A
PROJECT REPORT ON

# EXPENSIFY - INCOME EXPENSE TRACKER

**By**

**CHAVDA DHARMI (CE-24) (19CEUBS127)**
**HATVANI MITTAL (CE-82) (19CEUBS126)**

**B.Tech CE Semester-VI**
**Subject: System Design Practice**

**Guided by:**
Prof.Pandav K. Patel
Assistant Professor
Dept. of Comp. Engg.

**Faculty of Technology**
**Department of Computer Engineering**
**Dharmsinh Desai University**

**Faculty of Technology**
**Department of Computer Engineering**
**Dharmsinh Desai University**

## CERTIFICATE

This is to certify that the practical / term work carried out in the subject of

**System Design Practice** and recorded in this journal is the

bonafide work of

**CHAVDA DHARMI (CE-24) (19CEUBS127)**
**HATVANI MITTAL (CE-82) (19CEUBS126)**

of B.Tech semester **VI** in the branch of **Computer Engineering**

during the academic year **2021-2022**.

| | |
|---|---|
| Prof. Pandav K. Patel | Dr. C. K. Bhensdadia, |
| Assistant Professor, | Head, |
| Dept. of Computer Engg., | Dept. of Computer Engg., |
| Faculty of Technology | Faculty of Technology |
| Dharmsinh Desai University, Nadiad | Dharmsinh Desai University, Nadiad |

# Contents

## 1)Abstract

Expensify is a website for expense management. It is designed for keeping track of their expenditure with easy and effective way through computerized system. It tends to eliminate manual paper works. It is used to identify and eliminate wasteful spending habits in financial life. It will help to maintain control of your finances, and promote better financial habits like saving and investing.

# 1) Introduction

Expensify is basically a website that records the amount of income and expense for end user.

Firstly, the registered user can successfully login to the system. Ans then user can enter the records of the daily expenses and income. Information entered can further be edited or deleted permanently. All the records would be stored according to date and would be visible in history. Further user can view their statistics by graph according to month which he/she selected changes reflected in graph. Further they can download excel file of their added records.

## Technologies/Tools used

### Technologies:

- Django
- Bootstrap
- Chart Js
- HTML
- CSS

### Tools:

- Git
- Visual studio code
- Chrome Browser

# 3)Software Requirement Specifications

## 1)Manage User

**R.1.1: Profile**

***Description:*** This section includes information related to user's profile.

**R.1.1.1: View profile**

***Description:*** This option shows the profile of the user. It displays their user id, username and email id.

***Input:*** Click on the view profile option.

***Output:*** Profile screen would be displayed.

**R.1.1.2: Edit profile**

***Description:*** This option edits the profile of the User.

***Input:*** Click on the edit profile button .

## 2) Manage registration/login

### R.2.1: End User

**R.2.1.1: Registration**

***Description:*** If User doesn't have any exiting account then they have to register themselves.

***Input:*** User have to provide their name, email, and password.

***Output:*** User would be redirected to login page.

**R.2.1.2: Login**

***Description*:** If User already have an account then this option will be used to display home page by logging in.

***Input:*** Operator have to give their username and password.

## 3) Manage Expenses

**R.3.1: Add expense category**

***Description:*** This function will allow user to add the expense category name.

***Input:*** Click on the Add category.

***Output:*** Add category form would be displayed.

**R.3.2: Add expense record**

***Description:*** This function will allow user to add the expense information.

***Input:*** Click on the Add Expense.

***Output:*** Add record form would be displayed.

**R.3.3: View expenses record**

***Description:*** This function will show the list of all the expenses records along with date in tabular formate.

***Input:*** Click on the Dashboard.

***Output:*** List of all the entered expenses would be visible.

**R.3.4: Edit expense record**

***Description:*** This function will let user to edit any expense record and save changes to the database.

***Input:*** Click on buy edit record button.

***Output:*** edit form would be visible where user can edit and save changes to database.

**R.3.5 : Delete expense record**

***Description:*** This function will let operator to delete any expense record.

***Input:*** Click on delete record button.

***Output:*** Expense record would be deleted from the list.

**R.3.6 : Search expense record**

***Description:*** This function will let user search all the recorded expense with respect to amount, category, date and description.

***Input:*** Type text in the search bar.

***Output:*** Matched expense records would be visible accordingly.

**R.3.7 : View expense summary**

***Description:*** This function will let user to view history of all the entered expenses in different types of charts based on months we selected.

***Input:*** Click on Expense summary button from sidebar.

***Output:*** Expenses records summary would be visible.

**R.3.8: Export expense**

***Description:*** This function will let user download all expenses which recorded in database in excel sheet on their system.

***Input:*** Click on delete record button.

***Output:*** Logged in user's expense records would be downloaded.

## <u>4)</u> <u>Manage Income</u>

**R.4.1: Add Income source**

***Description:*** This function will allow user to add the income source.

***Input:*** Click on the Add source.

***Output:*** Add source form would be displayed

**R.4.2: Add Income record**

***Description:*** This function will allow user to add the income information.

***Input:*** Click on the Add Income.

***Output:*** Add record form would be displayed.

**R.4.3 : View income record**

***Description:*** This function will show the list of all the income records along with date in tabular formate.

***Input:*** Click on the Dashboard.
***Output:*** List of all the entered income would be visible.

### R.4.4: Edit income record

***Description:*** This function will let user to edit any income record and save changes to the database.

***Input:*** Click on buy edit record button.

***Output:*** edit form would be visible where user can edit and save changes to database.

### R.4.5: Delete income record

***Description:*** This function will let operator to delete any income record.

***Input:*** Click on delete record button.

***Output:*** Income record would be deleted from the list.

### R.4.6 : Search income

***Description:*** This function will let user search all the recorded income with respect to amount, category, date and description.

***Input:*** Type text in the search bar.

***Output:*** Matched income records would be visible accordingly.

**R.4.7 : View income summary**

***Description:*** This function will let user to view history of all the entered income in different types of charts based on months we selected.

***Input:*** Click on Income summary button from sidebar.

***Output:*** Income records summary would be visible.

**R.4.8 : Export income**

***Description:*** This function will let user download all income which recorded in database in excel sheet on their system.

***Input:*** Click on delete record button.

***Output:*** Logged in user's income records would be downloaded.

## 5)Logout user

**R.5.1: logout**

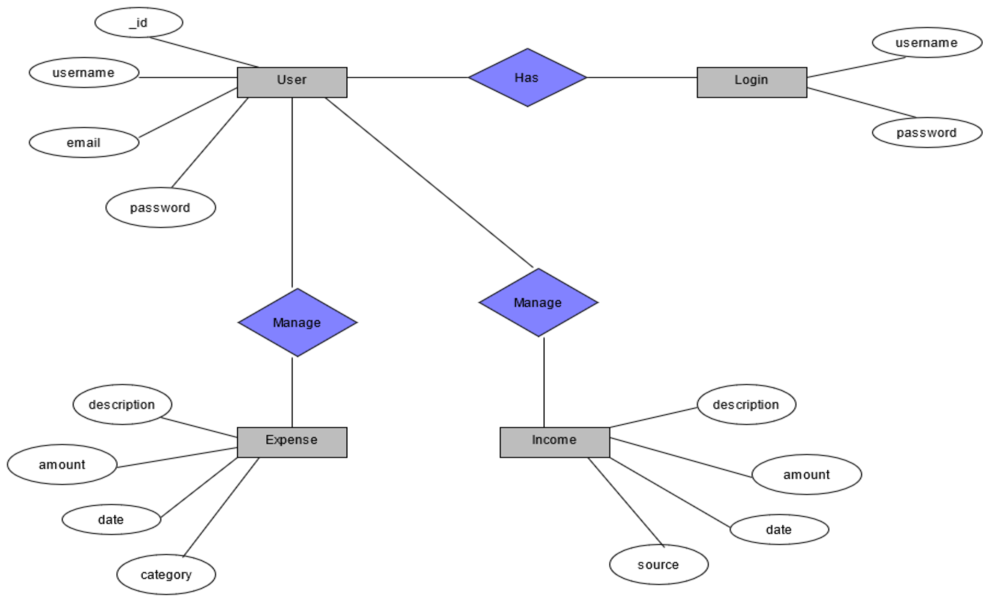
***Description:*** Operator can logout after completion of task. All the entered records would not be affected due to logout.

***Input:*** Click on logout option in side bar.

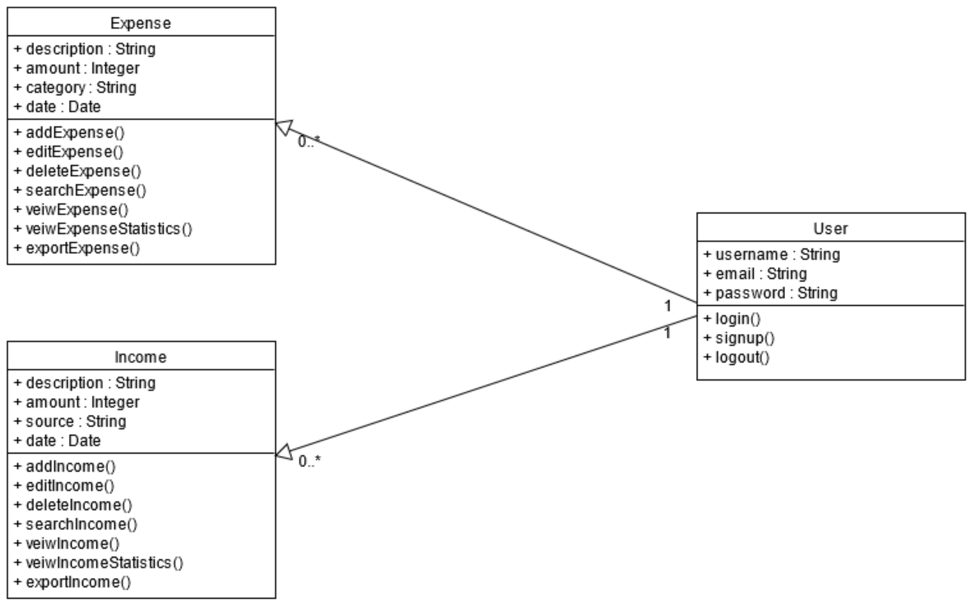
***Output:*** Operator would be redirected to login page again.

