

PAMANTASAN NG LUNGSOD NG VALENZUELA

PF102

EVENT DRIVEN PROGRAMMING

Financeo

System Documentation

Group Name:

BACKLOGS

Members:

Cruz, Kerby (System Developer)

Gonzales, Gabriel Aaron (System Developer)

Suson, Jeriel Samuel (Design & Documentation)

Tato, Adrian (Design & Documentation)

INTRODUCTION

This document outlines the requirements, features, and considerations for the developments of the Personal Finance & Budgeting Website. The goal of the project is to provide users with an intuitive platform to manage their personal finances, track spending, create budgets, and generate insightful reports, all in a secure and user-friendly environment.

FEATURES

1. User Authentication

- 1.1: Users can create a new account with an email, username, and a strong password.
- 1.2: User passwords are securely hashed and salted (handled by Django's auth system).
- 1.3: Users can log in with their credentials.
- 1.4: Users are able to log out.
- 1.5: Users are able to reset their password via a secure email link.

2. Dashboard

- 2.1: Upon login, users are presented with a dashboard summarizing their financial state.
- 2.2: The dashboard displays:
 - Total balance across all accounts.
 - A high-level overview of monthly budget status
 - A list of recent transactions.
 - Quick-add forms for new transactions.

3. Account Management

- 3.1: Users are able to perform CRUD (Create, Read, Update, Delete) operations for their financial accounts.
- 3.2: Account properties includes:
 - Account Name
 - Account Type
 - Current Balance

4. Transaction Management

- 4.1: Users are able to perform CRUD operations for their transactions.

4.2: Transaction properties includes:

- Date
- Amount
- Type ("Income" or "Expense")
- Category (linked from user-defined categories)
- Account (linked from user's accounts)
- Description (optional)

4.3: When a transaction is added, updated, or deleted, the balance of the associated account is automatically recalculated.

5. Category Management

5.1: Users can perform CRUD operations for their income and expense categories.

5.2: Each category has a name and a type ("Income" or "Expense").

6. Budget Management

6.1: Users are able to perform CRUD operations for their monthly budgets.

6.2: The budget defines the spending limit for a specific Category over a specific time

period (e.g., "Groceries" - \$500 - November 2025).

6.3: The system displays the progress against each budget (e.g., "\$250 of \$500 spent").

7. Reporting

7.1: The system provides the reports on spending by category (pie chart).

7.2: The system provides reports on income vs. expense over time (bar chart).

7.3: All reports are filterable by date range.

8. Security

8.1: All user data is private and segregated. The user cannot see, modify, or even know about the existence of data belonging to another user.

- (Implementation: All data models have a mandatory Foreign Key to the User model, and all database queries are filtered by user=request.user.)

8.2: The application uses Django's built-in protections against common vulnerabilities (SQL Injection via the ORM, Cross-Site Scripting (XSS) via templates, and Cross-Site Request Forgery (CSRF) via tokens).

8.3: All web traffic is served over HTTPS (SSL/TLS).

8.4: Application secrets (secret key, database passwords) are not hard-coded and are managed via environment variables.

CODES

Financeo/urls.py

```
Code Blame Raw Copy Download Edit Search

1 from django.contrib import admin
2 from django.urls import path, include
3 from django.conf import settings
4 from django.conf.urls.static import static
5
6 urlpatterns = [
7     path('admin/', admin.site.urls),
8     path('', include('users.urls')),
9     path('dashboard/', include('finance.urls')),
10    path('categories/', include('categories.urls')),
11    path('reports/', include('reports.urls')),
12    path('settings/', include('settings.urls')),
13    path('transactions/', include('transactions.urls')),
14 ]
15
16 if settings.DEBUG:
17     urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

categories/apps.py

```
Code Blame Raw Copy Download Edit Search

1 from django.apps import AppConfig
2
3
4 class CategoriesConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'categories'
```

categories/forms.py

```
Code Blame Raw Copy Download Edit Search

1 from django import forms
2 from finance.models import Category
3
4 class CategoryForm(forms.ModelForm):
5     class Meta:
6         model = Category
7         fields = ['name', 'type']
8         widgets = {
9             'name': forms.TextInput(attrs={
10                 'class': 'w-full px-3 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-blue-500',
11                 'placeholder': 'Category name',
12             }),
13             'type': forms.Select(attrs={
14                 'class': 'w-full px-3 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-blue-500',
15             }),
16         }
17
18     def __init__(self, *args, user=None, **kwargs):
19         super().__init__(*args, **kwargs)
20         self.user = user
21
22     def clean_name(self):
23         name = self.cleaned_data.get('name')
24         type_val = self.cleaned_data.get('type')
25
26         if not name or not type_val:
27             return name
```

Activate Windows
Go to Settings to activate Windows.

