

url.py

```
from django.contrib import admin
from django.urls import path

from django.contrib import admin
from django.urls import path
from pft import views
from django.views.generic import TemplateView

urlpatterns = [
    path('logout/', views.custom_logout, name="logout"),
    path('pdf/', views.pdf, name='pdf'),
    path('admin/', admin.site.urls),
    path('login/', views.login_page, name='login'),
    path('register/', views.register_page, name='register'),
    path("", views.expenses, name='expenses'),
    path('update_expense/<id>', views.update_expense, name='update_expense'),
    path('delete_expense/<id>', views.delete_expense, name='delete_expense'),
    path('ai_suggestion/', views.ai_suggestion, name='ai_suggestion'),
    path('chart_analysis/', views.chart_analysis, name='chart_analysis'),
    path('upload_tr/', TemplateView.as_view(template_name="upload_transaction.html"),
name="upload_tr"),
    path('extract_text/', views.extract_text, name="extract_text"),
]
```

View.py

```
# Import necessary libraries
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth.decorators import login_required
from django.contrib.auth import authenticate, login, logout
from django.contrib import messages
from django.contrib.auth.models import User # Import User model
from .models import Expense
from groq import Groq
from django.views.decorators.csrf import csrf_exempt
from django.http import JsonResponse
from datetime import datetime
from django.utils.timezone import now
from datetime import datetime
from django.db.models import Sum
```

```

import json
import calendar
import os
import matplotlib.pyplot as plt
import seaborn as sns
from django.conf import settings
import os
import pytesseract
from PIL import Image
import cv2
import numpy as np
from datetime import datetime, timedelta
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from django.core.files.storage import default_storage
from .models import Expense
from .forms import TransactionImageForm

# Create Expense page
@login_required(login_url='/login/')
def expenses(request):
    user = request.user # Get the logged-in user

    # Get current month and year
    current_month = now().month
    current_year = now().year

    if request.method == 'POST':
        data = request.POST
        name = data.get('name')
        price = int(data.get('price', 0))

        Expense.objects.create(
            user=user,
            name=name,
            price=price,
            added_at=datetime.now() # Store current date and time
        )
        return redirect('/')

    # Fetch all expenses and order by latest
    queryset = Expense.objects.filter(user=user).order_by('-added_at')

    # Get salary for the current month (assuming salary is stored per month)

```

```

    salary_entry = Expense.objects.filter(user=user, added_at__month=current_month,
added_at__year=current_year).first()
    current_month_salary = salary_entry.salary if salary_entry else 0

# Calculate total expenses for the current month
total_monthly_expense = Expense.objects.filter(
    user=user,
    added_at__month=current_month,
    added_at__year=current_year
).aggregate(Sum('price'))['price__sum'] or 0 # Default to 0 if no expenses

# Set status color based on salary vs. expense
expense_status = "red" if total_monthly_expense > current_month_salary else "green"

context = {
    'expenses': queryset,
    'total_sum': total_monthly_expense,
    'current_month_salary': current_month_salary,
    'expense_status': expense_status
}
return render(request, 'expenses.html', context)

```

```

# Update the Expenses data
@login_required(login_url='/login/')
def update_expense(request, id):
    queryset = get_object_or_404(Expense, id=id)

    if request.method == 'POST':
        data = request.POST
        name = data.get('name')
        price = int(data.get('price', 0))

        queryset.name = name
        queryset.price = price
        queryset.added_at = datetime.now() # Update the timestamp when modified
        queryset.save()
        return redirect('/')

    context = {'expense': queryset}
    return render(request, 'update_expense.html', context)

```

```

# Delete the Expenses data

```

```

@login_required(login_url='/login/')
def delete_expense(request, id):
    queryset = Expense.objects.get(id=id)
    queryset.delete()
    return redirect('/')

# Login page for user
def login_page(request):
    if request.method == "POST":
        try:
            username = request.POST.get('username')
            password = request.POST.get('password')
            user_obj = User.objects.filter(username=username).first()
            if not user_obj:
                messages.error(request, "Username not found")
                return redirect('/login/')
            user_auth = authenticate(username=username, password=password)
            if user_auth:
                login(request, user_auth)
                return redirect('expenses')
            messages.error(request, "Wrong Password")
            return redirect('/login/')
        except Exception as e:
            messages.error(request, "Something went wrong")
            return redirect('/register/')
    return render(request, "login.html")

# Register page for user
def register_page(request):
    if request.method == "POST":
        try:
            username = request.POST.get('username')
            password = request.POST.get('password')
            salary = request.POST.get('salary')

            if User.objects.filter(username=username).exists():
                messages.error(request, "Username is taken")
                return redirect('/register/')

            user_obj = User.objects.create(username=username)
            user_obj.set_password(password)
            user_obj.save()

            Expense.objects.create(user=user_obj, salary=int(salary))