## 4)**Design**

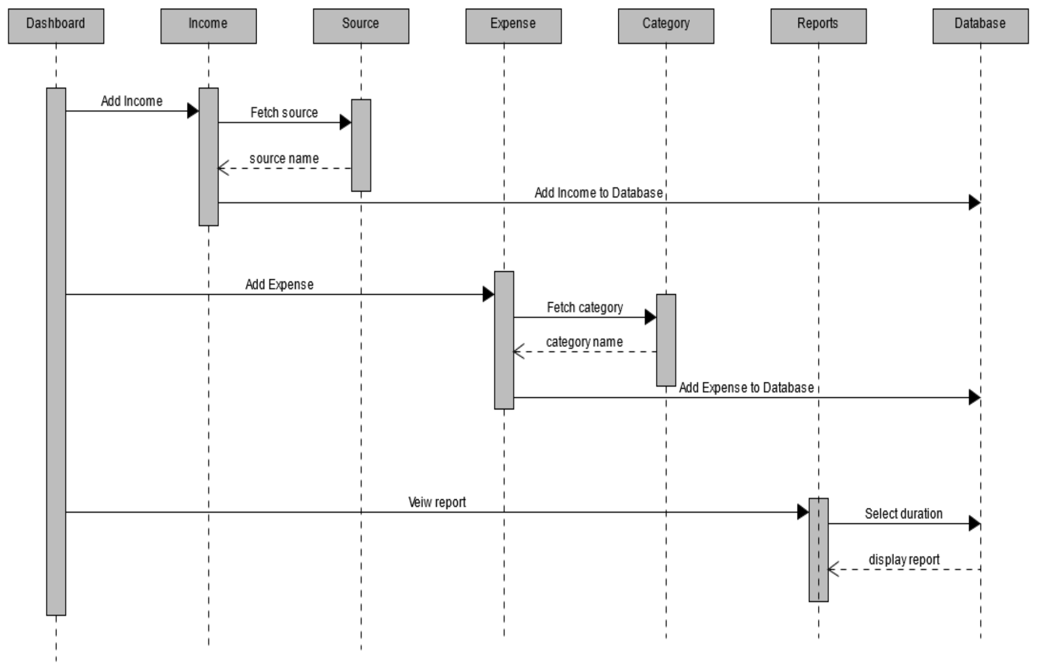
### i. E-R diagram



### ii. Class diagram



## iii. Sequence diagram



## iv. Data Dictionary

| Users | | | | | | | |
|---|---|---|---|---|---|---|---|
| Sr No. | Field name | Data type | Required | Unique | PK/FK | Referred table | Description |
| 1 | _id | Int32 | yes | yes | yes | - | Autoincrement |
| 2 | name | String | yes | yes | no | - | - |
| 3 | email | String | yes | no | no | - | - |
| 4 | password | String | yes | no | no | - | - |

| Category | | | | | | | |
|---|---|---|---|---|---|---|---|
| SrNo. | Field name | Data type | Required | Unique | PK/FK | Referred table | Description |
| 1 | name | String | yes | yes | no | - | - |

| **Expense** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Sr No. | Field name | Data type | Required | Unique | PK/FK | Referred table | Description |
| 1 | _id | Int32 | yes | yes | yes | - | Autoincrement |
| 2 | amount | Int32 | yes | yes | no | - | - |
| 3 | description | String | yes | no | no | - | - |
| 4 | category | String | yes | no | yes | Category | - |
| 5 | date | Date | yes | no | no | - | - |

| Source | | | | | | | |
|---|---|---|---|---|---|---|---|
| Sr No. | Field name | Data type | Required | Unique | PK/FK | Referred table | Description |
| 1 | name | String | yes | yes | no | - | - |

| **Income** | | | | | | | |
|---|---|---|---|---|---|---|---|
| Sr No. | Field name | Data type | Required | Unique | PK/FK | Referred table | Description |
| 1 | _id | Int32 | yes | yes | yes | - | Autoincrement |
| 2 | amount | Int32 | yes | yes | no | - | - |
| 3 | description | String | yes | no | no | - | - |
| 4 | source | String | yes | no | yes | Source | - |
| 5 | date | Date | yes | no | no | - | - |

# 5)Implementation Details

## a).Modules

### Login-Registration

This module is the base for authentication authorization to ensure the security aspect of the user.

It consist of all the login and registration functionality. User can login to the system if account already exist. User can register with unique username to create an account.

User is supposed to provide correct credentials to successfully login to the system.

User can logout from the system whenever he/she wants. This functions would not delete the activities performed by user.

### Expense Records

This module is basically used in entering the record of the expense. Once we enter the information and click on add record the following information would be added to the database.

User can edit the information of expense by clicking on the edit button. All the updated fields will be updated to the database.

User can delete the record of any particular expense by clicking on the delete button. That particular record would be deleted from the list of the expense.

User can search the expense from the list with respect to either amount, description, category date.

User can view statistics of the expense based on selected date by all types of graphs(Pie ,Bar ,Line,

Doughnut).

User can also export records of all expense in excel

Formate in their system by clicking on export CSV

Button.

## Income Records

This module is basically used in entering the record of their income. Once we enter the information and click on add record the following information would be added to the database.

User can edit the information of income by clicking on the edit button. All the updated fields will be updated to the database.

User can delete the record of any particular income by clicking on the delete button. That particular record would be deleted from the list of the income.

User can search the income from the list with respect to either amount, description, source or date.

User can view statistics of the income based on selected date by all types of graphs(Pie ,Bar ,Line,

Doughnut).

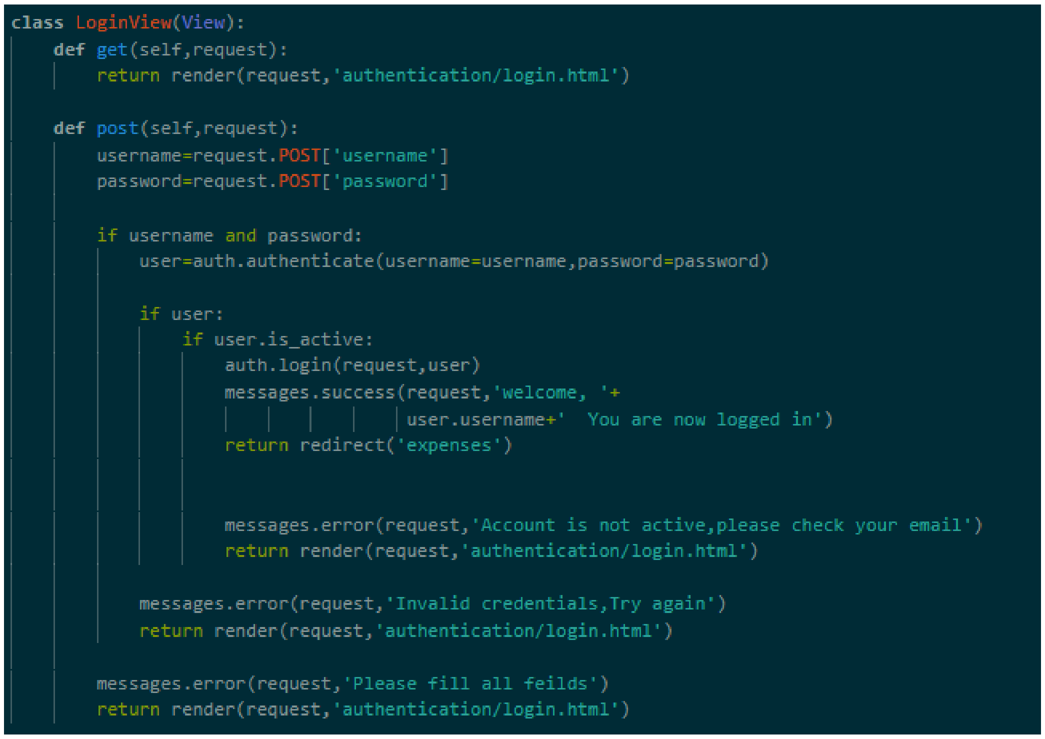
User can also export records of all income in excel

Format in their system by clicking on Export CSV

Button.

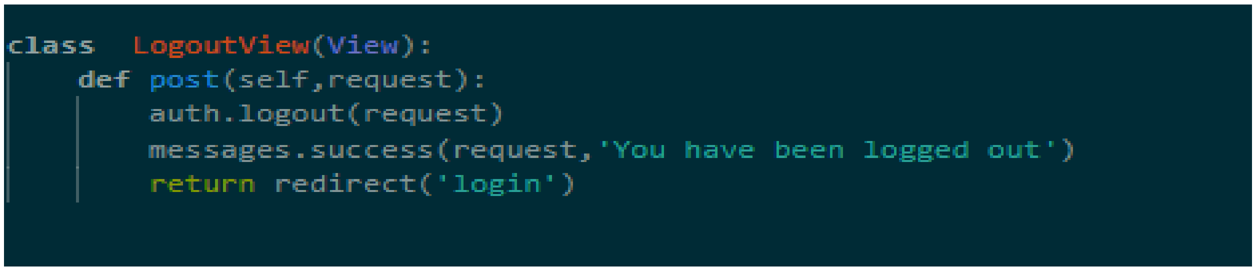
## b).Major functionalities

### Login:



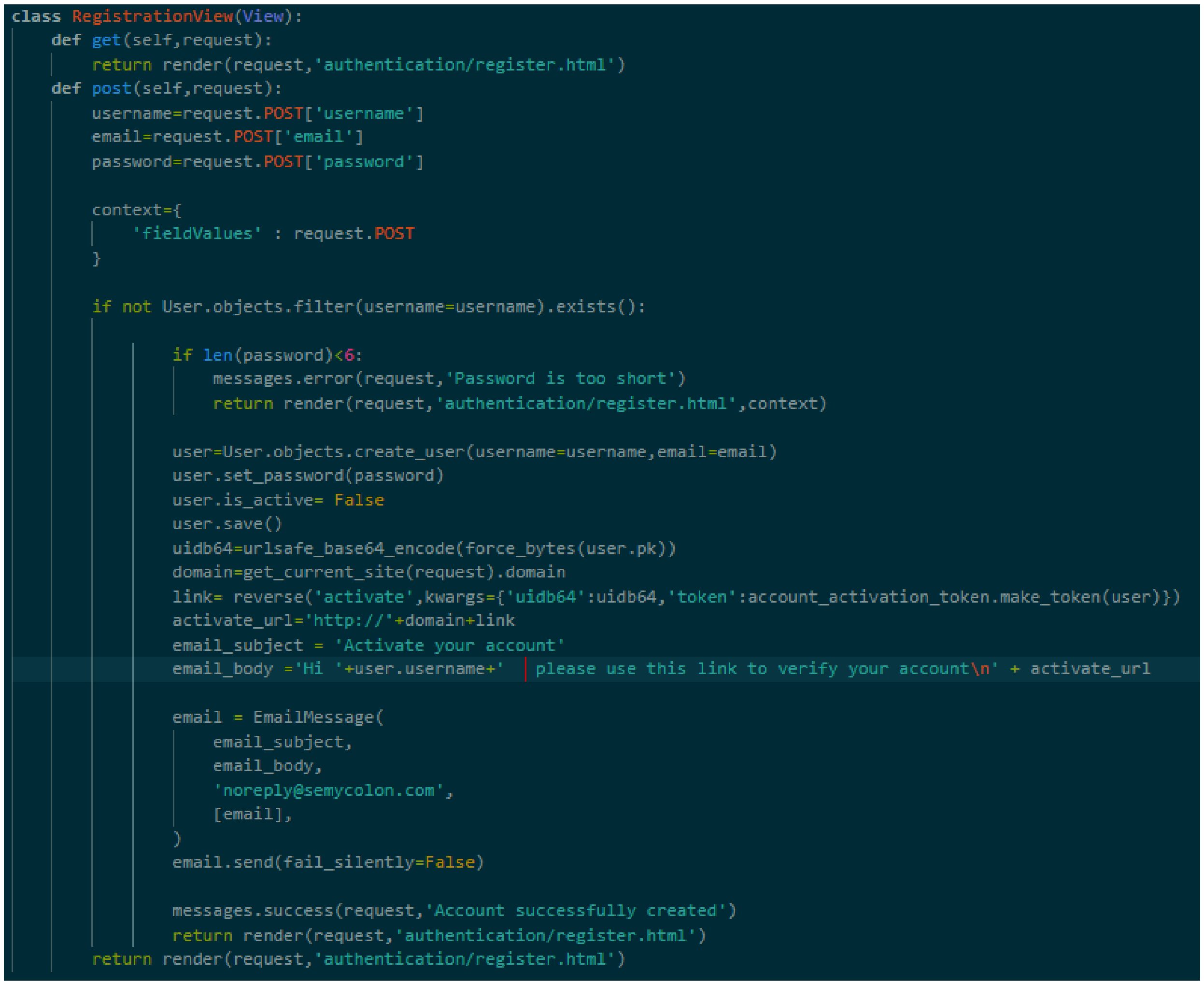
```
class LoginView(View):
    def get(self,request):
        return render(request,'authentication/login.html')

    def post(self,request):
        username=request.POST['username']
        password=request.POST['password']

        if username and password:
            user=auth.authenticate(username=username,password=password)

            if user:
                if user.is_active:
                    auth.login(request,user)
                    messages.success(request,'welcome, '+
                                     user.username+'  You are now logged in')
                    return redirect('expenses')


                    messages.error(request,'Account is not active,please check your email')
                    return render(request,'authentication/login.html')

            messages.error(request,'Invalid credentials,Try again')
            return render(request,'authentication/login.html')

        messages.error(request,'Please fill all feilds')
        return render(request,'authentication/login.html')
```