```

27         return name
28
29         # Check if category with same name and type exists for this user
30         query = Category.objects.filter(user=self.user, name=name, type=type_val)
31
32         # If updating, exclude the current instance
33         if self.instance.pk:
34             query = query.exclude(pk=self.instance.pk)
35
36         if query.exists():
37             raise forms.ValidationError(
38                 f"You already have a {type_val} category named '{name}'."
39             )
40
41         return name

```

Activate Windows
Go to Settings to activate Windows.

categories/urls.py

```

1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path("", views.category_list, name="categories"),
6      path("create/", views.create_category, name="create_category"),
7      path("<int:category_id>/update/", views.update_category, name="update_category"),
8      path("<int:category_id>/delete/", views.delete_category, name="delete_category"),
9  ]

```

categories/views.py

```

1  from django.shortcuts import render, redirect, get_object_or_404
2  from django.contrib.auth.decorators import login_required
3  from finance.models import Category, TransactionType
4  from .forms import CategoryForm
5
6  @login_required
7  def category_list(request):
8      user = request.user
9      categories = Category.objects.filter(user=user).order_by('type', 'name')
10
11      context = {
12          'categories': categories,
13          'transaction_types': TransactionType.choices,
14      }
15      return render(request, "categories/categories.html", context)
16
17  @login_required
18  def create_category(request):
19      if request.method == 'POST':
20          form = CategoryForm(request.POST, user=request.user)
21          if form.is_valid():
22              category = form.save(commit=False)
23              category.user = request.user
24              category.save()
25              return redirect('categories')
26      else:
27          form = CategoryForm(user=request.user)

```

Activate Windows
Go to Settings to activate Windows.

```

27     form = CategoryForm(user=request.user)
28
29     context = {
30         'form': form,
31         'action': 'Create',
32     }
33     return render(request, 'categories/categories.html', context)
34
35 @login_required
36 def update_category(request, category_id):
37     category = get_object_or_404(Category, id=category_id, user=request.user)
38     if request.method == 'POST':
39         form = CategoryForm(request.POST, instance=category, user=request.user)
40         if form.is_valid():
41             form.save()
42             return redirect('categories')
43     else:
44         form = CategoryForm(instance=category, user=request.user)
45
46     context = {
47         'form': form,
48         'category': category,
49         'action': 'Update',
50     }
51     return render(request, 'categories/categories.html', context)
52
53 @login_required
54 def delete_category(request, category_id):
55     category = get_object_or_404(Category, id=category_id, user=request.user)
56     if request.method == 'POST':

```

Activate Windows
Go to Settings to activate Windows.

```

57     category.delete()
58     return redirect('categories')
59     context = {
60         'category': category,
61     }
62     return render(request, 'categories/categories.html', context)

```

Activate Windows
Go to Settings to activate Windows.

categories/templates/categories/categories.html

```

Code Blame
1  {% load static tailwind_tags %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>{% if action %}{{ action }} Category{% else %}Categories{% endif %}</title>
8      {% tailwind_css %}
9  </head>
10 <body class="bg-[#318f76]">
11     <div class="flex min-h-screen">
12         <!-- Sidebar -->
13         <aside class="w-60 bg-[#304a46] text-gray-300 flex flex-col sticky top-0 h-screen">
14             <div class="flex items-center p-6">
15                 
16                 <span class="font-semibold text-white">{{ user.username }}</span>
17             </div>
18
19             <nav class="flex-1 px-4 py-2 space-y-2">
20                 <a href="{% url 'dashboard' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Dashboard</a>
21                 <a href="{% url 'accounts' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Accounts</a>
22                 <a href="{% url 'transactions' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Transactions</a>
23                 <a href="{% url 'categories' %}" class="block py-2 px-4 rounded bg-[#2a3633] text-white font-medium">Categories</a>
24                 <a href="{% url 'budgets' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Budgets</a>
25                 <a href="{% url 'reports' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Reports</a>
26                 <a href="{% url 'settings' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Settings</a>

```

Activate Windows
Go to Settings to activate Windows.

```

27         </nav>
28
29         <div class="p-4">
30             <a href="{% url 'logout' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Log-out</a>
31         </div>
32     </aside>
33
34     <!-- Main Content -->
35     <main class="flex-1 p-10 overflow-y-auto">
36
37         <!-- Category List View -->
38         {% if not action %}
39         <div class="flex justify-between items-center mb-6">
40             <h1 class="text-3xl font-bold text-white">Categories</h1>
41             <a href="{% url 'create_category' %}" class="px-4 py-2 bg-blue-600 text-white rounded hover:bg-blue-700">+ Add Category</a>
42         </div>
43
44         <!-- Categories List -->
45         <div class="bg-white rounded-lg shadow p-6">
46             <div class="space-y-4">
47                 {% for category in categories %}
48                 <div class="flex items-center justify-between p-4 border border-gray-200 rounded-lg hover:shadow-md transition-shadow">
49                     <div class="flex items-center gap-4 flex-1">
50                         <div class="flex-1">
51                             <h3 class="text-lg font-semibold text-gray-800">{{ category.name }}</h3>
52                         </div>
53                         <span class="px-4 py-2 rounded-full text-sm font-medium {% if category.type == 'INCOME' %}bg-green-100 text-green-800{% else %}bg-red-100 text-red-800{% endif %}">{{ category.get_type_display }}</span>
54                     </div>
55                 </div>
56             </div>

```