```

```

        messages.success(request, "Account created successfully")
        return redirect('/login')

    except Exception as e:
        messages.error(request, "Something went wrong")
        return redirect('/register')

    return render(request, "register.html")

# Logout function
def custom_logout(request):
    logout(request)
    return redirect('login')

# Generate the Bill
@login_required(login_url='/login/')
def pdf(request):
    if request.method == 'POST':
        data = request.POST
        salary = int(data.get('salary'))
        name = data.get('name')
        price = int(data.get('price', 0))

        Expense.objects.create(
            salary=salary,
            name=name,
            price=price,
        )
        return redirect('pdf')

    queryset = Expense.objects.all()
    if request.GET.get('search'):
        queryset = queryset.filter(
            name__icontains=request.GET.get('search'))

    # Calculate the total sum
    total_sum = sum(expense.price for expense in queryset)
    # Get the username
    username = request.user.username

    context = {'expenses': queryset, 'total_sum': total_sum, 'username': username}
    return render(request, 'pdf.html', context)

# Your Groq API Key

```

```
GROQ_API_KEY =  
"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"
```

```
@login_required(login_url='/login/')  
def ai_suggestion(request):  
    # Fetch current user's expenses for the current month  
    user = request.user  
    current_month = datetime.now().month  
    last_month = 12 if current_month == 1 else current_month - 1  
    current_year = datetime.now().year  
  
    expenses = Expense.objects.filter(user=user, added_at__month=last_month,  
    added_at__year=current_year)  
  
    if not expenses.exists():  
        return render(request, 'ai_suggestions.html', {'suggestion': "<p>No expenses found for last  
month.</p>"})  
  
    # Get user's salary  
    salary = user.salary if hasattr(user, 'salary') else 0  
  
    # Convert expenses into a structured format  
    expense_data = "".join([  
        f"<tr><td>{exp.name}</td><td>₹{exp.price}</td></tr>" for exp in expenses  
    ])  
  
    # Get the current month name  
    last_month_date = datetime.now().replace(day=1) - timedelta(days=1)  
    last_month_name = last_month_date.strftime("%B %Y")  
  
    # Prompt AI to respond in HTML format  
    prompt = f"""  
    Analyze my financial data and provide insights in a structured HTML format. Use bullet  
    points, percentages, and highlight problem areas where I exceed safe limits. If my spending is  
    well-managed, congratulate me. Otherwise, suggest areas to reduce expenses. Also, include a  
    separate section with suggestions for next month based on previous month's spending patterns.  
  
    Here's my financial data:  
  
    <h2>Monthly Financial Analysis - {last_month_name}</h2>  
  
    <h3>Income Summary</h3>  
    <p><strong>Monthly Salary:</strong> ₹{salary}</p>
```

Expense Breakdown

| Expense Name | Amount (₹) |
|----------------|------------|
| {expense_data} | |

Insights & Analysis

- Total Expenses:** ₹{{total_expenses}}
- Savings:** ₹{{savings}} ({{savings_percentage}}% of income)
- Major Spending Categories:** {{top_expense_categories}}
- High-Spending Areas:** {{problem_expenses}}

{{performance_evaluation}}

Suggestions for Next Month

- Budget Adjustment:** {{suggested_budget_changes}}
- Reduce Spending on:** {{cost_cutting_areas}}
- Savings Boost Strategy:** {{savings_recommendations}}
- Investment Opportunity:** {{investment_suggestions}}