### Logout:



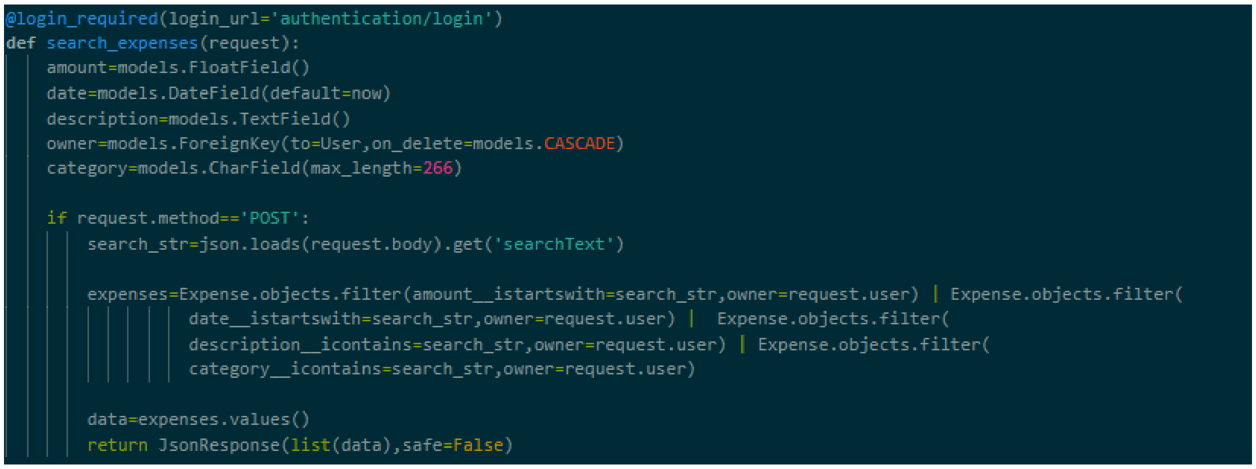
```
class  LogoutView(View):
    def post(self,request):
        auth.logout(request)
        messages.success(request,'You have been logged out')
        return redirect('login')
```

## Registration:



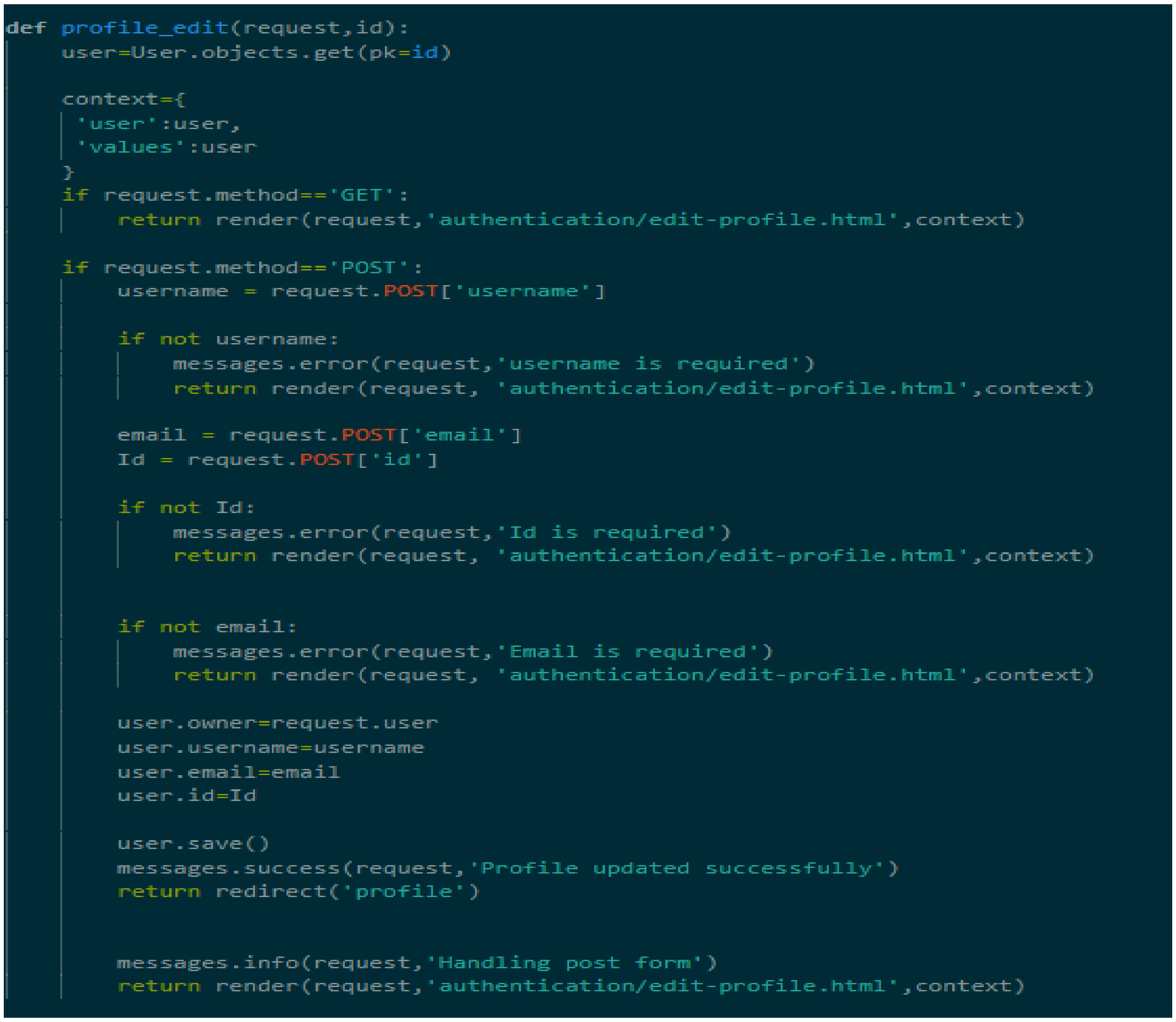
```python
class RegistrationView(View):
    def get(self,request):
        return render(request,'authentication/register.html')
    def post(self,request):
        username=request.POST['username']
        email=request.POST['email']
        password=request.POST['password']

        context={
            'fieldValues' : request.POST
        }

        if not User.objects.filter(username=username).exists():

                if len(password)<6:
                    messages.error(request,'Password is too short')
                    return render(request,'authentication/register.html',context)

                user=User.objects.create_user(username=username,email=email)
                user.set_password(password)
                user.is_active= False
                user.save()
                uidb64=urlsafe_base64_encode(force_bytes(user.pk))
                domain=get_current_site(request).domain
                link= reverse('activate',kwargs={'uidb64':uidb64,'token':account_activation_token.make_token(user)})
                activate_url='http://'+domain+link
                email_subject = 'Activate your account'
                email_body ='Hi '+user.username+'  please use this link to verify your account\n' + activate_url

                email = EmailMessage(
                    email_subject,
                    email_body,
                    'noreply@semycolon.com',
                    [email],
                )
                email.send(fail_silently=False)

                messages.success(request,'Account successfully created')
                return render(request,'authentication/register.html')
        return render(request,'authentication/register.html')
```

## Search Expense:



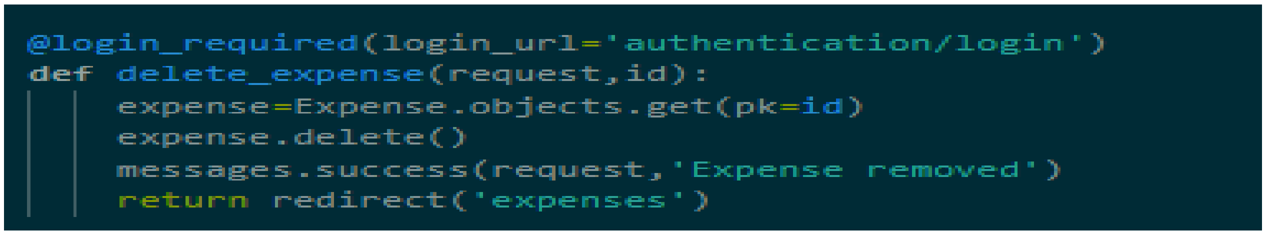
```python
@login_required(login_url='authentication/login')
def search_expenses(request):
    amount=models.FloatField()
    date=models.DateField(default=now)
    description=models.TextField()
    owner=models.ForeignKey(to=User,on_delete=models.CASCADE)
    category=models.CharField(max_length=266)

    if request.method=='POST':
        search_str=json.loads(request.body).get('searchText')

        expenses=Expense.objects.filter(amount__istartswith=search_str,owner=request.user) | Expense.objects.filter(
                date__istartswith=search_str,owner=request.user) |  Expense.objects.filter(
                description__icontains=search_str,owner=request.user) | Expense.objects.filter(
                category__icontains=search_str,owner=request.user)

        data=expenses.values()
        return JsonResponse(list(data),safe=False)
```

## Profile-Edit:



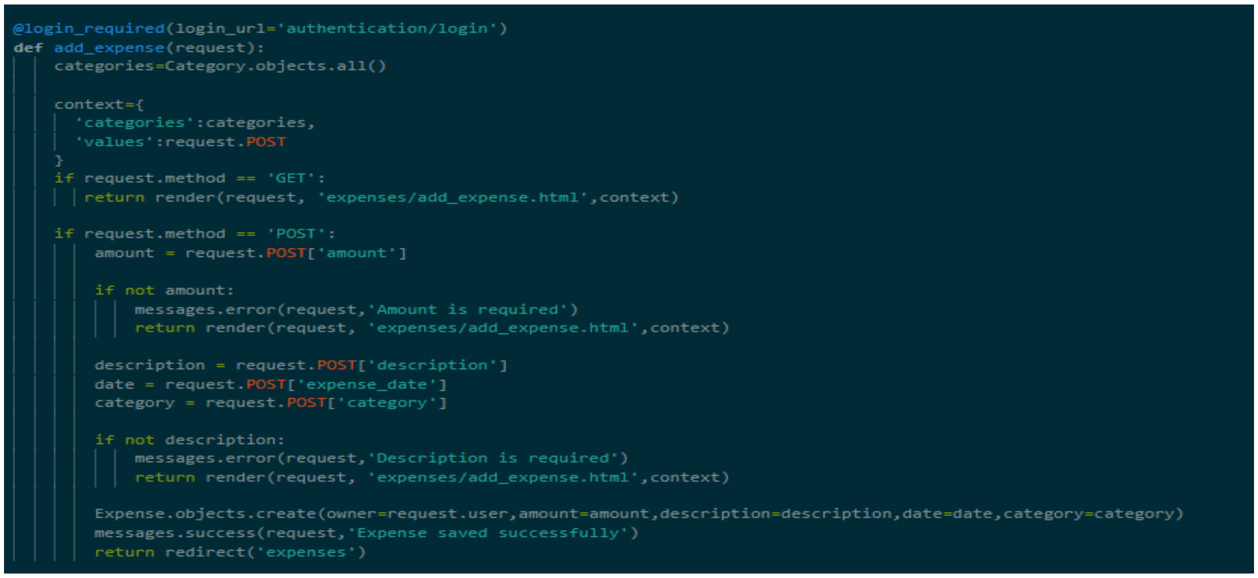
```
def profile_edit(request,id):
    user=User.objects.get(pk=id)

    context={
     'user':user,
     'values':user
    }
    if request.method=='GET':
        return render(request,'authentication/edit-profile.html',context)

    if request.method=='POST':
        username = request.POST['username']

        if not username:
            messages.error(request,'username is required')
            return render(request, 'authentication/edit-profile.html',context)

        email = request.POST['email']
        Id = request.POST['id']

        if not Id:
            messages.error(request,'Id is required')
            return render(request, 'authentication/edit-profile.html',context)


        if not email:
            messages.error(request,'Email is required')
            return render(request, 'authentication/edit-profile.html',context)

        user.owner=request.user
        user.username=username
        user.email=email
        user.id=Id

        user.save()
        messages.success(request,'Profile updated successfully')
        return redirect('profile')


        messages.info(request,'Handling post form')
        return render(request,'authentication/edit-profile.html',context)
```