```

57
58             <div class="flex gap-3">
59                 <a href="{% url 'update_category' category.id %}" class="ml-3 px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-blue-700 transition-all shadow-md">Update</a>
60                 <a href="{% url 'delete_category' category.id %}" class="px-4 py-2 bg-red-600 text-white rounded-lg hover:bg-red-700 transition-all shadow-md">Delete</a>
61             </div>
62         </div>
63         {% empty %}
64         <div class="text-center text-gray-500 py-8">
65             <p class="text-lg mb-4">No categories found.</p>
66             <a href="{% url 'create_category' %}" class="text-blue-600 hover:underline font-semibold">Create your first category</a>
67         </div>
68     {% endfor %}
69 </div>
70 {% endif %}
71
72 <!-- Create/Edit Category Form -->
73 {% if action %}
74 <div class="max-w-3xl mx-auto">
75     <h1 class="text-4xl font-bold text-white mb-8 drop-shadow-lg">{{ action }} Category</h1>
76
77     <div class="bg-white rounded-xl shadow-2xl overflow-hidden">
78         <div class="p-6">
79             <form method="post">
80                 {% csrf_token %}
81                 <div class="space-y-4">

```

```

87         <div>
88             <label class="block text-sm font-medium text-gray-700 mb-1">Category Name</label>
89             {{ form.name }}
90             {% if form.name.errors %}
91                 <div class="text-red-600 text-sm mt-1">{{ form.name.errors }}</div>
92             {% endif %}
93         </div>
94         <div>
95             <label class="block text-sm font-medium text-gray-700 mb-1">Type</label>
96             {{ form.type }}
97             {% if form.type.errors %}
98                 <div class="text-red-600 text-sm mt-1">{{ form.type.errors }}</div>
99             {% endif %}
100         </div>
101         {% if form.non_field_errors %}
102             <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded">
103                 {{ form.non_field_errors }}
104             </div>
105         {% endif %}
106     </div>
107     <div class="mt-6 flex gap-3">
108         <button type="submit" class="flex-1 px-8 py-4 bg-blue-600 text-white rounded-lg hover:bg-blue-700 font-bold text-lg transition-all duration-200">
109             {{ action }} Category
110         </button>
111         <a href="{% url 'categories' %}" class="flex-1 px-8 py-4 bg-gray-600 text-white rounded-lg hover:bg-gray-700 font-bold text-lg transition-all duration-200">
112             Cancel
113         </a>
114     </div>
115 </form>
116 </div>

```

Activate Windows
Go to Settings to activate Windows.

```

117     </div>
118 </div>
119 {% endif %}
120
121 <!-- Delete Confirmation View -->
122 {% if category and not action %}
123 <div class="max-w-3xl mx-auto">
124     <h1 class="text-4xl font-bold text-white mb-8 drop-shadow-lg">Delete Category</h1>
125
126     <div class="bg-white rounded-xl shadow-2xl overflow-hidden">
127         <!-- Warning Banner -->
128         <div class="bg-gradient-to-r from-red-50 to-orange-50 border-1-4 border-red-500 p-6">
129             <div class="flex items-start">
130                 <div class="flex-shrink-0">
131                     <svg class="h-6 w-6 text-red-500" fill="none" viewBox="0 0 24 24" stroke="currentColor">
132                         <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12 9v2m0 4h.01m-6.938 4h13.856c1.54 0 2.502-1.667 1.732-3.113.768-1.456 0-3.113-1.732-3.113z" />
133                     </svg>
134                 </div>
135                 <div class="ml-3">
136                     <h3 class="text-lg font-semibold text-red-800">Confirm Deletion</h3>
137                     <p class="mt-1 text-sm text-red-700">Are you sure you want to delete this category? This action cannot be undone.</p>
138                 </div>
139             </div>
140         </div>
141
142         <!-- Category Details -->
143         <div class="p-6">
144             <div class="bg-gray-50 rounded-lg p-6 border border-gray-200 mb-6">
145                 <h3 class="text-lg font-bold text-gray-800 mb-4">Category Details</h3>
146                 <div class="space-y-3">

```

Activate Windows
Go to Settings to activate Windows.


```

147         <div class="flex justify-between items-center py-2 border-b border-gray-200">
148             <span class="text-gray-600 font-medium">Name:</span>
149             <span class="text-gray-900 font-semibold">{{ category.name }}</span>
150         </div>
151         <div class="flex justify-between items-center py-2">
152             <span class="text-gray-600 font-medium">Type:</span>
153             <span class="px-3 py-1 rounded-full text-sm font-medium {% if category.type == 'INCOME' %}bg-green-100 text-green-800{% else %}bg-red-100 text-red-800{% endif %}">{{ category.get_type_display }}</span>
154         </div>
155     </div>
156 </div>
157 </div>
158 </div>
159
160     <form method="post" class="mt-8">
161         {% csrf_token %}
162         <div class="grid grid-cols-2 gap-4">
163             <button type="submit" class="px-8 py-4 bg-red-600 text-white rounded-lg hover:bg-red-700 font-bold text-lg transition-all duration-200 shadow">Delete Category</button>
164             <a href="{% url 'categories' %}" class="px-8 py-4 bg-green-600 text-white rounded-lg hover:bg-green-700 font-bold text-lg transition-all duration-200 shadow">Cancel</a>
165         </div>
166     </form>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 {% endif %}
175
176 </main>
177 </div>

```

Activate Windows
Go to Settings to activate Windows.

finance/admin.py

