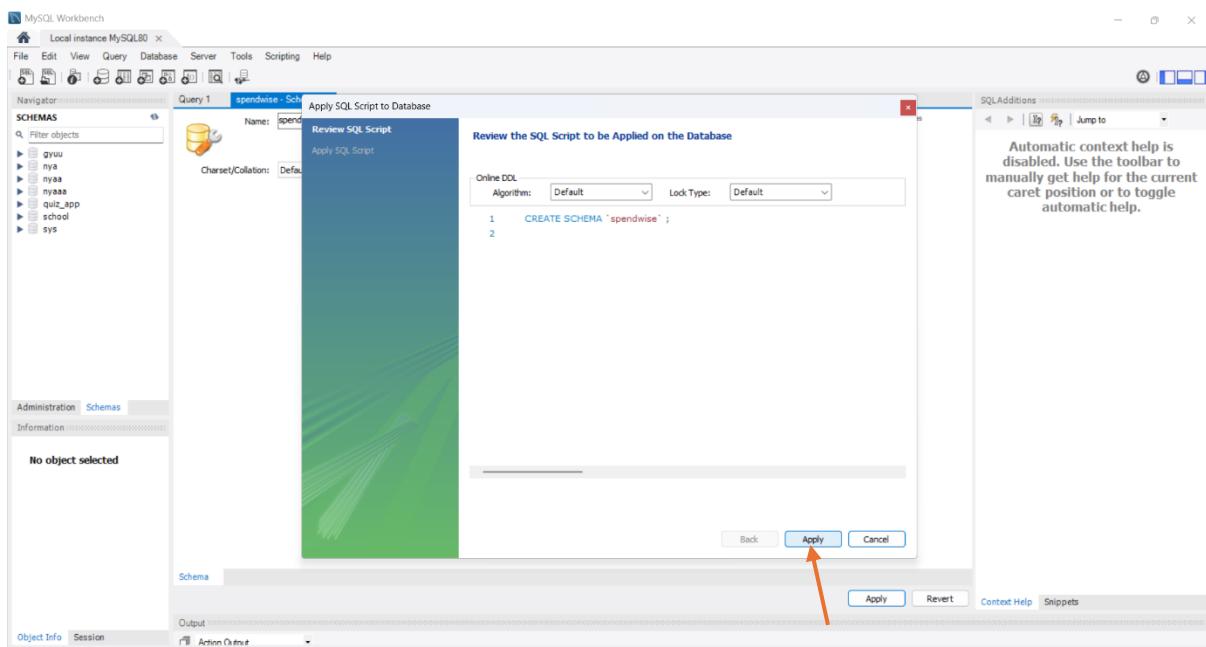
```
MYSQL_HOST = 'localhost'
MYSQL_USER = 'root'
MYSQL_PASSWORD = 'password' #your password
MYSQL_DB = 'spendwise'
```