""

try:

```
client = Groq(api_key=GROQ_API_KEY)
completion = client.chat.completions.create(
    model="mixtral-8x7b-32768",
    messages=[{"role": "user", "content": prompt}],
    temperature=0.7, # Lower randomness for structured output
    max_completion_tokens=2048, # Increase if output is too long
    top_p=1
)
```

```
suggestion = completion.choices[0].message.content.strip() if completion.choices else
```

```
"<p>AI couldn't generate insights.</p>"
```

```
except Exception as e:
```

```
suggestion = f"<p>Error: {str(e)}</p>"
```

```
print(suggestion)
```

```
return render(request, 'ai_suggestions.html', {'suggestion': suggestion, 'current_month_name':
last_month_name})
```

```

@login_required(login_url='/login/')
def chart_analysis(request):
    user = request.user
    current_year = now().year
    current_month = now().month

    # Define absolute path to save images
    plot_dir = os.path.join(settings.BASE_DIR, "static", "analysis_plots")
    os.makedirs(plot_dir, exist_ok=True)

    # Generate and save plots
    monthly_chart_path = os.path.join(plot_dir, "monthly_pie.png")
    yearly_chart_path = os.path.join(plot_dir, "yearly_trend.png")
    category_chart_path = os.path.join(plot_dir, "category_bar.png")

    # If no expenses, clear old charts
    if not Expense.objects.filter(user=user).exists():
        for img in [monthly_chart_path, yearly_chart_path, category_chart_path]:
            if os.path.exists(img):
                os.remove(img)

    # Check again if expenses exist
    monthly_expenses = Expense.objects.filter(user=user, added_at__year=current_year,
added_at__month=current_month)
    yearly_expenses = Expense.objects.filter(user=user, added_at__year=current_year)

    # ---- Monthly Pie Chart ----
    if monthly_expenses.exists():
        plt.figure(figsize=(6, 4))
        sns.set_style("whitegrid")
        data = monthly_expenses.values_list("name", "price")
        labels, values = zip(*data)
        plt.pie(values, labels=labels, autopct='%1.1f%%', startangle=90,
colors=sns.color_palette("coolwarm"))
        plt.title("Monthly Expenses Breakdown")
        plt.savefig(monthly_chart_path)
        plt.close()

    # ---- Yearly Line Chart ----
    if yearly_expenses.exists():
        plt.figure(figsize=(6, 4))
        sns.set_style("whitegrid")

```



```

monthly_totals = {m: 0 for m in range(1, 13)}
for exp in yearly_expenses:
    monthly_totals[exp.added_at.month] += exp.price

plt.plot(list(monthly_totals.keys()), list(monthly_totals.values()), marker='o', color='red')
plt.xticks(list(monthly_totals.keys()), list(monthly_totals.keys()))
plt.xlabel("Month")
plt.ylabel("Total Expense")
plt.title("Yearly Expense Trend")
plt.savefig(yearly_chart_path)
plt.close()

# ---- Category Bar Chart ---
if monthly_expenses.exists():
    plt.figure(figsize=(8, 5)) # Larger figure size for better visibility
    sns.set_style("whitegrid")

    # Extract expense names and values
    data = monthly_expenses.values_list("name", "price")
    labels, values = zip(*data)

    # Sort expenses in descending order
    sorted_data = sorted(zip(values, labels), reverse=True)
    values, labels = zip(*sorted_data)

    # Create a horizontal bar chart
    sns.barplot(y=list(labels), x=list(values), palette="viridis")

    plt.xlabel("Amount Spent (₹)", fontsize=12)
    plt.ylabel("Expense Category", fontsize=12)
    plt.title("Category-wise Expenses", fontsize=14)
    plt.xticks(fontsize=10)
    plt.yticks(fontsize=10)

    plt.savefig(category_chart_path, bbox_inches="tight") # Prevent text from getting cut off
    plt.close()

context = {
    "monthly_chart": "/static/analysis_plots/monthly_pie.png",
    "yearly_chart": "/static/analysis_plots/yearly_trend.png",
    "category_chart": "/static/analysis_plots/category_bar.png",
}
return render(request, "analysis.html", context)

```

```
def parse_transactions_groq(extracted_text):
    try:
        prompt = f"""
        Extract transaction details from the following text.
        Return a JSON list with this format:
        [
            {{ "name": "Store Name", "price": 100, "date": "YYYY-MM-DD" }}
        ]
        Ensure correct numerical values and dates. If any value is missing, add a highly relevant
        placeholder.
```

Text:

{extracted_text}

"""

```
client = Groq(api_key=GROQ_API_KEY)
completion = client.chat.completions.create(
    model="mixtral-8x7b-32768",
    messages=[{"role": "user", "content": prompt}],
    temperature=1,
    max_tokens=1024,
    top_p=1
)
```

```
response_text = completion.choices[0].message.content
# print("TR Data *****")
# print(response_text)
```

```
# Ensure we parse the JSON correctly
transactions = json.loads(response_text)
return transactions
```

```
except json.JSONDecodeError as json_err:
    print(f"JSON Error: {json_err}")
    return []
except Exception as e:
    print(f"Groq API Error: {e}")
    return []
```