```

1 from django.contrib import admin
2 from .models import Account, Category, Transaction, Budget
3
4 # Register your models here.
5 admin.site.register(Account)
6 admin.site.register(Category)
7 admin.site.register(Transaction)
8 admin.site.register(Budget)

```

finance/apps.py

```

1 from django.apps import AppConfig
2
3
4 class FinanceConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'finance'

```

finance/forms.py

```

1 from django import forms
2 from .models import Account, Budget
3
4 class AccountForm(forms.ModelForm):
5     class Meta:
6         model = Account
7         fields = ('name', 'type', 'current_balance')
8         widgets = {
9             'name': forms.TextInput(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
10             'type': forms.Select(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
11             'current_balance': forms.NumberInput(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
12         }
13
14 class UpdateAccountForm(AccountForm):
15     class Meta(AccountForm.Meta):
16         fields = ('name', 'type', 'current_balance')
17
18 class BudgetForm(forms.ModelForm):
19     class Meta:
20         model = Budget
21         fields = ('category', 'amount', 'start_date', 'end_date')
22         widgets = {
23             'category': forms.Select(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
24             'amount': forms.NumberInput(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
25             'start_date': forms.DateInput(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
26             'end_date': forms.DateInput(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),

```

```

27     }
28
29     class UpdateBudgetForm(BudgetForm):
30         class Meta(BudgetForm.Meta):
31             fields = ['category', 'amount', 'start_date', 'end_date']

```

Activate Windows

finance/models.py

```

1  from django.db import models
2  from django.contrib.auth.models import User
3  from django.db import transaction as db_transaction
4  from decimal import Decimal
5  from django.db.models.signals import post_save
6  from django.dispatch import receiver
7
8  # Create your models here.
9  class TransactionType(models.TextChoices):
10     INCOME = 'INCOME', 'Income'
11     EXPENSE = 'EXPENSE', 'Expense'
12
13  class AccountType(models.TextChoices):
14     CHECKING = 'CHECKING', 'Checking'
15     SAVINGS = 'SAVINGS', 'Savings'
16     CREDIT_CARD = 'CREDIT_CARD', 'Credit Card'
17     CASH = 'CASH', 'Cash'
18
19
20  # --- Models based on ERD ---
21
22  class Profile(models.Model):
23     """
24     Represents a user's profile, with additional information like a profile picture.
25     """
26     user = models.OneToOneField(User, on_delete=models.CASCADE)
27     profile_picture = models.ImageField(upload_to='profile_pics/', default='profile_pics/default.jpg')

```

Activate Windows
Go to Settings to activate Windows.

```

28
29     def __str__(self):
30         return f'{self.user.username} Profile'
31
32     @receiver(post_save, sender=User)
33     def create_or_update_user_profile(sender, instance, created, **kwargs):
34         if created:
35             Profile.objects.create(user=instance)
36             instance.profile.save()
37
38
39  class Account(models.Model):
40     """
41     Represents a user's financial account (e.g., checking, savings).
42     Tied to NFR-1.1: Belongs to one user.
43     """
44     user = models.ForeignKey(
45         User,
46         on_delete=models.CASCADE,
47         related_name="accounts"
48     )
49     name = models.CharField(max_length=100)
50     type = models.CharField(
51         max_length=50,
52         choices=AccountType.choices,
53         default=AccountType.CHECKING
54     )
55     current_balance = models.DecimalField(
56         max_digits=12,
57         decimal_places=2,

```

Activate Windows
Go to Settings to activate Windows.

```

58         default=0.00
59     )
60
61     def __str__(self):
62         return f"{self.name} ({self.get_type_display()}) - {self.user.username}"
63
64     class Meta:
65         # A user cannot have two accounts with the same name
66         unique_together = ('user', 'name')
67
68
69     class Category(models.Model):
70         """
71         A user-defined category for transactions (e.g., "Groceries", "Salary").
72         Tied to NFR-1.1: Belongs to one user.
73         """
74         user = models.ForeignKey(
75             User,
76             on_delete=models.CASCADE,
77             related_name="categories"
78         )
79         name = models.CharField(max_length=100)
80         type = models.CharField(
81             max_length=10,
82             choices=TransactionType.choices,
83             default=TransactionType.EXPENSE
84         )
85
86     def __str__(self):

```

Activate Windows
Go to Settings to activate Windows.

```

87         return f"{self.name} ({self.get_type_display()})"
88
89     class Meta:
90         verbose_name_plural = "Categories"
91         # A user cannot have two categories with the same name and type
92         unique_together = ('user', 'name', 'type')
93
94
95     class Transaction(models.Model):
96         """
97         Represents a single income or expense event.
98         Implements FR-4.3: Automatically updates Account balance on save/delete.
99         """
100         user = models.ForeignKey(
101             User,
102             on_delete=models.CASCADE,
103             related_name="transactions"
104         )
105         account = models.ForeignKey(
106             Account,
107             on_delete=models.CASCADE,
108             related_name="transactions"
109         )
110         category = models.ForeignKey(
111             Category,
112             on_delete=models.PROTECT, # Don't delete category if transactions use it
113             related_name="transactions"
114         )
115         amount = models.DecimalField(max_digits=12, decimal_places=2)

```

Activate Windows
Go to Settings to activate Windows.