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

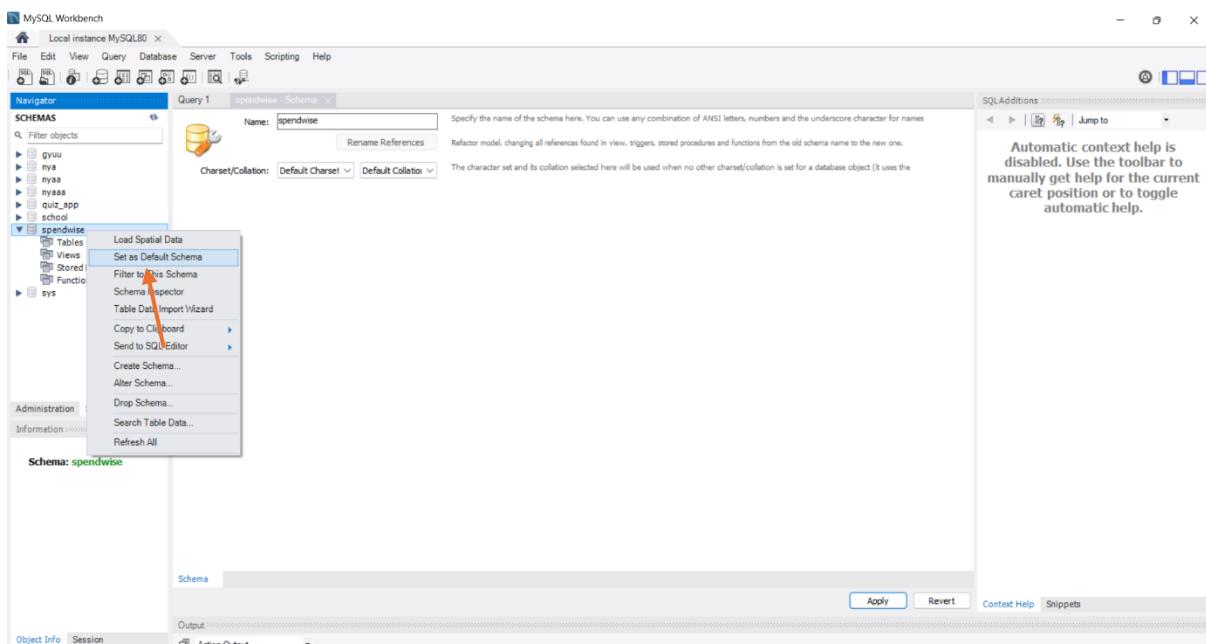


Step 4.2: Create a new database named spendwise.