@csrf_exempt

@login_required(login_url='/login/')

def extract_text(request):

```
    if request.method == "POST" and request.FILES.get("image"):
        try:
```

```

image = Image.open(request.FILES["image"])
extracted_text = pytesseract.image_to_string(image)

transactions = parse_transactions_groq(extracted_text)

# Ensure transactions are in correct format
if not isinstance(transactions, list):
    print("Invalid transactions format:", transactions)
    return JsonResponse({"success": False, "error": "Invalid response from Groq API"})

# Save transactions to Expense model
for trans in transactions:
    try:
        Expense.objects.create(
            user=request.user,
            name=trans.get("name", "Unknown Store"), # Default name
            price=float(trans.get("price", 0)), # Ensure price is a number
            added_at=trans.get("date", datetime.now()) # Default to current date
        )
    except Exception as e:
        print("Error saving transaction:", e)

    return JsonResponse({"success": True, "expenses": extracted_text, "transactions":
transactions})

except Exception as e:
    return JsonResponse({"success": False, "error": str(e)})

return JsonResponse({"success": False, "error": "No image uploaded"})

```

Ai_suggestions.html

```

{% extends "base.html" %}
{% block start %}
<style>
    body {
        background: linear-gradient(to right, #6a11cb, #2575fc);
        min-height: 100vh;
        display: flex;
        align-items: center;
        justify-content: center;
    }

```

```
}
```

```
.ai-container {  
  max-width: 600px;  
  background: #fff;  
  padding: 30px;  
  border-radius: 10px;  
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.2);  
  text-align: center;  
  animation: fadeIn 0.5s ease-in-out;  
}
```

```
.ai-header {  
  font-size: 24px;  
  font-weight: bold;  
  color: #6a11cb;  
  font-family: 'Poppins', sans-serif;  
  margin-bottom: 15px;  
}
```

```
.ai-content {  
  font-size: 16px;  
  font-family: 'Poppins', sans-serif;  
  color: #333;  
  background: #f8f9fa;  
  padding: 15px;  
  border-radius: 8px;  
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
  line-height: 1.6;  
  text-align: left;  
  animation: slideUp 0.5s ease-in-out;  
}
```

```
.btn-container {  
  margin-top: 20px;  
  display: flex;  
  justify-content: center;  
  gap: 10px;  
}
```

```
.btn-custom {  
  padding: 10px 20px;  
  border-radius: 8px;  
  font-size: 16px;
```

```

    font-weight: bold;
    transition: all 0.3s ease-in-out;
}

.btn-primary {
    background-color: #2575fc;
    border: none;
}

.btn-primary:hover {
    background-color: #1a5edb;
}

.btn-refresh {
    background-color: #28a745;
    border: none;
    color: white;
}

.btn-refresh:hover {
    background-color: #218838;
}

@keyframes fadeIn {
    from { opacity: 0; transform: translateY(-20px); }
    to { opacity: 1; transform: translateY(0); }
}

@keyframes slideUp {
    from { opacity: 0; transform: translateY(30px); }
    to { opacity: 1; transform: translateY(0); }
}
</style>

<div class="container d-flex justify-content-center">
  <div class="ai-container">
    <h3 class="ai-header">💡 AI Financial Suggestions for {{ request.user.username }}</h3>
    <p class="text-center">Report for {{ current_month_name }}</p> <!-- ✅ Fixed month
display -->

    <div class="ai-content">
      {{ suggestion|safe }} <!-- ✅ Renders AI HTML properly -->
    </div>
  </div>
</div>

```

```

        <div class="btn-container">
            <a href="{% url 'expenses' %}" class="btn btn-custom btn-primary">⬅ Back to
Expenses</a>
            <button onclick="window.location.reload();" class="btn btn-custom btn-refresh">🔄
Refresh Suggestion</button>
        </div>
    </div>
</div>

```

```
{% endblock %}
```

Analysis.html

```

{% extends "base.html" %}
{% block start %}

```

```

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
<link rel="stylesheet" href="https://cdn.datatables.net/1.11.5/css/jquery.dataTables.min.css">