```

116     date = models.DateField()
117     type = models.CharField(
118         max_length=10,
119         choices=TransactionType.choices
120     )
121     description = models.CharField(max_length=255, blank=True, null=True)
122
123     def __str__(self):
124         return f"{self.get_type_display()} - {self.amount} on {self.date}"
125
126     def save(self, *args, **kwargs):
127         """
128         Custom save method to implement FR-4.3 (auto-update balance).
129         Wraps the logic in an atomic transaction.
130         """
131         is_new = self.pk is None
132
133         with db_transaction.atomic():
134             if is_new:
135                 # This is a new transaction
136                 if self.type == TransactionType.INCOME:
137                     self.account.current_balance += self.amount
138                 else: # EXPENSE
139                     self.account.current_balance -= self.amount
140             else:
141                 # This is an update to an existing transaction
142                 old_self = Transaction.objects.get(pk=self.pk)
143
144                 # 1. Revert the old transaction from its account

```

Activate Windows
Go to Settings to activate Windows.

```

145         # (Handles case where account itself was changed)
146         if old_self.type == TransactionType.INCOME:
147             old_self.account.current_balance -= old_self.amount
148         else:
149             old_self.account.current_balance += old_self.amount
150         old_self.account.save()
151
152         # 2. Apply the new transaction to its account
153         if self.type == TransactionType.INCOME:
154             self.account.current_balance += self.amount
155         else:
156             self.account.current_balance -= self.amount
157
158         self.account.save()
159         super().save(*args, **kwargs) # Save the transaction itself
160
161     def delete(self, *args, **kwargs):
162         """
163         Custom delete method to implement FR-4.3 (auto-update balance).
164         Reverts the transaction's effect on the account balance.
165         """
166         with db_transaction.atomic():
167             if self.type == TransactionType.INCOME:
168                 self.account.current_balance -= self.amount
169             else: # EXPENSE
170                 self.account.current_balance += self.amount
171             self.account.save()
172             super().delete(*args, **kwargs) # Delete the transaction itself
173

```

Activate Windows
Go to Settings to activate Windows.

```

174     class Meta:
175         ordering = ['-date', '-id']
176
177
178     class Budget(models.Model):
179         """
180         A user-defined budget for a specific category over a time period.
181         Tied to NFR-1.1: Belongs to one user.
182         """
183         user = models.ForeignKey(
184             User,
185             on_delete=models.CASCADE,
186             related_name="budgets"
187         )
188         category = models.ForeignKey(
189             Category,
190             on_delete=models.CASCADE, # If category is deleted, budget is too
191             related_name="budgets"
192         )
193         amount = models.DecimalField(max_digits=12, decimal_places=2)
194         start_date = models.DateField()
195         end_date = models.DateField()
196
197         def __str__(self):
198             return f"Budget for {self.category.name} ({self.start_date} to {self.end_date})"
199
200         class Meta:
201             # A user cannot have overlapping budgets for the same category
202             unique_together = ('user', 'category', 'start_date', 'end_date')

```

Activate Windows
Go to Settings to activate Windows.

finance/urls.py

```

1     from django.urls import path
2     from . import views
3
4     urlpatterns = [
5         path('', views.dashboard_view, name='dashboard'),
6         path('accounts/', views.accounts_view, name='accounts'),
7         path('accounts/update/<int:account_id>', views.update_account, name='update_account'),
8         path('budgets/', views.budgets_view, name='budgets'),
9         path('budgets/update/<int:budget_id>', views.update_budget, name='update_budget'),
10    ]

```

finance/views.py

```

1     from django.shortcuts import render, redirect, get_object_or_404
2     from django.contrib.auth.decorators import login_required
3     from django.db.models import Sum, F
4     from finance.models import Transaction, Account, Category, AccountType, Budget
5     from .forms import AccountForm, UpdateAccountForm, BudgetForm, UpdateBudgetForm
6     from django.utils import timezone
7     from datetime import timedelta
8
9     # Create your views here.
10    @login_required
11    def dashboard_view(request):
12        # Get the current user
13        user = request.user
14
15        # Define the time period for the report (e.g., last 30 days)
16        end_date = timezone.now().date()
17        start_date = end_date - timedelta(days=30)
18
19        # Filter transactions for the current user and the last 30 days
20        transactions = Transaction.objects.filter(user=user, date__range=[start_date, end_date])
21
22        # --- 1. Income vs. Expense Summary ---
23        income_total = transactions.filter(type='INCOME').aggregate(total=Sum('amount'))['total'] or 0
24        expense_total = transactions.filter(type='EXPENSE').aggregate(total=Sum('amount'))['total'] or 0
25
26        # --- 2. Top 5 Expense Categories ---
27        top_expense_categories = (

```

Activate Windows
Go to Settings to activate Windows.