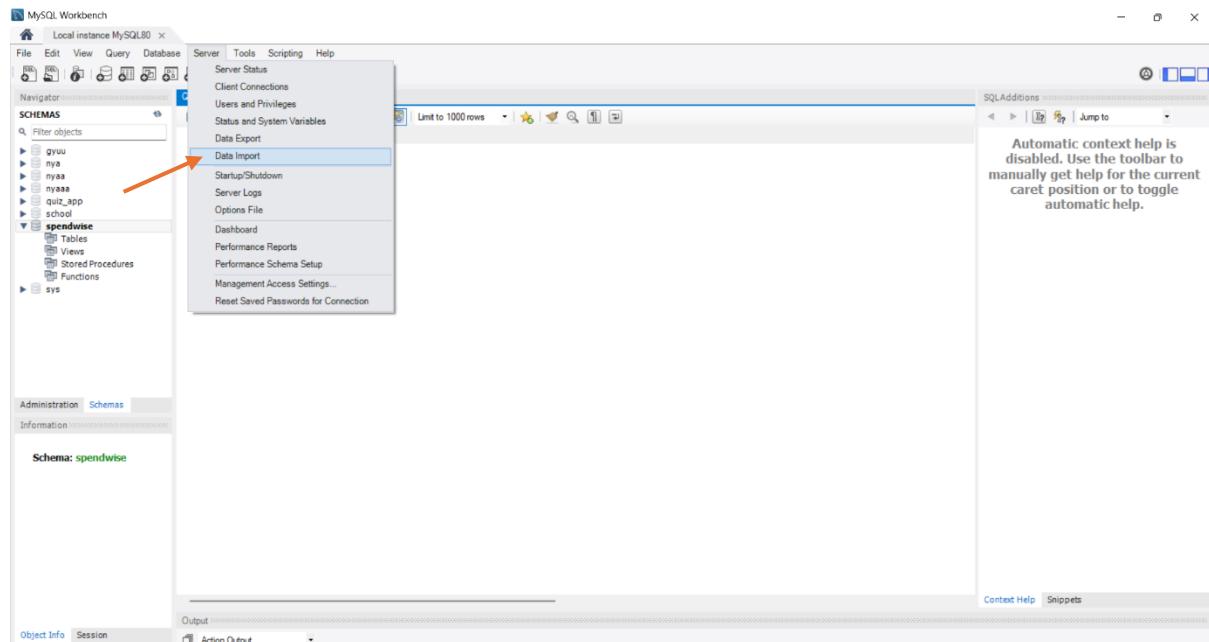




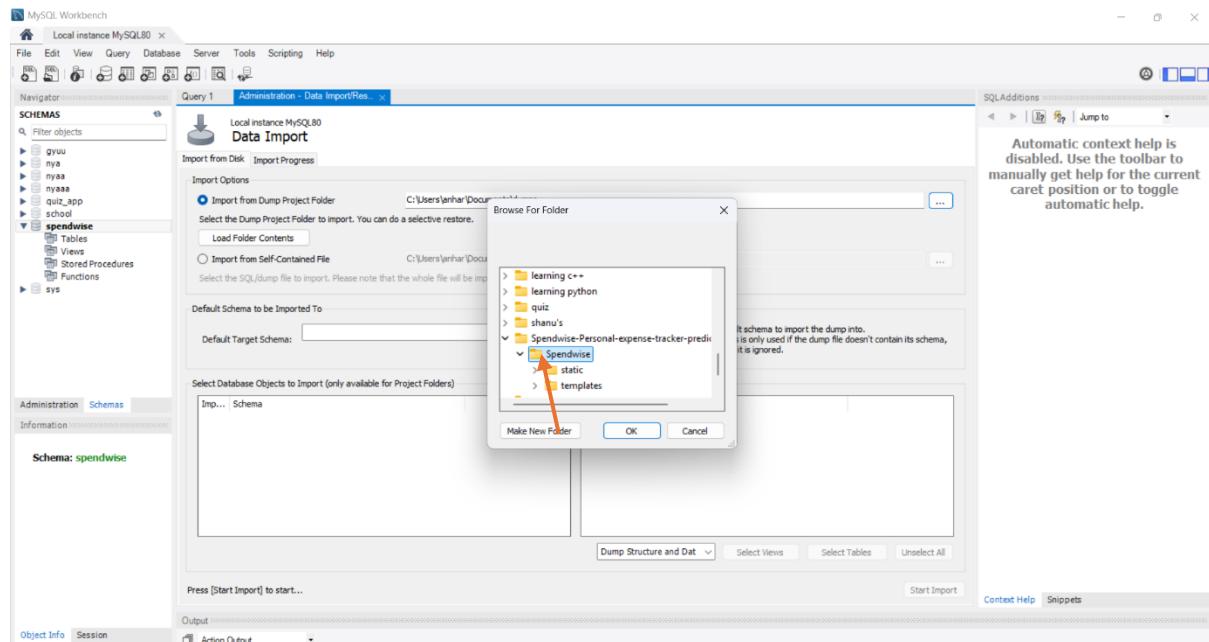
Step 4.3: Set “spendwise” as Default Schema.



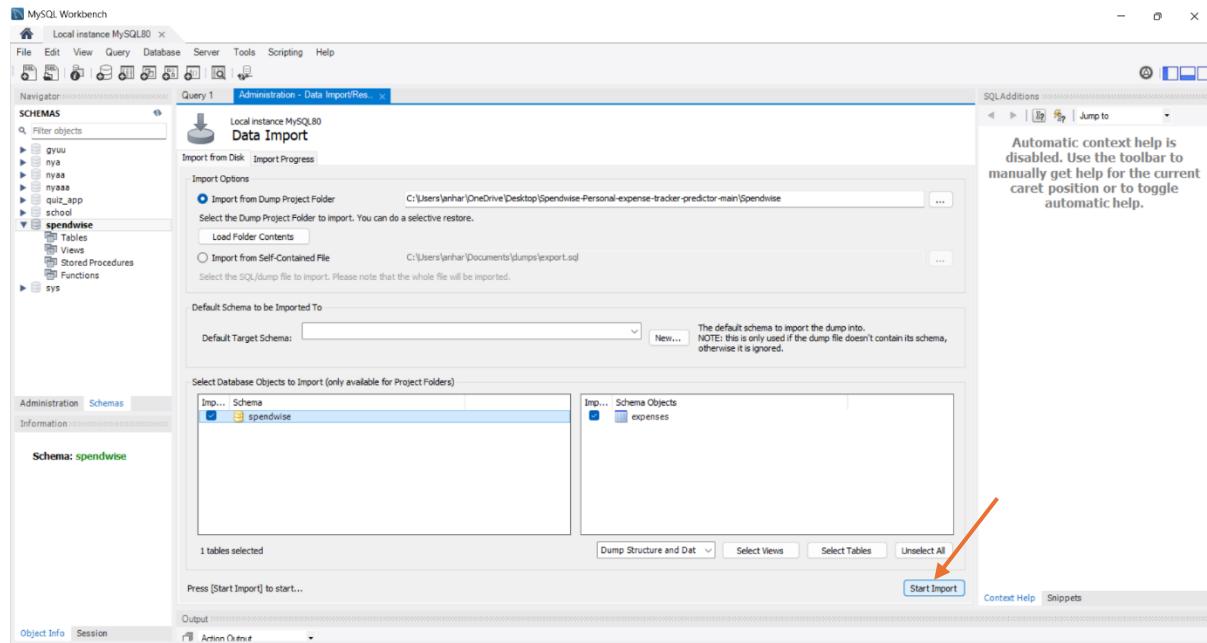
Step 4.4 : Click server and select Data Import



Step 4.5 : Select Spendwise, click “OK”.