```

```

<style>
    .text {
        color: green;
        font-weight: bold;
        font-family: 'Times New Roman', Times, serif;
    }
    .ok {
        color: white;
        text-decoration: none;
    }
    .ok:hover {
        color: white;
        text-decoration: none;
    }
</style>

```

```

<div class="container mt-5">
    <div class="d-flex justify-content-between align-items-center">
        <h4 class="text">Expense Tracker</h4>
        <div>
            <button class="btn btn-primary"><a class="ok" href="/">Home</a></button>

```

```

        <button class="btn btn-primary"><a class="ok" href="{% url 'pdf' %}">Total
Expenses</a></button>
        <button class="btn btn-primary"><a class="ok" href="{% url 'ai_suggestion' %}">AI
Suggestion</a></button>
        <button class="btn btn-primary"><a class="ok" href="{% url 'ai_suggestion' %}">Future
Needs</a></button>
        <button class="btn btn-danger"><a class="ok" href="{% url 'logout'
%}">Logout</a></button>
    </div>
</div>

```

```

<hr>

```

```

<h2 class="text-center" style="color: blue"><img alt="Financial Analysis Dashboard" data-bbox="480 328 505 345"/> Financial Analysis Dashboard</h2>

```

```

<div class="row">
    <div class="col-md-6">
        <div class="card shadow p-3">
            <h4 class="text-center">Monthly Expenses Breakdown</h4>
            
        </div>
    </div>

    <div class="col-md-6">
        <div class="card shadow p-3">
            <h4 class="text-center">Yearly Expense Trend</h4>
            
        </div>
    </div>
</div>

<div class="row mt-4">
    <div class="col-md-12">
        <div class="card shadow p-3">
            <h4 class="text-center">Category-wise Expense Distribution</h4>
            
        </div>
    </div>
</div>
</div>

```

```
{% endblock %}
```

Base.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>{{page}}</title>
```

```
<style>
```

```
  table {
```

```
    width: 80%;
```

```
    margin: 20px auto;
```

```
    border-collapse: collapse;
```

```
  }
```

```
  th,
```

```
  td {
```

```
    padding: 10px;
```

```
    text-align: left;
```

```
    border: 1px solid #ccc;
```

```
  }
```

```
  th {
```

```
    background-color: #f2f2f2;
```

```
  }
```

```
  tr:nth-child(even) {
```

```
    background-color: #f2f2f2;
```

```
  }
```

```
  tr:hover {
```

```
    background-color: #ddd;
```

```
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  {% block start %}
```

```
  {% endblock %}
```



```

<script>
    console.log('Hey Django')
</script>
</body>

</html>

```

Expenses.html

```

{% extends "base.html" %}
{% block start %}

```

```

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
<link rel="stylesheet" href="https://cdn.datatables.net/1.11.5/css/jquery.dataTables.min.css">

```

```

<style>
    .text {
        color: green;
        font-weight: bold;
        font-family: 'Times New Roman', Times, serif;
    }
    .ok {
        color: white;
        text-decoration: none;
    }
    .ok:hover {
        color: white;
        text-decoration: none;
    }
</style>

```

```

<div class="container mt-5">
    <div class="d-flex justify-content-between align-items-center">
        <h4 class="text">Expense Tracker</h4>
        <div>
            <button class="btn btn-primary"><a class="ok" href="{% url 'pdf' %}">Total
Expenses</a></button>
            <button class="btn btn-primary"><a class="ok" href="{% url 'ai_suggestion' %}">AI
Suggestion</a></button>
            <button class="btn btn-primary"><a class="ok" href="{% url 'ai_suggestion' %}">Future
Needs</a></button>

```

```

        <button class="btn btn-primary"><a class="ok" href="{% url 'chart_analysis'
{%}">Expense Analysis</a></button>
        <button class="btn btn-danger"><a class="ok" href="{% url 'logout'
{%}">Logout</a></button>
    </div>
</div>
<hr>
<div class="row">
    <div class="col-md-5">
        <form class="card p-3 shadow-lg" method="post" enctype="multipart/form-data">
            {% csrf_token %}
            <h5>Enter Your Expenses</h5>
            <hr>
            <div class="form-group">
                <label>Expenses Reason</label>
                <input type="text" name="name" class="form-control" required>
            </div>
            <div class="form-group">
                <label>Amount</label>
                <input name="price" type="number" class="form-control" required>
            </div>
            <button type="submit" class="btn btn-success">Add Data</button>
            <hr>
            <center><a href="/upload_tr" class="register-link text-primary">Upload
Image</a></center>
        </form>
    </div>
    <style type="text/css">
        #expenseTable th{
            background: blue;
            color: white;
        }
    </style>
    <div class="col-md-7">
        <div style="display: flex;
flex-direction: row;
justify-content: space-around;
align-items: center;">
            <h5 style="color: black;">Salary: ₹{{ current_month_salary }}</h5>
            <h5 style="color: {{ expense_status }};">Total Expense (This Month): ₹{{ total_sum
}}</h5>
        </div>