## Delete Expenes:



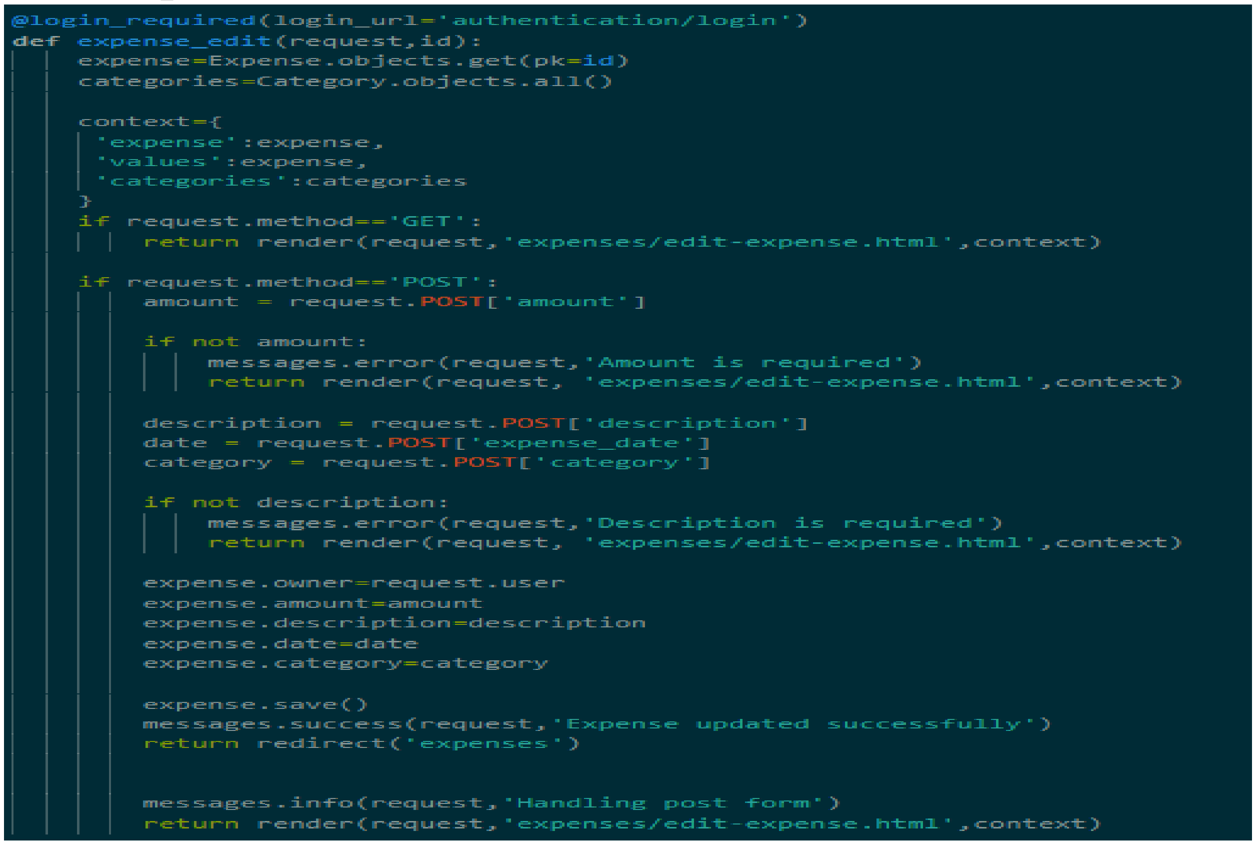
```
@login_required(login_url='authentication/login')
def delete_expense(request,id):
        expense=Expense.objects.get(pk=id)
        expense.delete()
        messages.success(request,'Expense removed')
        return redirect('expenses')
```

## Add Expense:



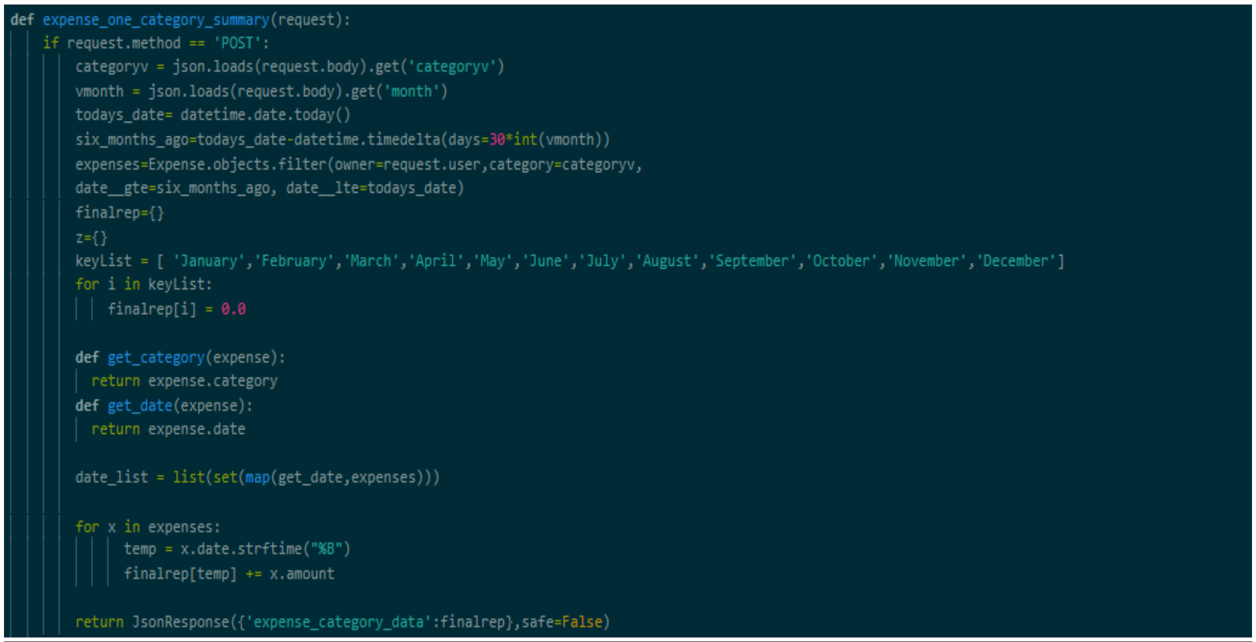
```python
@login_required(login_url='authentication/login')
def add_expense(request):
    categories=Category.objects.all()

    context={
        'categories':categories,
        'values':request.POST
    }
    if request.method == 'GET':
        return render(request, 'expenses/add_expense.html',context)

    if request.method == 'POST':
        amount = request.POST['amount']

        if not amount:
            messages.error(request,'Amount is required')
            return render(request, 'expenses/add_expense.html',context)

        description = request.POST['description']
        date = request.POST['expense_date']
        category = request.POST['category']

        if not description:
            messages.error(request,'Description is required')
            return render(request, 'expenses/add_expense.html',context)

        Expense.objects.create(owner=request.user,amount=amount,description=description,date=date,category=category)
        messages.success(request,'Expense saved successfully')
        return redirect('expenses')
```

## Edit Expense :



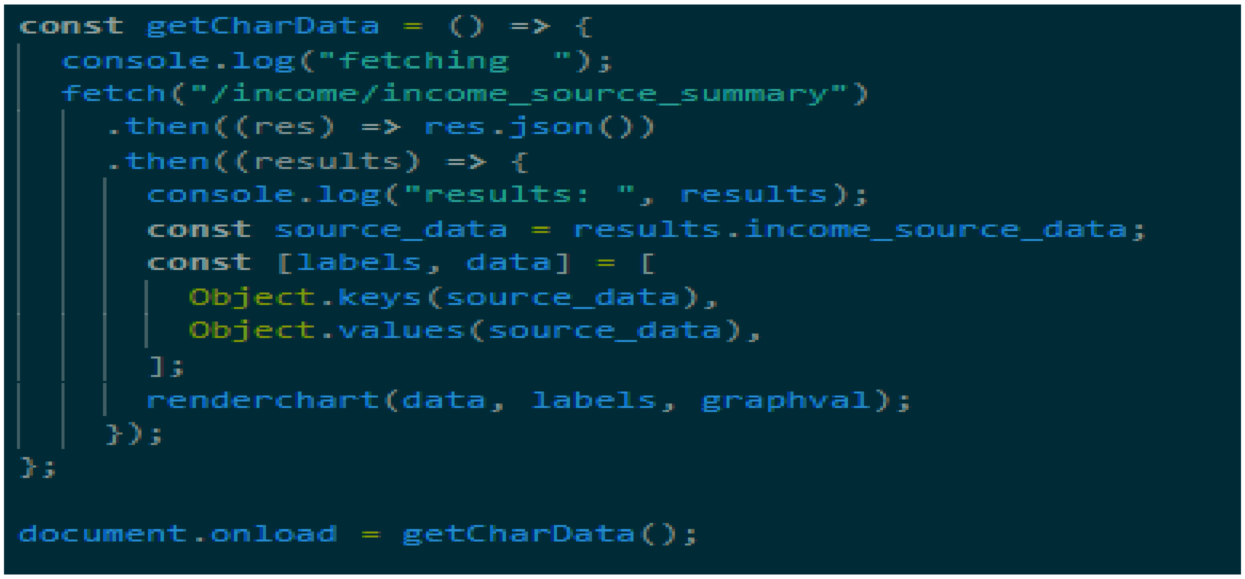
```python
@login_required(login_url='authentication/login')
def expense_edit(request,id):
    expense=Expense.objects.get(pk=id)
    categories=Category.objects.all()

    context={
        'expense':expense,
        'values':expense,
        'categories':categories
    }
    if request.method=='GET':
        return render(request,'expenses/edit-expense.html',context)

    if request.method=='POST':
        amount = request.POST['amount']

        if not amount:
            messages.error(request,'Amount is required')
            return render(request, 'expenses/edit-expense.html',context)

        description = request.POST['description']
        date = request.POST['expense_date']
        category = request.POST['category']

        if not description:
            messages.error(request,'Description is required')
            return render(request, 'expenses/edit-expense.html',context)

        expense.owner=request.user
        expense.amount=amount
        expense.description=description
        expense.date=date
        expense.category=category

        expense.save()
        messages.success(request,'Expense updated successfully')
        return redirect('expenses')


        messages.info(request,'Handling post form')
        return render(request,'expenses/edit-expense.html',context)
```