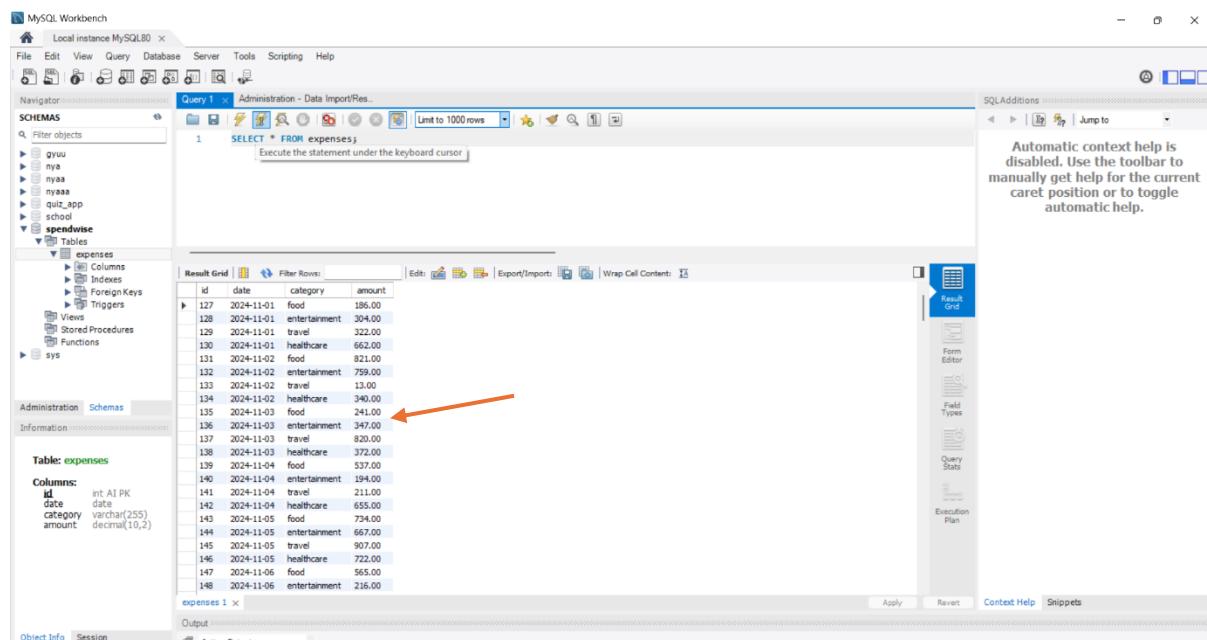


Step 4.6 : Click “Start Import”.



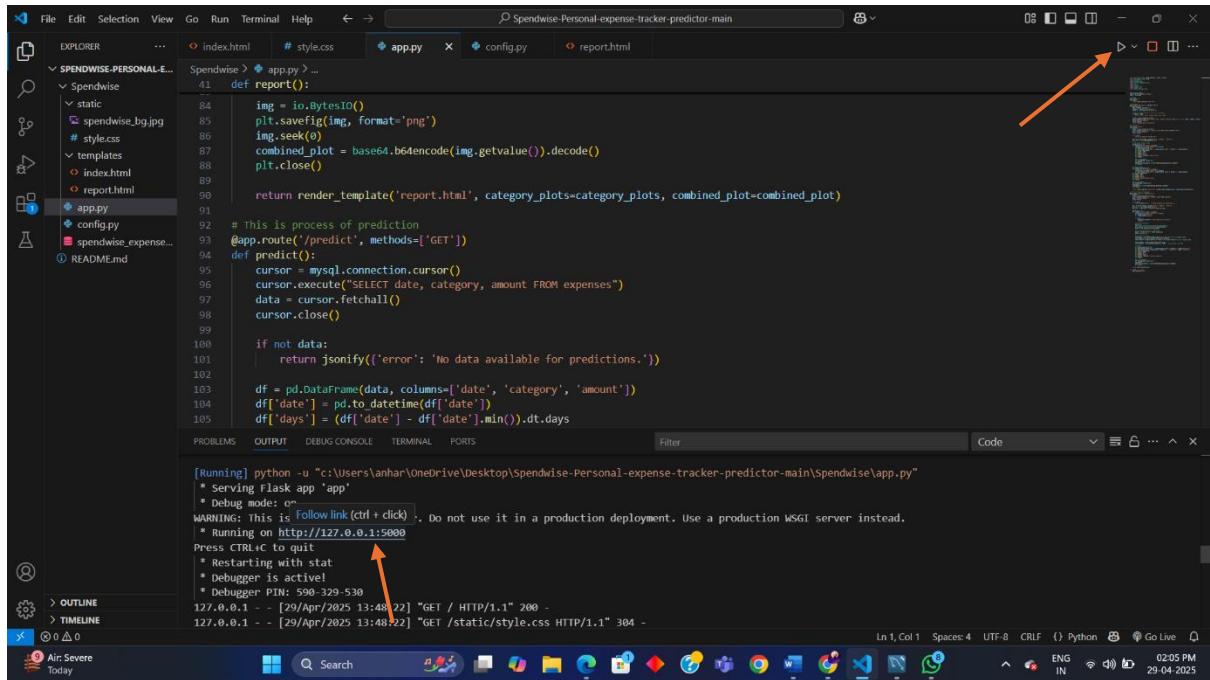
Step 4.7 : We have already provided some demo data.
Check it by running the following query :