```

28     transactions.filter(type='EXPENSE')
29     .values('category__name')
30     .annotate(total_spent=Sum('amount'))
31     .order_by('-total_spent')[:5]
32 )
33
34 # --- 3. Account Balances ---
35 accounts = Account.objects.filter(user=user)
36
37 # --- 4. Recent Transactions ---
38 recent_transactions = Transaction.objects.filter(user=user).order_by('-date', '-id')[:10]
39
40 context = {
41     'start_date': start_date,
42     'end_date': end_date,
43     'income_total': income_total,
44     'expense_total': expense_total,
45     'top_expense_categories': top_expense_categories,
46     'accounts': accounts,
47     'recent_transactions': recent_transactions,
48 }
49
50 return render(request, 'finance/dashboard.html', context)
51
52 @login_required
53 def accounts_view(request):
54     if request.method == 'POST':
55         action = request.POST.get('action')
56         if action == 'create':
57             form = AccountForm(request.POST)

```

Activate Windows
Go to Settings to activate Windows.

```

58         if form.is_valid():
59             account = form.save(commit=False)
60             account.user = request.user
61             account.save()
62         elif action == 'delete':
63             account_id = request.POST.get('account_id')
64             Account.objects.filter(id=account_id, user=request.user).delete()
65         elif action == 'update':
66             account_id = request.POST.get('account_id')
67             return redirect('update_account', account_id=account_id)
68         return redirect('accounts')
69
70 accounts = Account.objects.filter(user=request.user)
71 form = AccountForm()
72
73 context = {
74     'accounts': accounts,
75     'form': form,
76 }
77 return render(request, 'finance/accounts.html', context)
78
79 @login_required
80 def update_account(request, account_id):
81     account = get_object_or_404(Account, id=account_id, user=request.user)
82     if request.method == 'POST':
83         form = UpdateAccountForm(request.POST, instance=account)
84         if form.is_valid():
85             form.save()
86             return redirect('accounts')
87     else:

```

Activate Windows
Go to Settings to activate Windows.

```

88         form = UpdateAccountForm(instance=account)
89         return render(request, 'finance/update_account.html', {'form': form, 'account': account})
90
91     @login_required
92     def account_transactions_view(request, account_id):
93         account = get_object_or_404(Account, id=account_id, user=request.user)
94         transactions = Transaction.objects.filter(account=account, user=request.user).order_by('-date')
95         context = {
96             'account': account,
97             'transactions': transactions,
98         }
99         return render(request, 'finance/account_transactions.html', context)
100
101     @login_required
102     def budgets_view(request):
103         if request.method == 'POST':
104             action = request.POST.get('action')
105             if action == 'create':
106                 form = BudgetForm(request.POST)
107                 if form.is_valid():
108                     budget = form.save(commit=False)
109                     budget.user = request.user
110                     budget.save()
111             elif action == 'delete':
112                 budget_id = request.POST.get('budget_id')
113                 Budget.objects.filter(id=budget_id, user=request.user).delete()
114             elif action == 'update':
115                 budget_id = request.POST.get('budget_id')
116                 return redirect('update_budget', budget_id=budget_id)
117                 return redirect('budgets')

```

Activate Windows
Go to Settings to activate Windows.

```

118         budgets = Budget.objects.filter(user=request.user)
119         form = BudgetForm()
120         form.fields['category'].queryset = Category.objects.filter(user=request.user)
121
122         context = {
123             'budgets': budgets,
124             'form': form,
125         }
126         return render(request, 'finance/budgets.html', context)
127
128     @login_required
129     def update_budget(request, budget_id):
130         budget = get_object_or_404(Budget, id=budget_id, user=request.user)
131         if request.method == 'POST':
132             form = UpdateBudgetForm(request.POST, instance=budget)
133             if form.is_valid():
134                 form.save()
135                 return redirect('budgets')
136             else:
137                 form = UpdateBudgetForm(instance=budget)
138         return render(request, 'finance/update_budget.html', {'form': form, 'budget': budget})

```

Activate Windows
Go to Settings to activate Windows.

finance/templates/finance/accounts.html

```

1  {% load static tailwind_tags %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Financial Accounts</title>
8      {% tailwind_css %}
9  </head>
10 <body class="bg-[#318f76]">
11
12     <div class="flex min-h-screen">
13         <aside class="w-60 bg-[#3a4a46] text-gray-300 flex flex-col">
14             <div class="flex items-center p-6">
15                 
16                 <span class="font-semibold text-white">{{ user.username }}</span>
17             </div>
18
19             <nav class="flex-1 px-4 py-2 space-y-2">
20                 <a href="{% url 'dashboard' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Dashboard</a>
21                 <a href="{% url 'accounts' %}" class="block py-2 px-4 rounded bg-[#2a3633] text-white font-medium">Accounts</a>
22                 <a href="{% url 'transactions' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Transactions</a>
23                 <a href="{% url 'categories' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Category</a>
24                 <a href="{% url 'budgets' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Budgets</a>
25                 <a href="{% url 'reports' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Reports</a>
26                 <a href="{% url 'settings' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Settings</a>

```

Activate Windows
Go to Settings to activate Windows.