## Expense-Report:



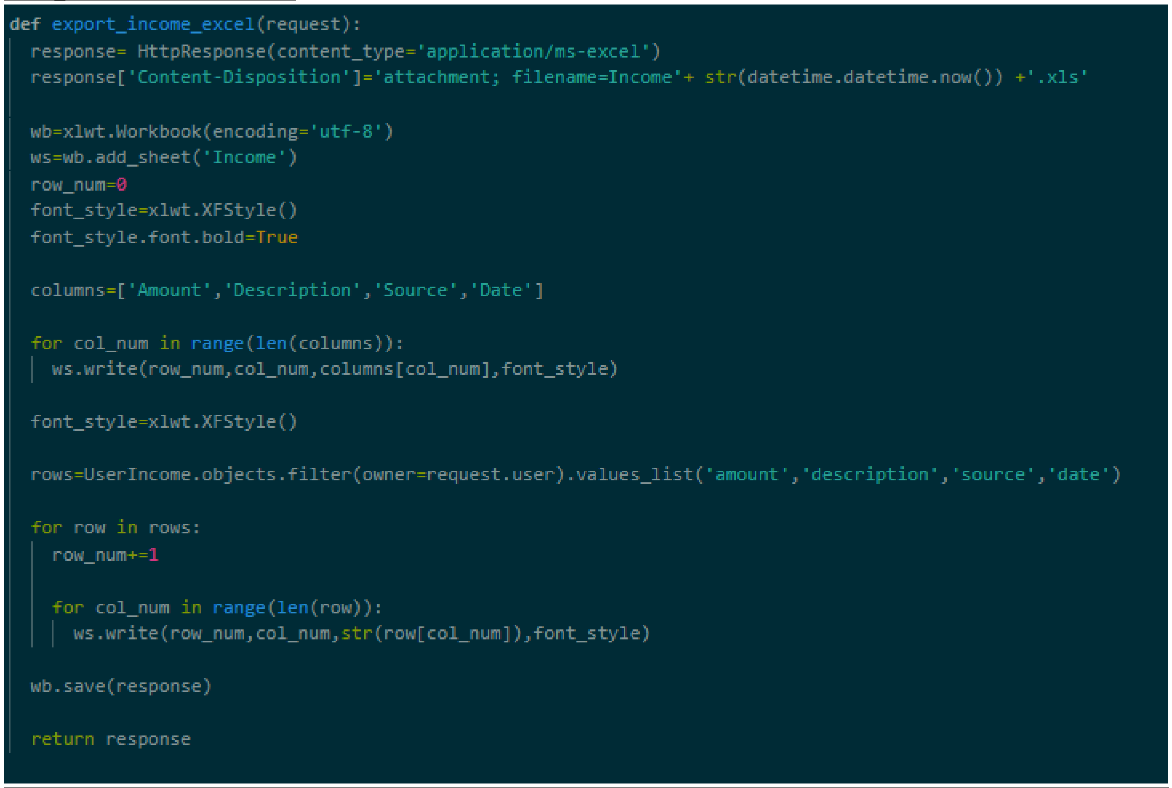
```python
def expense_one_category_summary(request):
    if request.method == 'POST':
        categoryv = json.loads(request.body).get('categoryv')
        vmonth = json.loads(request.body).get('month')
        todays_date= datetime.date.today()
        six_months_ago=todays_date-datetime.timedelta(days=30*int(vmonth))
        expenses=Expense.objects.filter(owner=request.user,category=categoryv,
        date__gte=six_months_ago, date__lte=todays_date)
        finalrep={}
        z={}
        keyList = [ 'January','February','March','April','May','June','July','August','September','October','November','December']
        for i in keyList:
            finalrep[i] = 0.0

        def get_category(expense):
            return expense.category
        def get_date(expense):
            return expense.date

        date_list = list(set(map(get_date,expenses)))

        for x in expenses:
                temp = x.date.strftime("%B")
                finalrep[temp] += x.amount

        return JsonResponse({'expense_category_data':finalrep},safe=False)
```

## Js file of Income Chart:



```javascript
const getCharData = () => {
  console.log("fetching  ");
  fetch("/income/income_source_summary")
    .then((res) => res.json())
    .then((results) => {
      console.log("results: ", results);
      const source_data = results.income_source_data;
      const [labels, data] = [
        Object.keys(source_data),
        Object.values(source_data),
      ];
      renderchart(data, labels, graphval);
    });
};

document.onload = getCharData();
```

## Export-Income:



```
def export_income_excel(request):
  response= HttpResponse(content_type='application/ms-excel')
  response['Content-Disposition']='attachment; filename=Income'+ str(datetime.datetime.now()) +'.xls'

  wb=xlwt.Workbook(encoding='utf-8')
  ws=wb.add_sheet('Income')
  row_num=0
  font_style=xlwt.XFStyle()
  font_style.font.bold=True

  columns=['Amount','Description','Source','Date']

  for col_num in range(len(columns)):
    ws.write(row_num,col_num,columns[col_num],font_style)

  font_style=xlwt.XFStyle()

  rows=UserIncome.objects.filter(owner=request.user).values_list('amount','description','source','date')

  for row in rows:
    row_num+=1

    for col_num in range(len(row)):
      ws.write(row_num,col_num,str(row[col_num]),font_style)

  wb.save(response)

  return response
```

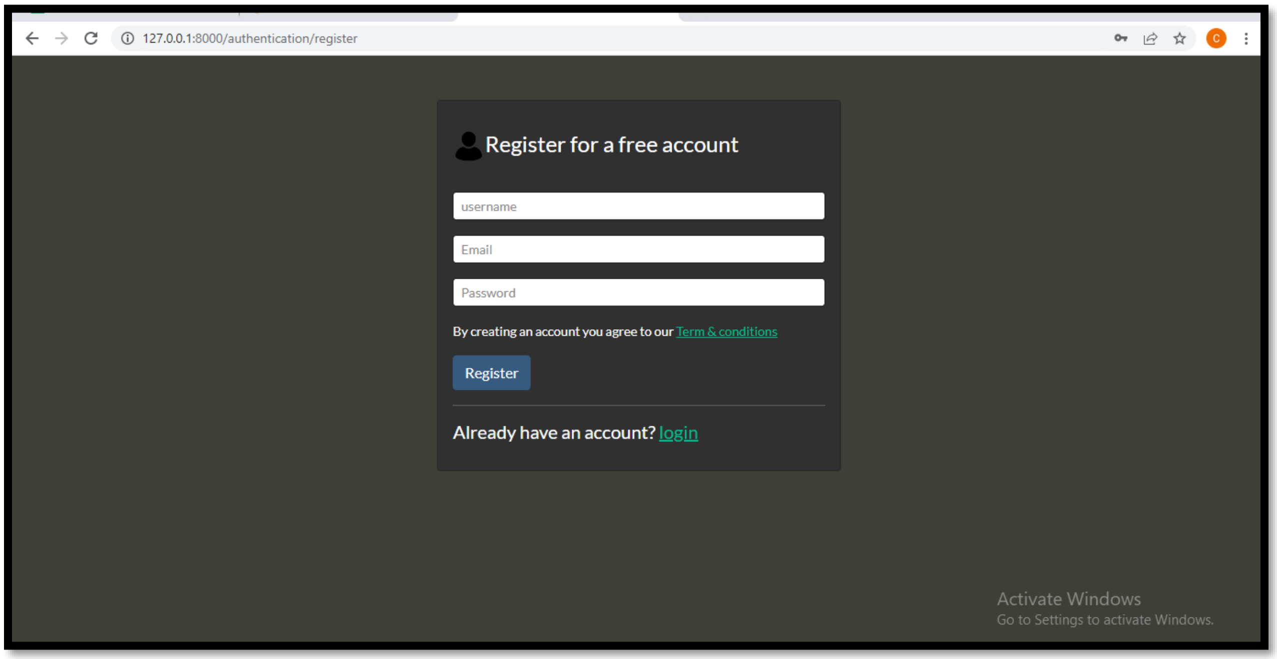
## **5) Testing**

**Testing method:** Manual testing

| Sr no. | Test Scenario | Expected result | Actual result | Status |
|---|---|---|---|---|
| 1 | Login with incorrect credentials | Alert box is popped up with a message | Alert box is popped up | Success |
| 2 | Entering different password and confirm password | 'password does not match' alert box should be displayed | 'password does not match' alert box is displayed | Success |
| 3 | Clicking on login/register button with empty fields | 'invalid input' alert box should be appeared | 'invalid input' alert box is appeared | Success |
| 4 | Add Expense/income record to the database | Record should be added to the list of all the Record | Record is added to the list | Success |
| 5 | Edit Expense/income record | Edit form should be visible | Edit form is visible | Success |
| 6 | Click on Export Excel | Record File should be download | Record File download | Success |
| 7 | Expense/income record click on Delete | That recorded should be Deleted | Recorded Deleted | Success |
| 18 | Click on Expense/income Report | Expense/income page should be visible and display summary | Expense/income page is visible display summary | Success |
| 12 | Click on edit profile button | Profile should be updated | Profile is updated | Success |

# 7)Screen-shots

- **Registration**



127.0.0.1:8000/authentication/register

Register for a free account

username

Email

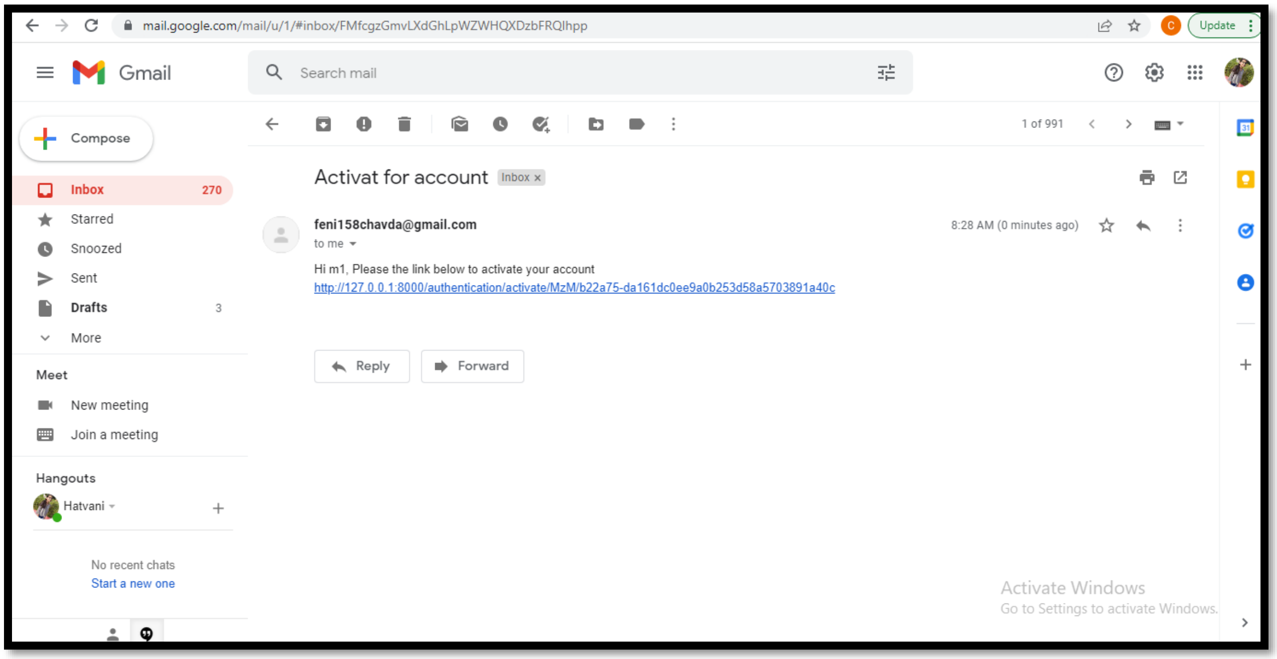
Password

By creating an account you agree to our Term & conditions

Register

Already have an account? login

Activate Windows
Go to Settings to activate Windows.



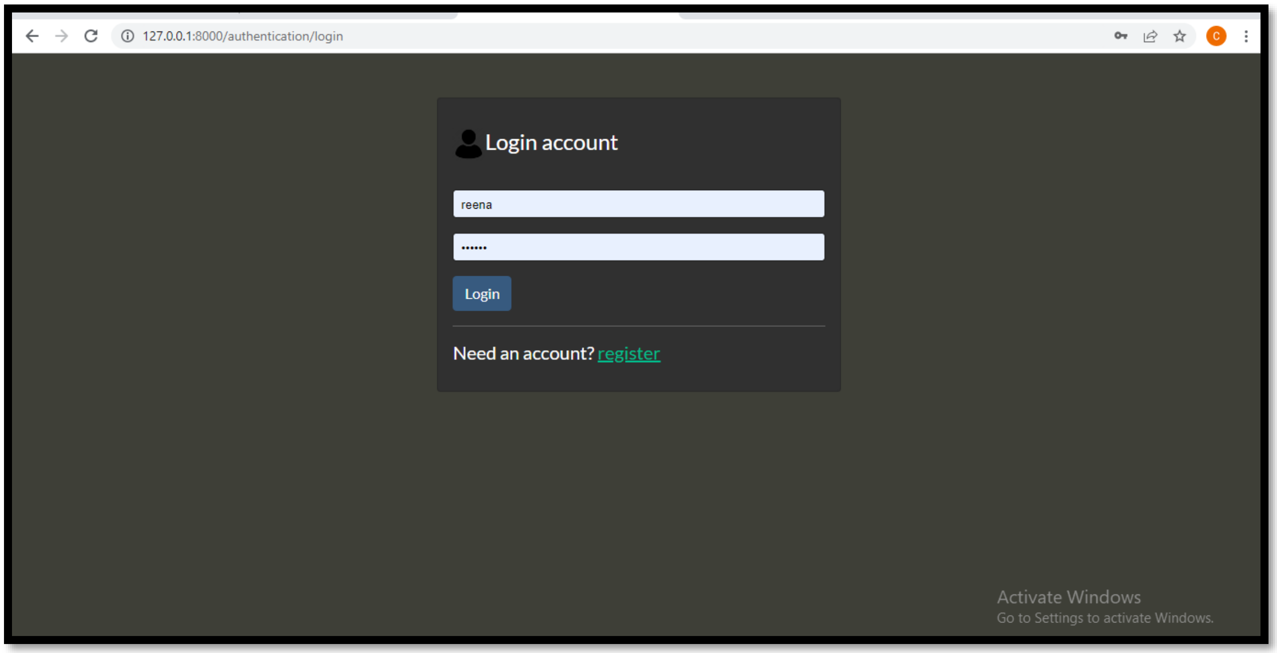
mail.google.com/mail/u/1/#inbox/FMfcgzGmvLXdGhLpWZWHQXDzbFRQlhpp