SELECT * FROM expenses;



Step 4.8 : Minimize MySQL Workbench.

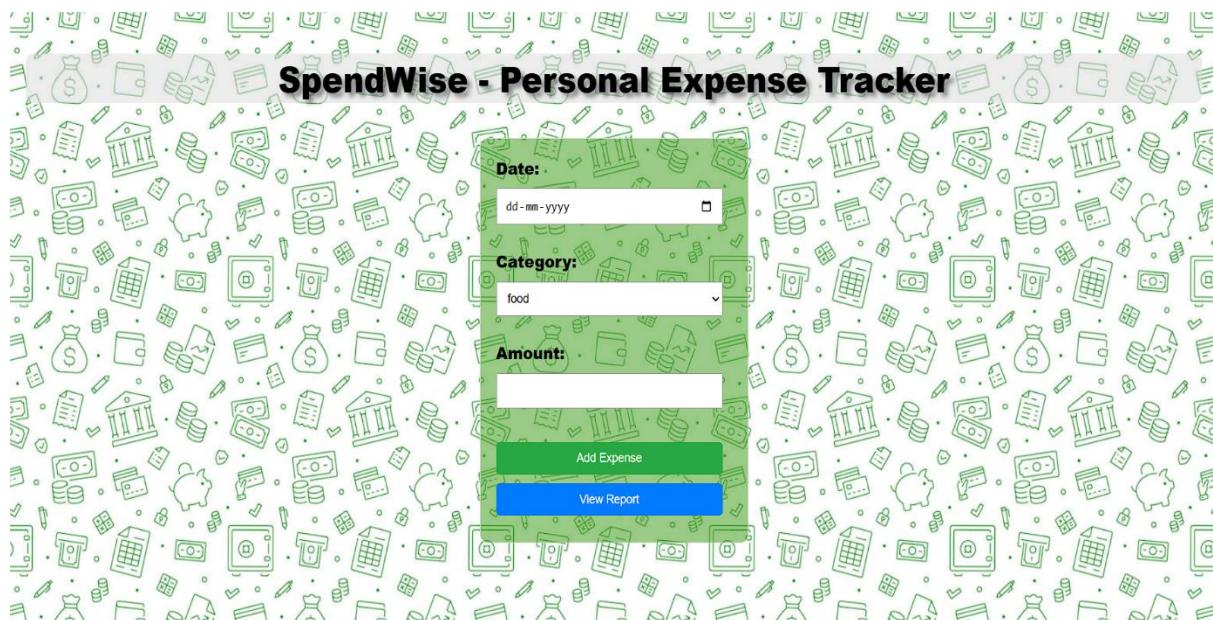
Step 5 : Run app.py in your code editor and open the link – <https://127.0.0.1:5000> – in your terminal