        <table id="expenseTable" class="table table-bordered mt-3">

```

```

<thead>
  <tr>
    <th scope="col">S.No.</th>
    <th scope="col">Expenses Reason</th>
    <th scope="col">Amount</th>
    <th scope="col">Date</th>
    <th scope="col">Actions</th>
  </tr>
</thead>
<tbody>
  {% for expense in expenses %}
  <tr>
    <td>{{ forloop.counter }}</td>
    <td>{{ expense.name }}</td>
    <td>{{ expense.added_at }}</td>
    <td>&#x20B9;{{ expense.price }}</td>
    <td>
      <a href="/update_expense/{{ expense.id }}" class="">  </a>
      <a href="/delete_expense/{{ expense.id }}" class="">  </a>
    </td>
  </tr>
  {% endfor %}
</tbody>
</table>

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

{% endblock %}
<!-- Include jQuery & DataTables JS -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="https://cdn.datatables.net/1.11.5/js/jquery.dataTables.min.js"></script>

<!-- Initialize DataTable -->
<script>
$(document).ready(function() {
  $('#expenseTable').DataTable({
    "paging": true,      // Enable pagination
    "searching": true,   // Enable search box
    "ordering": true,    // Enable sorting
    "info": true,        // Show info text
    "lengthMenu": [5, 10, 25, 50, 100], // Control number of rows displayed
    "language": {

```

```

        "search": "🔍 Search:",
        "lengthMenu": "Show _MENU_ records per page",
        "info": "Showing _START_ to _END_ of _TOTAL_ records"
    }
});
});
</script>

```

Login.html

```
{% extends 'base.html' %}
```

```

{% block start %}
<!DOCTYPE html>
<html lang="en">

```

```
<head>
```

```
    <meta charset="utf-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
    <link rel="stylesheet"
```

```
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
```

```
    <title>Login - Finance Tracker</title>
```

```
    <style>
```

```
        body {
```

```
            background: linear-gradient(to right, #6a11cb, #2575fc);
```

```
            height: 100vh;
```

```
            display: flex;
```

```
            align-items: center;
```

```
            justify-content: center;
```

```
        }
```

```
        .login-container {
```

```
            background: #fff;
```

```
            padding: 30px;
```

```
            border-radius: 10px;
```

```
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
```

```
            width: 100%;
```

```
            max-width: 400px;
```

```
        }
```

```

.login-container h4 {
    font-family: 'Poppins', sans-serif;
    font-weight: 600;
    text-align: center;
}

.form-control {
    border-radius: 8px;
    padding: 12px;
    border: 1px solid #ccc;
}

.btn-login {
    width: 100%;
    padding: 12px;
    font-size: 16px;
    background-color: #2575fc;
    border: none;
    color: white;
    border-radius: 8px;
    transition: 0.3s;
}

.btn-login:hover {
    background-color: #1a5edb;
}

.register-link {
    text-align: center;
    display: block;
    margin-top: 10px;
}
</style>
</head>

<body>
<div class="login-container">
    {% if messages %}
    {% for message in messages %}
    <div class="alert alert-success {{ message.tags }}" role="alert">
        {{ message }}
    </div>
    {% endfor %}
    {% endif %}

```

```

<h4>Login</h4>
<form action="" method="post">
  {% csrf_token %}
  <div class="form-group mt-3">
    <label for="username"><i class="fas fa-user"></i> Username</label>
    <input type="text" class="form-control" name="username" placeholder="Enter
username" required>
  </div>
  <div class="form-group mt-3">
    <label for="password"><i class="fas fa-lock"></i> Password</label>
    <input type="password" class="form-control" name="password" placeholder="Enter
password" required>
  </div>
  <div class="form-group mt-4">
    <button type="submit" class="btn btn-login">Login</button>
  </div>
</form>
<a href="{% url 'register' %}" class="register-link text-primary">Create an Account</a>
</div>
</body>