```

27         </nav>
28
29         <div class="p-4">
30             <a href="{% url 'logout' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Log-out</a>
31         </div>
32     </aside>
33
34     <main class="flex-1 p-10">
35         {% block content %}
36         <div class="mb-8">
37             <h2 class="text-2xl font-bold text-white mb-4">Add New Account</h2>
38             <form method="post" class="bg-white p-6 rounded-lg shadow-md">
39                 {% csrf_token %}
40                 <input type="hidden" name="action" value="create">
41                 {{ form.as_p }}
42                 <div class="mt-4">
43                     <button type="submit" class="inline-flex justify-center py-2 px-4 border border-transparent shadow-sm text-sm font-medium rounded-md text-white bg-indigo-600">
44                         Add Account
45                     </button>
46                 </div>
47             </form>
48         </div>
49
50         <div>
51             <h2 class="text-2xl font-bold text-white mb-4">Your Accounts</h2>
52             <div class="bg-white p-6 rounded-lg shadow-md">
53                 <table class="min-w-full divide-y divide-gray-200">
54                     <thead class="bg-gray-50">
55                         <tr>
56                             <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Account Name</th>

```

Activate Windows
Go to Settings to activate Windows.

```

57                             <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Type</th>
58                             <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Balance</th>
59                             <th scope="col" class="relative px-6 py-3">
60                                 <span class="sr-only">Delete</span>
61                             </th>
62                         </tr>
63                     </thead>
64                     <tbody class="bg-white divide-y divide-gray-200">
65                         {% for account in accounts %}
66                         <tr>
67                             <td class="px-6 py-4 whitespace-nowrap text-sm font-medium text-gray-900">{{ account.name }}</td>
68                             <td class="px-6 py-4 whitespace-nowrap text-sm text-gray-500">{{ account.get_type_display }}</td>
69                             <td class="px-6 py-4 whitespace-nowrap text-sm text-gray-500">{{ account.current_balance|floatformat:2 }}</td>
70                             <td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
71                                 <form method="post" class="inline">
72                                     {% csrf_token %}
73                                     <input type="hidden" name="action" value="update">
74                                     <input type="hidden" name="account_id" value="{{ account.id }}">
75                                     <button type="submit" class="text-blue-600 hover:text-blue-900">Update</button>
76                                 </form>
77                             </td>
78                             <td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
79                                 <form method="post" class="inline">
80                                     {% csrf_token %}
81                                     <input type="hidden" name="action" value="delete">
82                                     <input type="hidden" name="account_id" value="{{ account.id }}">
83                                     <button type="submit" class="text-red-600 hover:text-red-900">Delete</button>
84                                 </form>
85                             </td>
86                         </tr>

```

Activate Windows
Go to Settings to activate Windows.

```

87                             {% empty %}
88                             <tr>
89                                 <td colspan="4" class="px-6 py-4 whitespace-nowrap text-sm text-gray-500 text-center">No accounts found.</td>
90                             </tr>
91                         {% endfor %}
92                     </tbody>
93                 </table>
94             </div>
95         </div>
96     {% endblock %}
97 </main>
98 </div>
99
100 </body>
101 </html>

```

Activate Windows

finance/templates/finance/budgets.html

```

1  {% load static tailwind_tags %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Budgets</title>
8      {% tailwind_css %}
9  </head>
10 <body class="bg-[#318f76]">
11
12     <div class="flex min-h-screen">
13         <aside class="w-60 bg-[#3a4a46] text-gray-300 flex flex-col">
14             <div class="flex items-center p-6">
15                 
16                 <span class="font-semibold text-white">{{ user.username }}

```

Activate Windows
Go to Settings to activate Windows.

```

27     </nav>
28
29     <div class="p-4">
30         <a href="{% url 'logout' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Log-out</a>
31     </div>
32 </aside>
33
34 <main class="flex-1 p-10">
35     {% block content %}
36         <div class="mb-8">
37             <h2 class="text-2xl font-bold text-white mb-4">Add New Budget</h2>
38             <form method="post" class="bg-white p-6 rounded-lg shadow-md">
39                 {% csrf_token %}
40                 <input type="hidden" name="action" value="create">
41                 {{ form.as_p }}
42                 <div class="mt-4">
43                     <button type="submit" class="inline-flex justify-center py-2 px-4 border border-transparent shadow-sm text-sm font-medium rounded-md text-white bg-[#318f76]">
44                         Add Budget
45                     </button>
46                 </div>
47             </form>
48         </div>
49
50         <div>
51             <h2 class="text-2xl font-bold text-white mb-4">Your Budgets</h2>
52             <div class="bg-white p-6 rounded-lg shadow-md">
53                 <table class="min-w-full divide-y divide-gray-200">
54                     <thead class="bg-gray-50">
55                         <tr>
56                             <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Category</th>

```

Activate Windows
Go to Settings to activate Windows.