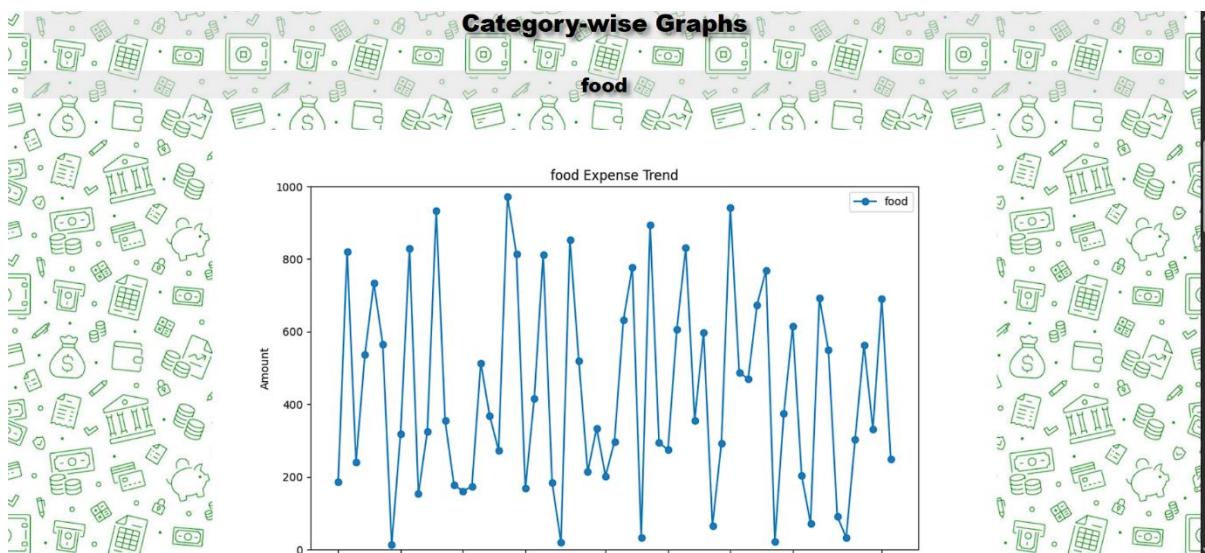
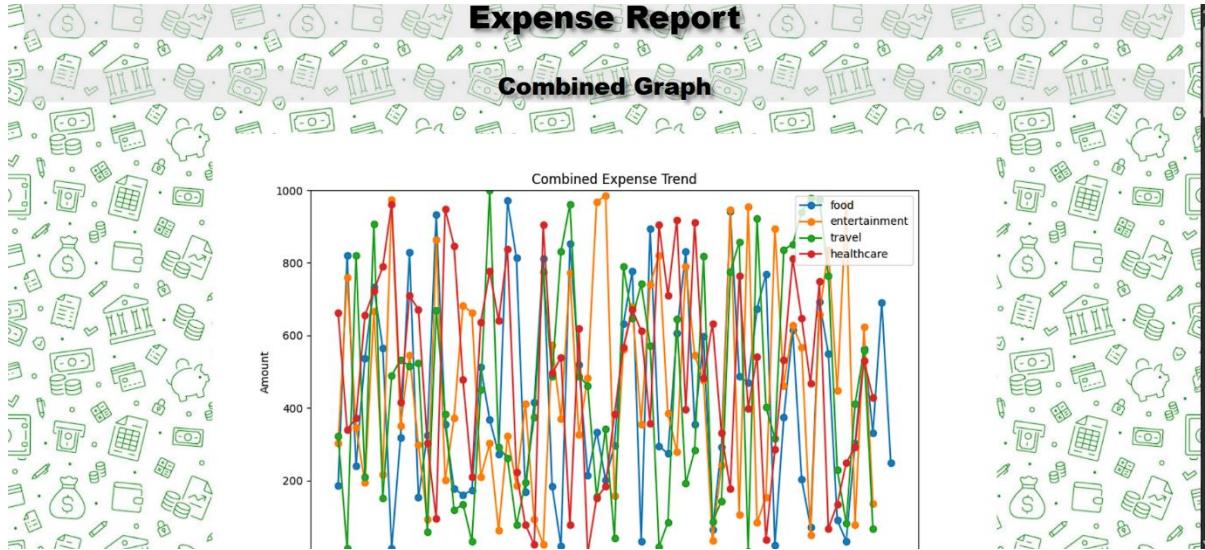
```
[Running] python -u "c:\users\anhar\onedrive\desktop\spendwise-personal-expense-tracker-predictor-main\spendwise\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is Follow link \(ctrl + click\). Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 598-329-530