</html>
{% endblock %}

```

Pdf.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Expense</title>

  <!-- Add Bootstrap CSS Link -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/css/bootstrap.min.css"
rel="stylesheet">
  <style>
    body {
      font-family: 'Times New Roman', Times, serif;
    }

```

```
.container {  
    max-width: 700px;  
    margin: 30px auto;  
}
```

```
.expense-container {  
    padding: 20px;  
    border: 1px solid #000;  
    border-radius: 5px;  
    background-color: #fff;  
    margin-top: 30px;  
}
```

```
.expense-header {  
    text-align: center;  
}
```

```
.expense-title {  
    font-size: 24px;  
    color: #333;  
    margin-top: 20px;  
}
```

```
.expense-table {  
    width: 100%;  
    margin-top: 20px;  
    border-collapse: collapse;  
}
```

```
.expense-table th,  
.expense-table td {  
    border: 1px solid #ddd;  
    padding: 12px;  
    text-align: left;  
}
```

```
.expense-total {  
    font-size: 20px;  
    font-weight: bold;  
    color: #333;  
    margin-top: 20px;  
}
```

```
.expense-footer {
```

```

        margin-top: 20px;
        font-size: 16px;
        color: #555;
    }

    .btn-print {
        margin-left: 48%;
        margin-top: 2%;
    }

    img {
        width: 140px;
        height: 150px;
    }

    #date {
        margin-left: 120px;
    }
</style>
</head>

<body>

<div class="container expense-container">
    <div class="expense-header">
        
        <div class="expense-title">
            <h4>{{ username }}, Your Finance Record</h4>
        </div>
    </div>
    <br>

    <table class="expense-table">
        <thead>
            <tr>
                <th>Expenses Reason</th>
                <th>Amount</th>
            </tr>
        </thead>
        <tbody>
            {% for expense in expenses %}
            <tr>
                <td>{{ expense.name }}</td>
                <td> &#x20B9;{{ expense.price }}</td>
            </tr>
            </tbody>
        </table>
    </div>
</body>
</html>

```



```

        </tr>
        {% endfor %}
    </tbody>
</table>

<p class="expense-total">Total Expenses: &#x20B9;{{ total_sum }}</p>
<div class="expense-footer">
    <p>Dear {{ username }}, You Expenses is &#x20B9;{{total_sum}}.</p>
</div>
</div>
<!-- Add Bootstrap JS and jQuery -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.min.js"></script>

</body>

</html>

```

Register.html

```

<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"

integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpuuCO
mLASjC" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <title>Register - Job Portal</title>
    <style>
        body {
            background: linear-gradient(to right, #6a11cb, #2575fc);
            height: 100vh;
            display: flex;
            align-items: center;
            justify-content: center;
        }
    </style>

```

```
.login-container {
  background: #fff;
  padding: 30px;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  width: 100%;
  max-width: 400px;
}

.login-container h4 {
  font-family: 'Poppins', sans-serif;
  font-weight: 600;
  text-align: center;
}

.form-control {
  border-radius: 8px;
  padding: 12px;
  border: 1px solid #ccc;
}

.btn-login {
  width: 100%;
  padding: 12px;
  font-size: 16px;
  background-color: #2575fc;
  border: none;
  color: white;
  border-radius: 8px;
  transition: 0.3s;
}

.btn-login:hover {
  background-color: #1a5edb;
}

.register-link {
  text-align: center;
  display: block;
  margin-top: 10px;
}
</style>
</head>
```

```

<body>
  <div class="container">
    <div class="col-md-4 mx-auto">
      <div class="card shadow p-4">
        <h4 class="text-center mb-3" style="font-family: 'Times New Roman', Times,
serif;">Register</h4>
        {% if messages %}
        {% for message in messages %}
        <div class="alert alert-success {{ message.tags }}" role="alert">
          {{ message }}
        </div>
        {% endfor %}
        {% endif %}
        <form action="" method="post">
          {% csrf_token %}
          <div class="form-group mb-3">
            <label for="username"><i class="fas fa-user"></i> Username</label>
            <input type="text" class="form-control" name="username"
placeholder="Username" required>
          </div>
          <div class="form-group mb-3">
            <label for="password"><i class="fas fa-lock"></i> Password</label>
            <input type="password" class="form-control" name="password"
placeholder="Password" required>
          </div>
          <div class="form-group mb-3">
            <label for="salary"><i class="fas fa-inr"></i> Salary</label>
            <input type="number" class="form-control" name="salary" placeholder="Salary">
          </div>
          <button class="btn btn-primary w-100">Register</button>
        </form>
        <p class="text-center mt-3"><a href="{% url 'login' %}">Already have an account? Log
In</a></p>
      </div>
    </div>
  </div>
</body>

</html>