Activat for account Inbox

feni158chavda@gmail.com
to me
8:28 AM (0 minutes ago)

Hi m1, Please the link below to activate your account
http://127.0.0.1:8000/authentication/activate/MzM/b22a75-da161dc0ee9a0b253d58a5703891a40c

- **Login :**



127.0.0.1:8000/authentication/login

Login account

reena

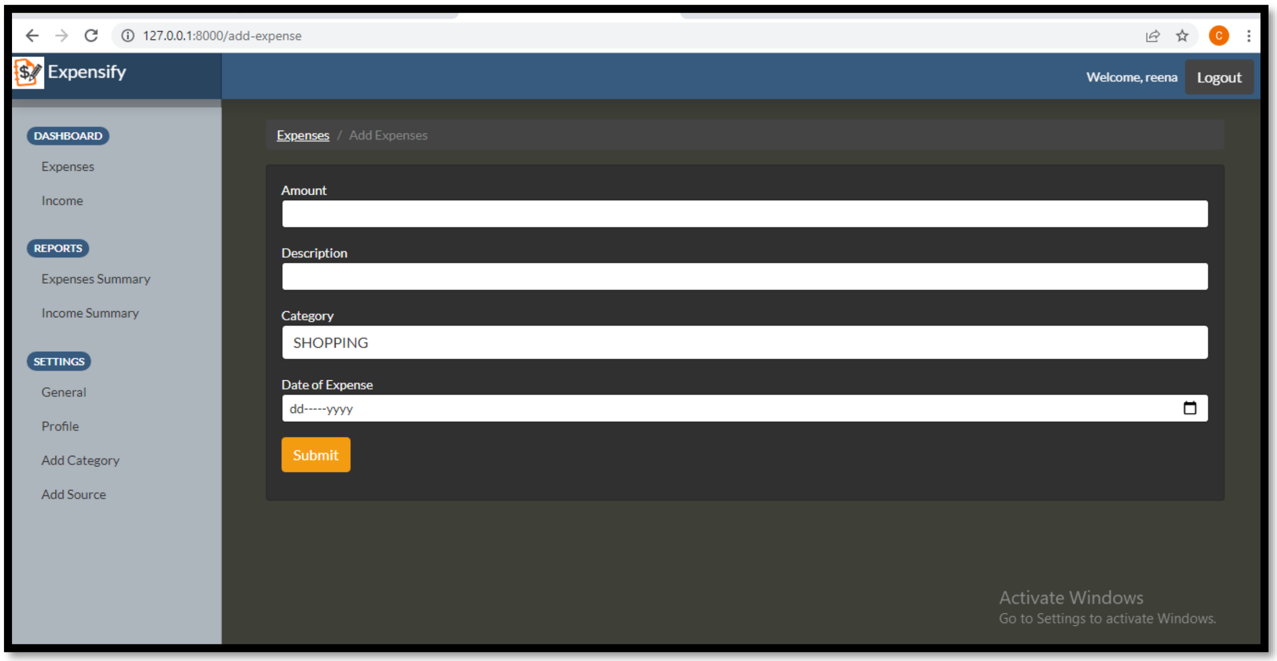
••••••

Login

Need an account? register

Activate Windows
Go to Settings to activate Windows.

- **Add-Expense:**



127.0.0.1:8000/add-expense

Expensify

Welcome, reena   Logout

DASHBOARD
Expenses
Income
REPORTS
Expenses Summary
Income Summary
SETTINGS
General
Profile
Add Category
Add Source

Expenses / Add Expenses

Amount

Description

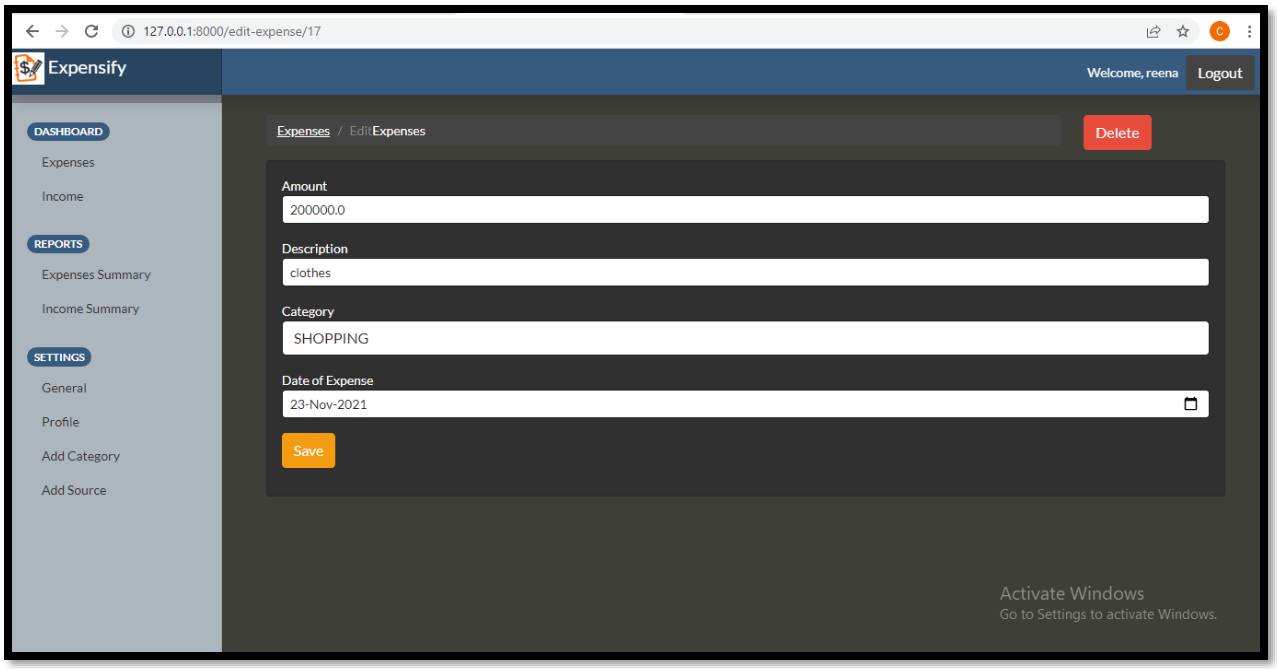
Category
SHOPPING

Date of Expense
dd-----yyyy

Submit

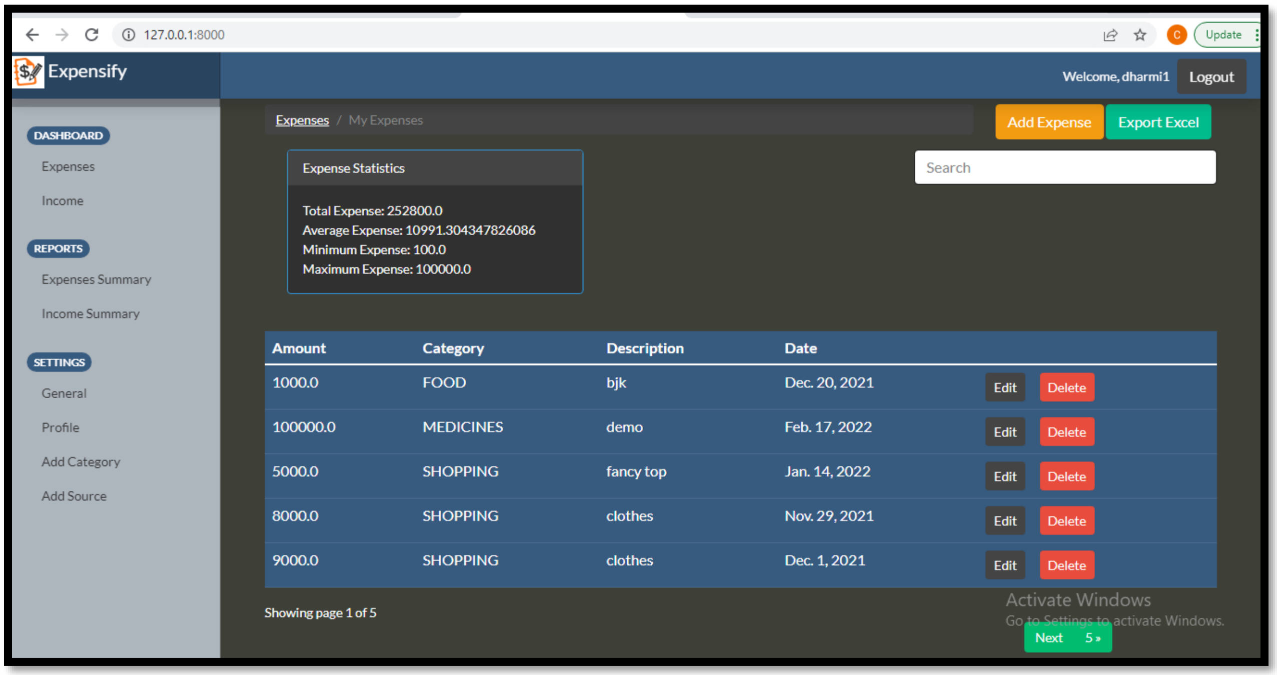
Activate Windows
Go to Settings to activate Windows.

- **Edit-Expense:**



Amount: 200000.0
Description: clothes
Category: SHOPPING
Date of Expense: 23-Nov-2021

- **Pagination and Display Statistic:**



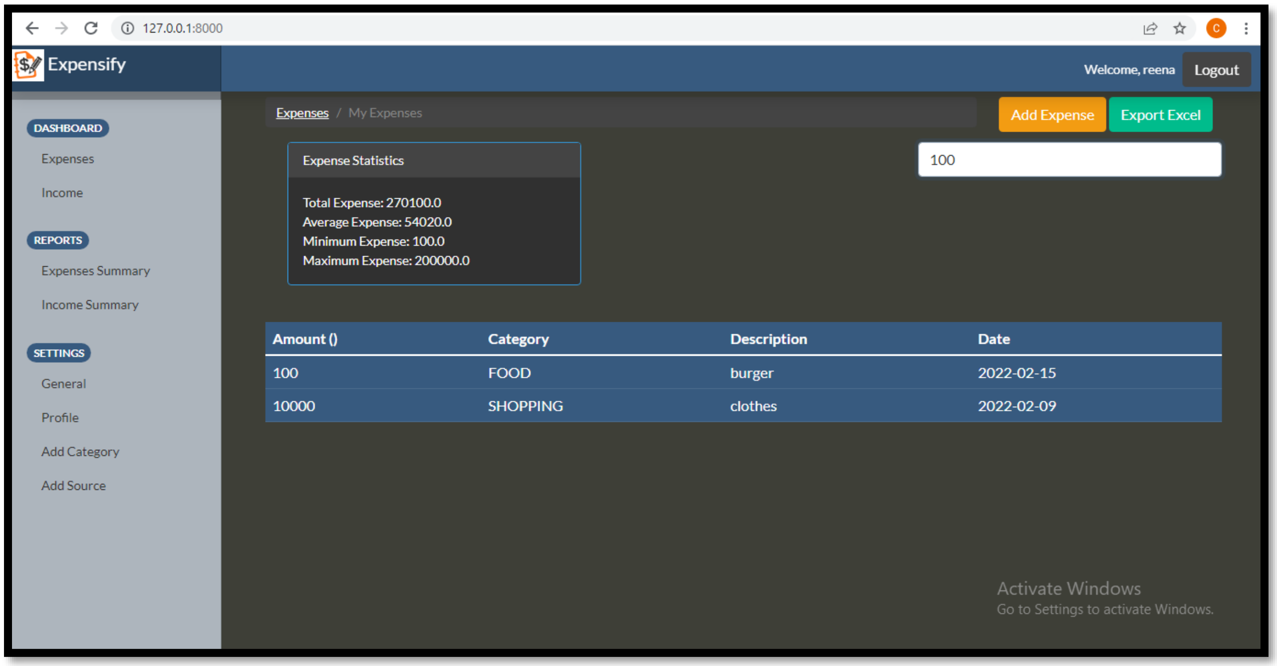
Expense Statistics

Total Expense: 252800.0
Average Expense: 10991.304347826086
Minimum Expense: 100.0
Maximum Expense: 100000.0

| Amount | Category | Description | Date | | |
|---|---|---|---|---|---|
| 1000.0 | FOOD | bjk | Dec. 20, 2021 | Edit | Delete |
| 100000.0 | MEDICINES | demo | Feb. 17, 2022 | Edit | Delete |
| 5000.0 | SHOPPING | fancy top | Jan. 14, 2022 | Edit | Delete |
| 8000.0 | SHOPPING | clothes | Nov. 29, 2021 | Edit | Delete |
| 9000.0 | SHOPPING | clothes | Dec. 1, 2021 | Edit | Delete |