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

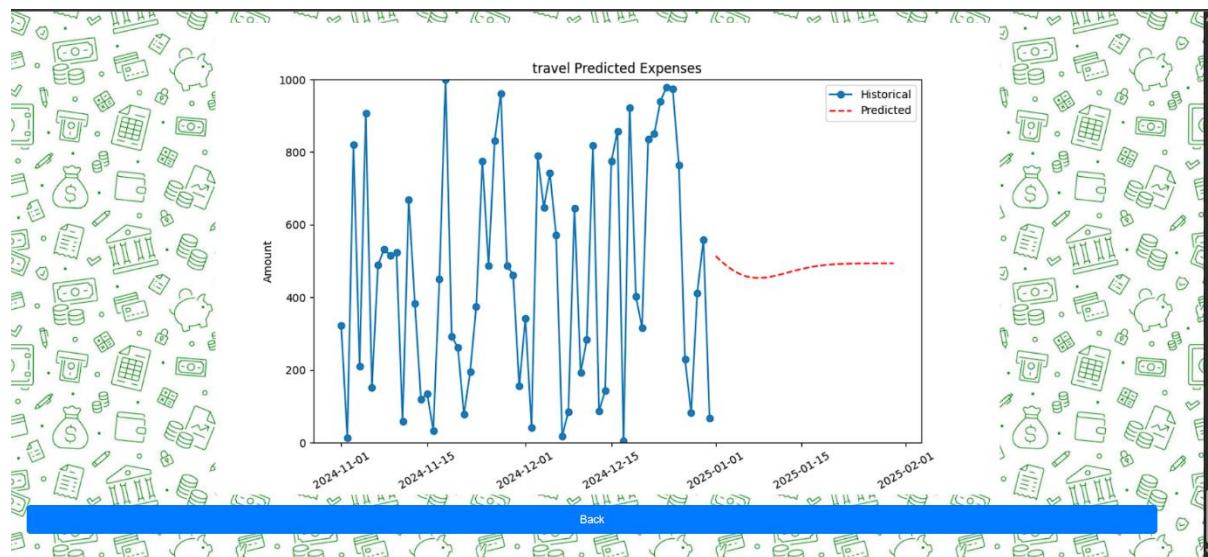
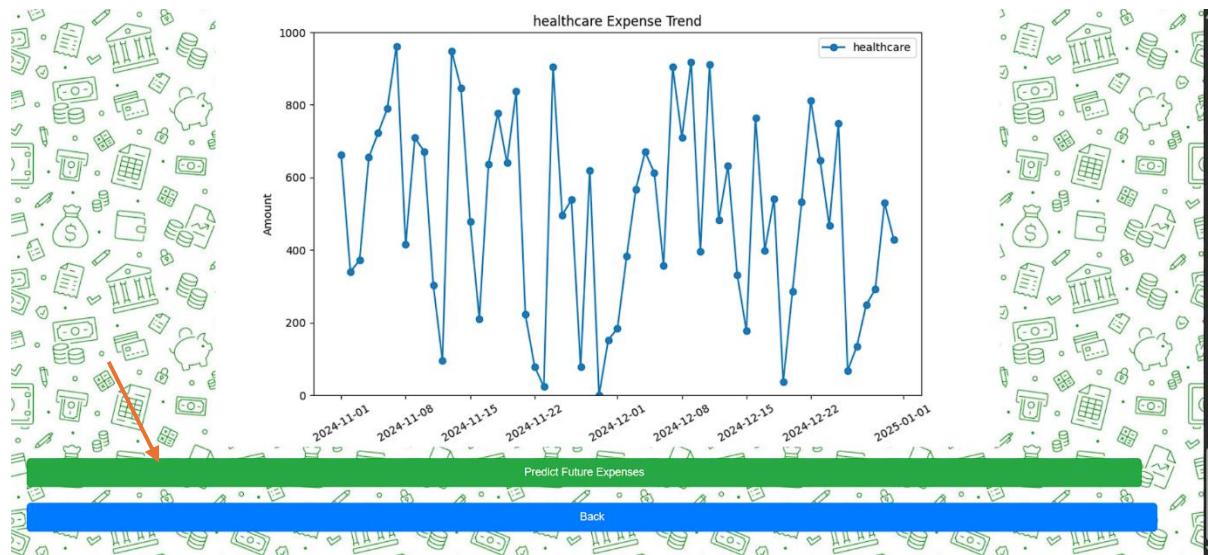
Step 1: This is the home page of our website. Fill the form and click “Add Expense” button to store your data. After which you can click on “View Report” button to view your reports.



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 😊