```

Update_expense.html

```
{% extends "base.html" %}
```

{% block start %}

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">

<style>

```
body {  
    background: linear-gradient(to right, #6a11cb, #2575fc);  
    height: 100vh;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
}  
.update-container {  
    background: #fff;  
    padding: 30px;  
    border-radius: 10px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);  
    width: 100%;  
    max-width: 400px;  
}  
.update-container h4 {  
    font-family: 'Poppins', sans-serif;  
    font-weight: 600;  
    text-align: center;  
    color: #333;  
}  
.form-control {  
    border-radius: 8px;  
    padding: 12px;  
    border: 1px solid #ccc;  
    transition: all 0.3s ease-in-out;  
}  
.form-control:focus {  
    border-color: #6a11cb;  
    box-shadow: 0 0 8px rgba(106, 17, 203, 0.3);  
}  
.btn-update {  
    width: 100%;  
    padding: 12px;  
    font-size: 16px;  
    background: #6a11cb;  
    border: none;  
    color: white;  
    border-radius: 8px;
```

```

        transition: 0.3s;
    }
    .btn-update:hover {
        background: #4a0d9e;
    }
</style>

<div class="container">
    <div class="update-container">
        <h4>Update Expense</h4>
        <hr>
        <form method="post" enctype="multipart/form-data">
            {% csrf_token %}
            <div class="form-group">
                <label>Expenses Reason</label>
                <input type="text" name="name" value="{{ expense.name }}" class="form-control"
required>
            </div>
            <div class="form-group">
                <label>Amount</label>
                <input type="number" name="price" value="{{ expense.price }}" class="form-control"
required>
            </div>
            <button type="submit" class="btn btn-update">Update</button>
        </form>
    </div>
</div>

{% endblock %}

```

Upload_transaction.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Extract Transactions</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <style>
        body {
            background-color: #f8f9fa;

```

```

        font-family: Arial, sans-serif;
    }
    .container {
        max-width: 600px;
        margin-top: 20px;
    }
    .table-container {
        margin-top: 20px;
        background: white;
        padding: 20px;
        border-radius: 10px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
</style>
</head>
<body>

<div class="container">
    <!-- Expense Page Button -->
    <a href="/" class="btn btn-primary mb-3">Expense Page</a>

    <div class="card p-3">
        <h4 class="text-center">Upload Receipt</h4>
        <form id="uploadForm" enctype="multipart/form-data">
            <div class="mb-3">
                <input type="file" id="imageInput" class="form-control" accept="image/*">
            </div>
            <button type="submit" class="btn btn-success w-100">Upload</button>
        </form>
    </div>

    <div id="output" class="table-container d-none">
        <h5>Extracted Transactions</h5>
        <table class="table table-striped">
            <thead class="table-dark">
                <tr>
                    <th>Name</th>
                    <th>Price</th>
                    <th>Date</th>
                </tr>
            </thead>
            <tbody id="transactionsTable"></tbody>

        </table>

```

```

        <br>
        <h3 style="color: green">Expense extracted and added successfully</h3>
    </div>
</div>

<script>
$(document).ready(function(){
    $("#uploadForm").submit(function(e){
        e.preventDefault();
        var formData = new FormData();
        formData.append("image", $("#imageInput")[0].files[0]);

        $.ajax({
            url: "/extract_text/",
            type: "POST",
            data: formData,
            processData: false,
            contentType: false,
            success: function(response){
                if(response.success){
                    $("#output").removeClass("d-none");
                    $("#transactionsTable").empty();
                    response.transactions.forEach(function(trans) {
                        $("#transactionsTable").append(`
                            <tr>
                                <td>${trans.name || 'Unknown'}</td>
                                <td>${trans.price !== null ? trans.price : 'N/A'}</td>
                                <td>${trans.date || 'N/A'}</td>
                            </tr>
                        `);
                    });
                } else {
                    $("#output").addClass("d-none");
                    alert("Error: " + response.error);
                }
            }
        });
    });
});
</script>

</body>
</html>

```