Showing page 1 of 5

- **Expense Search :**



127.0.0.1:8000

Expensify

Welcome, reena Logout

Expenses / My Expenses

100

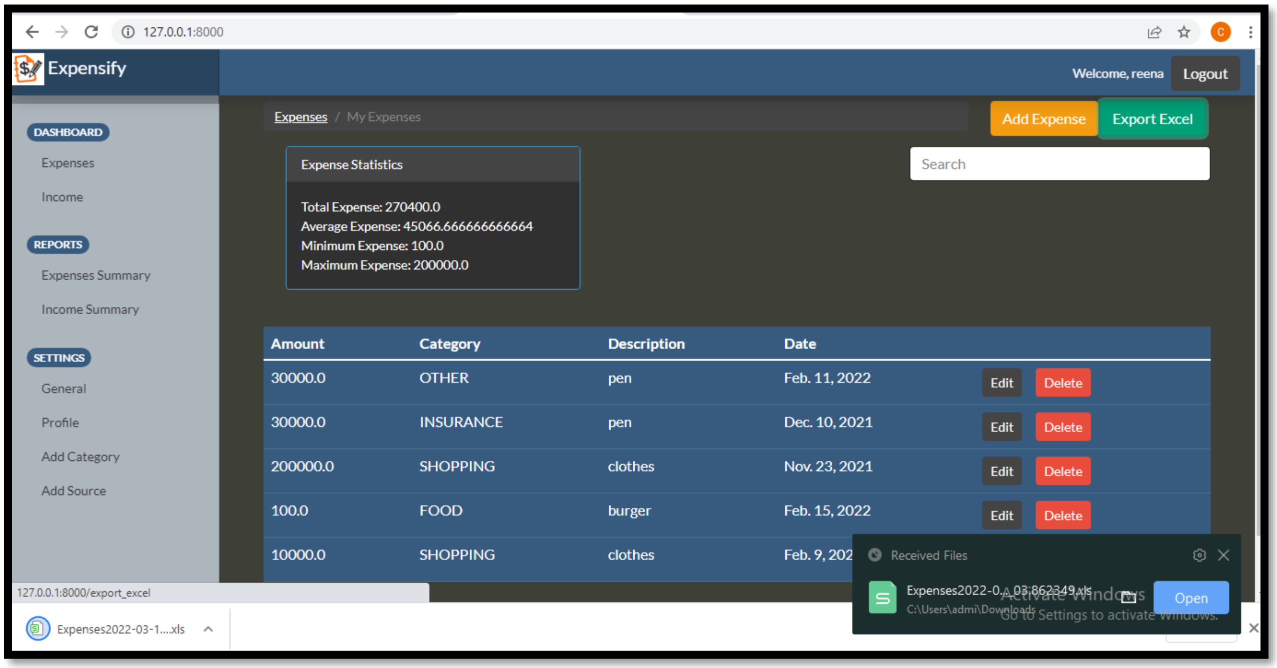
Expense Statistics

Total Expense: 270100.0
Average Expense: 54020.0
Minimum Expense: 100.0
Maximum Expense: 200000.0

| Amount () | Category | Description | Date |
|---|---|---|---|
| 100 | FOOD | burger | 2022-02-15 |
| 10000 | SHOPPING | clothes | 2022-02-09 |

Activate Windows
Go to Settings to activate Windows.

- **Export Excel File:**



127.0.0.1:8000

Expensify

Welcome, reena Logout

Expenses / My Expenses

Search

Expense Statistics

Total Expense: 270400.0
Average Expense: 45066.666666666664
Minimum Expense: 100.0
Maximum Expense: 200000.0

| Amount | Category | Description | Date | |
|---|---|---|---|---|
| 30000.0 | OTHER | pen | Feb. 11, 2022 | Edit Delete |
| 30000.0 | INSURANCE | pen | Dec. 10, 2021 | Edit Delete |
| 200000.0 | SHOPPING | clothes | Nov. 23, 2021 | Edit Delete |
| 100.0 | FOOD | burger | Feb. 15, 2022 | Edit Delete |
| 10000.0 | SHOPPING | clothes | Feb. 9, 202 | |

127.0.0.1:8000/export_excel

Expenses2022-03-1....xls

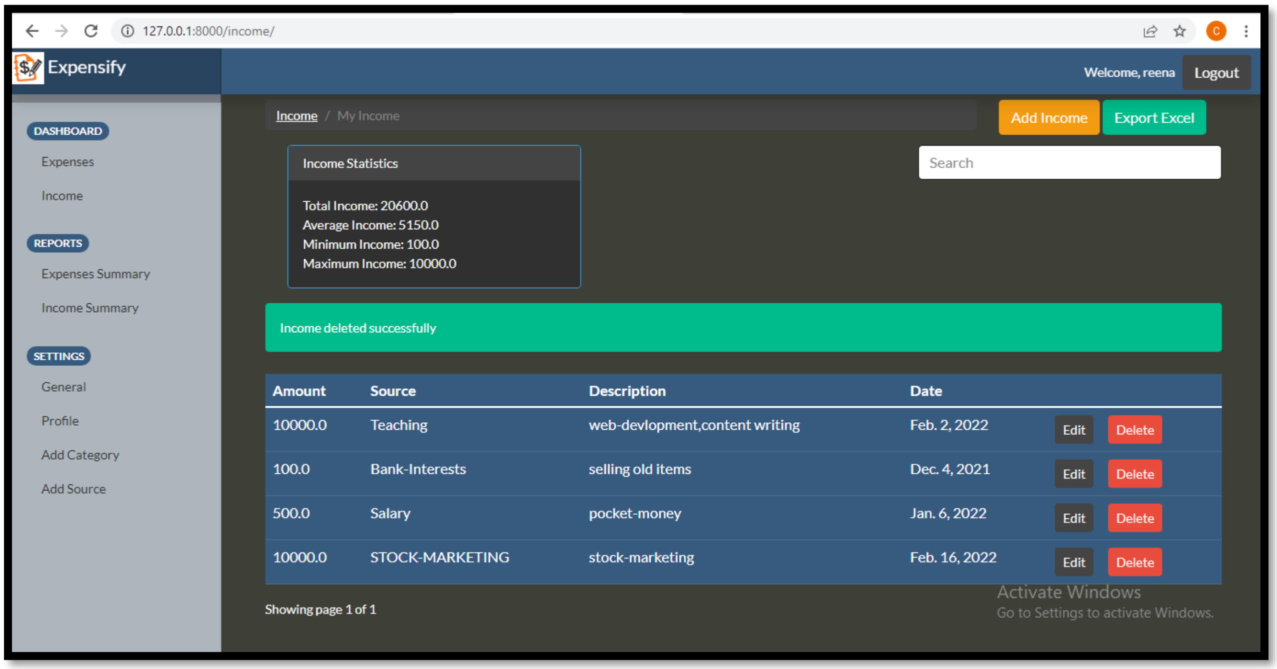
Received Files

Expenses2022-0...03.862349.xls
C:\Users\admi\Downloads

Open

Activate Windows
Go to Settings to activate Windows.

- **Income Delete:**



127.0.0.1:8000/income/

Expensify

Welcome, reena Logout

DASHBOARD
- Expenses
- Income

REPORTS
- Expenses Summary
- Income Summary

SETTINGS
- General
- Profile
- Add Category
- Add Source

Income / My Income

Add Income Export Excel

Search

Income Statistics

Total Income: 20600.0
Average Income: 5150.0
Minimum Income: 100.0
Maximum Income: 10000.0

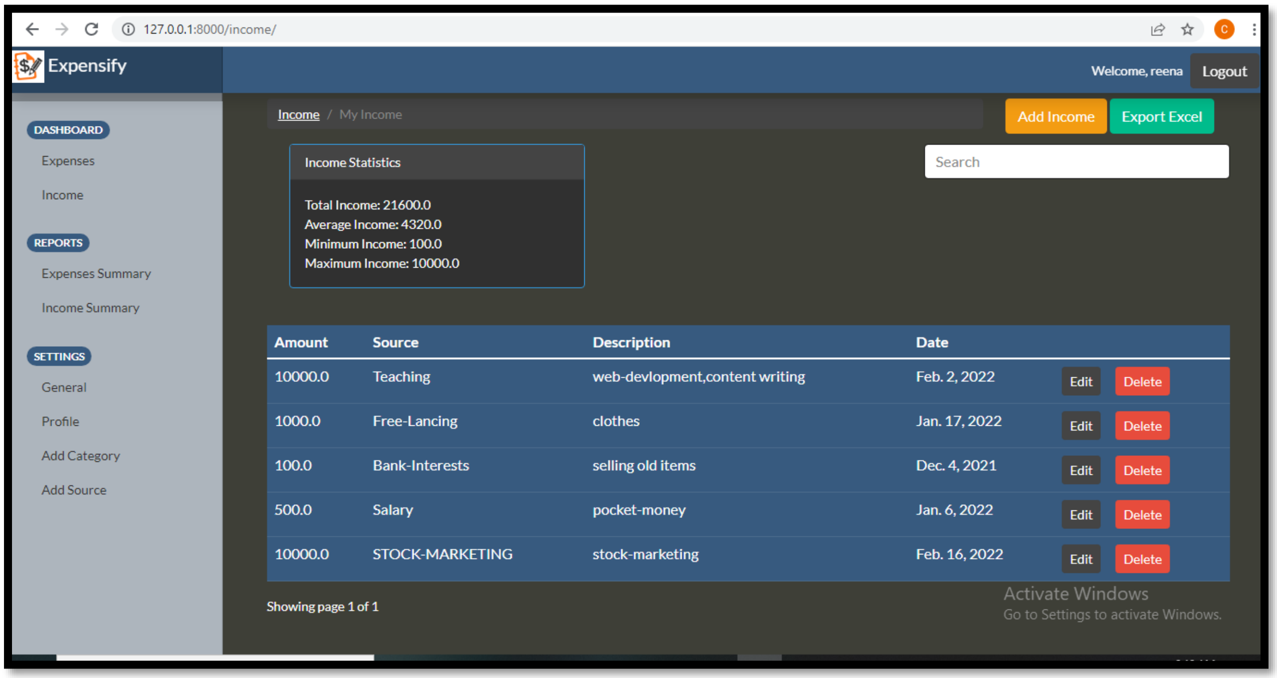
Income deleted successfully

| Amount | Source | Description | Date | | |
|---|---|---|---|---|---|
| 10000.0 | Teaching | web-devlopment,content writing | Feb. 2, 2022 | Edit | Delete |
| 100.0 | Bank-Interests | selling old items | Dec. 4, 2021 | Edit | Delete |
| 500.0 | Salary | pocket-money | Jan. 6, 2022 | Edit | Delete |
| 10000.0 | STOCK-MARKETING | stock-marketing | Feb. 16, 2022 | Edit | Delete |

Showing page 1 of 1

Activate Windows
Go to Settings to activate Windows.

- **Display Income Record and Statistic:**



127.0.0.1:8000/income/

Expensify

Welcome, reena Logout

DASHBOARD
- Expenses
- Income

REPORTS
- Expenses Summary
- Income Summary

SETTINGS
- General
- Profile
- Add Category
- Add Source

Income / My Income

Add Income Export Excel

Search

Income Statistics

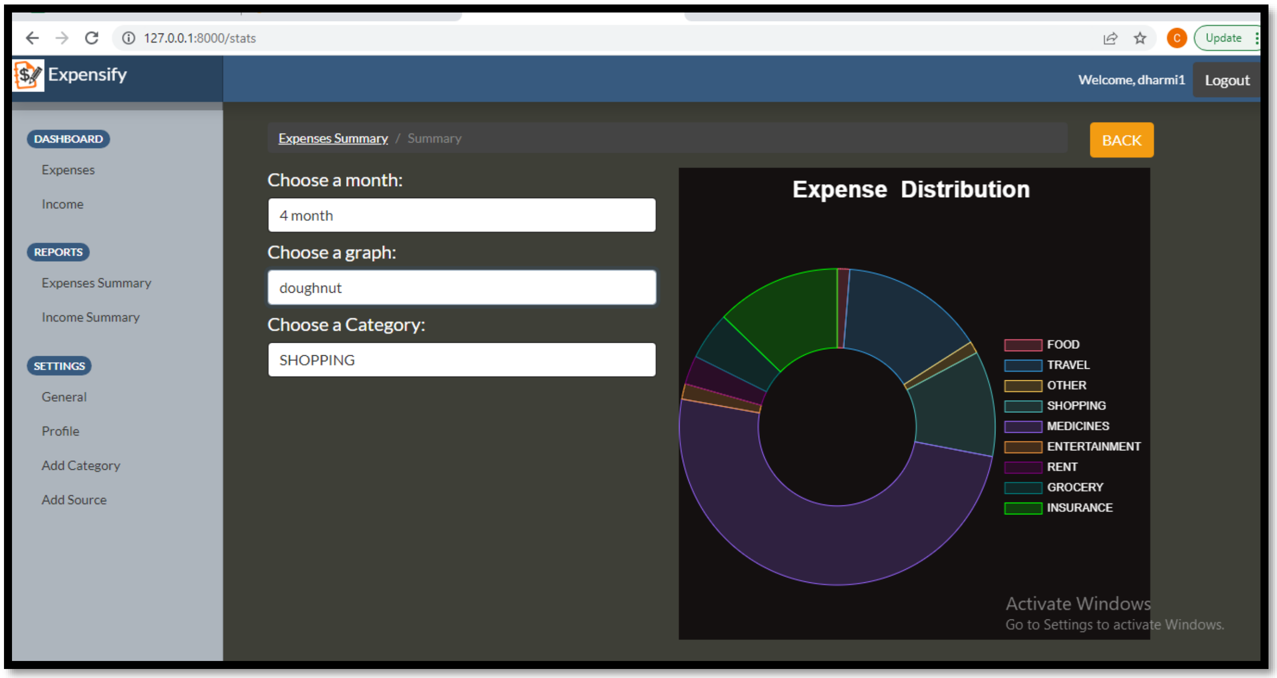
Total Income: 21600.0
Average Income: 4320.0
Minimum Income: 100.0
Maximum Income: 10000.0

| Amount | Source | Description | Date | | |
|---|---|---|---|---|---|
| 10000.0 | Teaching | web-devlopment,content writing | Feb. 2, 2022 | Edit | Delete |
| 1000.0 | Free-Lancing | clothes | Jan. 17, 2022 | Edit | Delete |
| 100.0 | Bank-Interests | selling old items | Dec. 4, 2021 | Edit | Delete |
| 500.0 | Salary | pocket-money | Jan. 6, 2022 | Edit | Delete |
| 10000.0 | STOCK-MARKETING | stock-marketing | Feb. 16, 2022 | Edit | Delete |

Showing page 1 of 1

Activate Windows
Go to Settings to activate Windows.

- **Expenses and Income Report** Category wise and Filter it with month and One Category **:**


127.0.0.1:8000/stats

Expensify

Welcome, dharmi1 Logout

DASHBOARD
Expenses
Income
REPORTS
Expenses Summary
Income Summary
SETTINGS
General
Profile
Add Category
Add Source

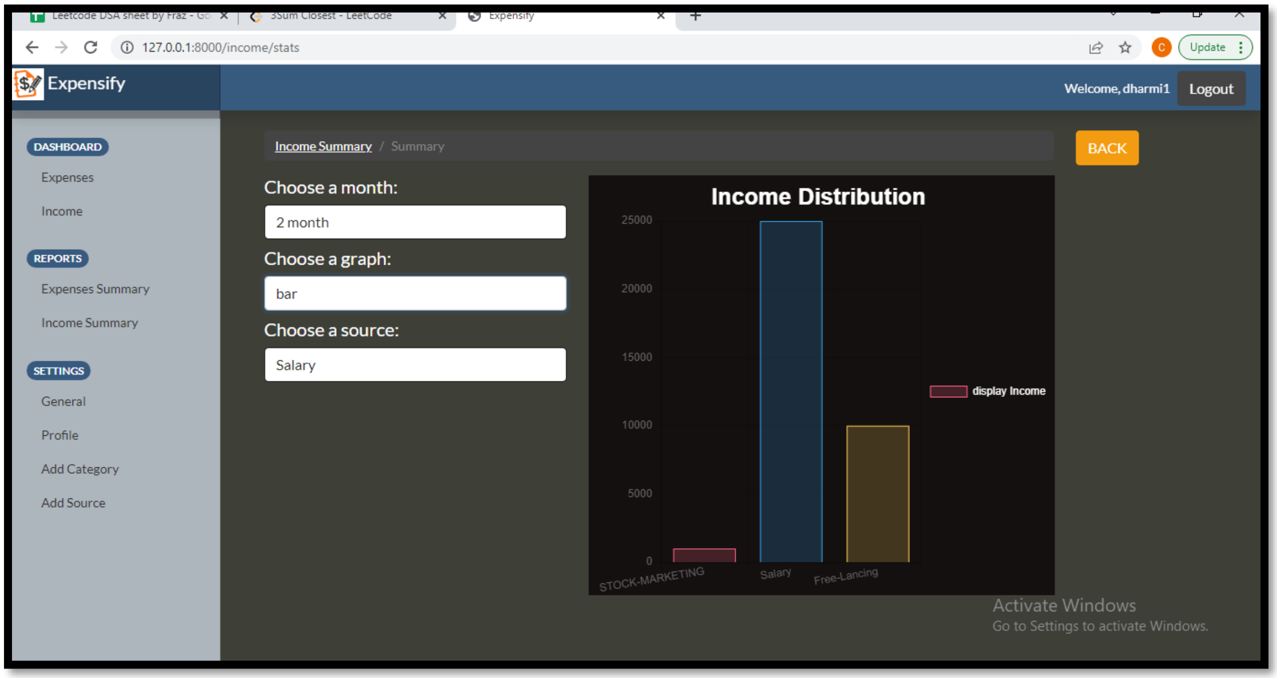
Expenses Summary / Summary

BACK

Choose a month: 4 month

Choose a graph: doughnut

Choose a Category: SHOPPING


127.0.0.1:8000/income/stats

Expensify

Welcome, dharmi1 Logout

DASHBOARD
Expenses
Income
REPORTS
Expenses Summary
Income Summary
SETTINGS
General
Profile
Add Category
Add Source

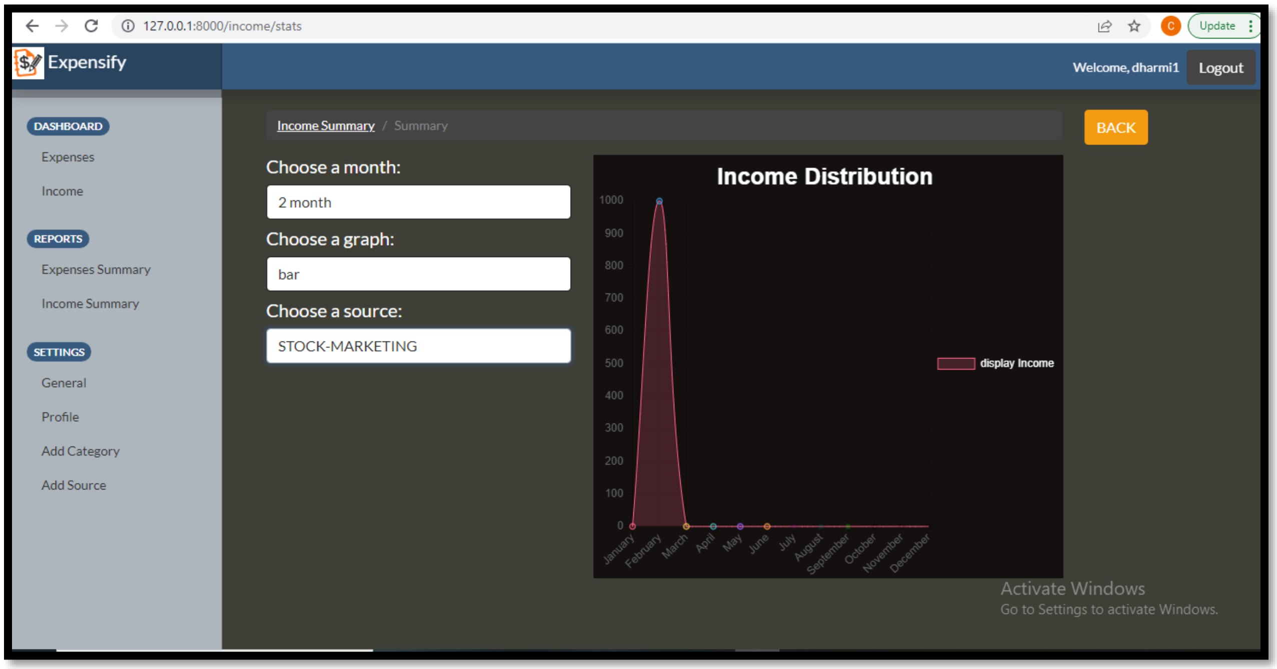
Income Summary / Summary

BACK

Choose a month: 2 month

Choose a graph: bar

Choose a source: Salary


127.0.0.1:8000/income/stats

Choose a month: 2 month

Choose a graph: bar

Choose a source: STOCK-MARKETING

## 8)<u>Conclusion</u>

The functionalities that are implemented in the system are prepared after understanding all functionalities according to software requirements specifications (SRS). Functionalities that are successfully implemented in
the system are as follows:

- Login
- Registration
- User authentication
- Logout
- Add Expense/Income record
- Edit Expense/Income record
- Delete Expense/Income record
- Search Expense/Income record
- View Statistic Expense/Income record
- Export Excel Expense/Income record file
- Expense/Income Report
- Add Source/Category
- Edit profile

After implementing all these functionalities, comprehensive testing was performed on the system to determine possible errors.

## 9)Limitations and future extensions

**Limitations:**

- This project is suitable for small scale organization
- When type of category/source is very high then using graphs see the summary is Difficult.

**Future extensions:**

- We implement Functionality that convert rupees to other currency so is it use for any user.

## 10)Bibliography

Following links and websites were referred during the development of this project:

- www.docs.djangoproject.com
- https://www.chartjs.org/
- https://github.com/
- https://stackoverflow.com/
- https://www.w3schools.com/
- https://getbootstrap.com/docs/4.0
- http://www.umletino.com/
- https://www.python.org/doc/