```

57         <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Amount</th>
58         <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">Start Date</th>
59         <th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">End Date</th>
60         <th scope="col" class="relative px-6 py-3">
61             <span class="sr-only">Delete</span>
62         </th>
63     </tr>
64 </thead>
65 <tbody class="bg-white divide-y divide-gray-200">
66     {% for budget in budgets %}
67     <tr>
68         <td class="px-6 py-4 whitespace-nowrap text-sm font-medium text-gray-900">{{ budget.category.name }}</td>
69         <td class="px-6 py-4 whitespace-nowrap text-sm text-gray-500">${{ budget.amount|floatformat:2 }}</td>
70         <td class="px-6 py-4 whitespace-nowrap text-sm text-gray-500">{{ budget.start_date }}</td>
71         <td class="px-6 py-4 whitespace-nowrap text-sm text-gray-500">{{ budget.end_date }}</td>
72         <td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
73             <form method="post" class="inline">
74                 {% csrf_token %}
75                 <input type="hidden" name="action" value="update">
76                 <input type="hidden" name="budget_id" value="{{ budget.id }}">
77                 <button type="submit" class="text-blue-600 hover:text-blue-900">Update</button>
78             </form>
79         </td>
80         <td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
81             <form method="post" class="inline">
82                 {% csrf_token %}
83                 <input type="hidden" name="action" value="delete">
84                 <input type="hidden" name="budget_id" value="{{ budget.id }}">
85                 <button type="submit" class="text-red-600 hover:text-red-900">Delete</button>
86             </form>
87         </td>
88     </tr>
89     {% empty %}
90     <tr>
91         <td colspan="5" class="px-6 py-4 whitespace-nowrap text-sm text-gray-500 text-center">No budgets found.</td>
92     </tr>
93     {% endfor %}
94 </tbody>
95 </table>
96 </div>
97 </div>
98 {% endblock %}
99 </main>
100 </div>
101 </body>
102 </html>

```

finance/templates/finance/dashboard.html

```

1     {% load static tailwind_tags %}
2     <!DOCTYPE html>
3     <html lang="en">
4     <head>
5         <meta charset="UTF-8">
6         <meta name="viewport" content="width=device-width, initial-scale=1.0">
7         <title>Financial Dashboard</title>
8         {% tailwind_css %}
9     </head>
10    <body class="bg-[#318f76]">
11
12        <div class="flex min-h-screen">
13            <aside class="w-60 bg-[#3a4a46] text-gray-300 flex flex-col">
14                <div class="flex items-center p-6">
15                    
16                    <span class="font-semibold text-white">{{ user.username }}</span>
17                </div>
18
19                <nav class="flex-1 px-4 py-2 space-y-2">
20                    <a href="{% url 'dashboard' %}" class="block py-2 px-4 rounded bg-[#2a3633] text-white font-medium">Dashboard</a>
21                    <a href="{% url 'accounts' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Accounts</a>
22                    <a href="{% url 'transactions' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Transactions</a>
23                    <a href="{% url 'categories' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Category</a>
24                    <a href="{% url 'budgets' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Budgets</a>
25                    <a href="{% url 'reports' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Reports</a>
26                    <a href="{% url 'settings' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Settings</a>

```

```

27         </nav>
28
29         <div class="p-4">
30             <a href="{% url 'logout' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Log-out</a>
31         </div>
32     </aside>
33
34     <main class="flex-1 p-10">
35         <!-- <h1 class="text-3xl font-bold text-white mb-8">Financial Report {{{ start_date|date:"M d, Y" }} - {{{ end_date|date:"M d, Y" }}}</h1> -->
36
37         <!-- Income vs. Expense Summary -->
38         <div class="grid grid-cols-1 md:grid-cols-2 gap-6 mb-8">
39             <div class="bg-white rounded-lg shadow p-6">
40                 <h2 class="text-lg font-semibold text-gray-700 mb-2">Total Income</h2>
41                 <p class="text-3xl font-bold text-green-500">{{{ income_total|floatformat:2 }}}</p>
42             </div>
43             <div class="bg-white rounded-lg shadow p-6">
44                 <h2 class="text-lg font-semibold text-gray-700 mb-2">Total Expenses</h2>
45                 <p class="text-3xl font-bold text-red-500">{{{ expense_total|floatformat:2 }}}</p>
46             </div>
47         </div>
48
49         <!-- Top Expense Categories -->
50         <div class="bg-white rounded-lg shadow p-6 mb-8">
51             <h2 class="text-lg font-semibold text-gray-700 mb-4">Top 5 Expense Categories</h2>
52             <div class="space-y-4">
53                 {% for category in top_expense_categories %}
54                     <div class="flex items-center">
55                         <span class="w-1/3 text-gray-600">{{{ category.category_name }}}</span>
56                         <div class="w-2/3 bg-gray-200 rounded-full h-6">

```

Activate Windows
Go to Settings to activate Windows.

```

57                 <div class="bg-blue-500 h-6 rounded-full text-white text-sm flex items-center justify-center" style="width: {% widthratio category.total_spent %}>
58                     {{{ category.total_spent|floatformat:2 }}}
59                 </div>
60             </div>
61         </div>
62         {% endfor %}
63     </div>
64 </div>
65
66 <div class="grid grid-cols-1 md:grid-cols-2 gap-6">
67     <!-- Account Balances -->
68     <div class="bg-white rounded-lg shadow p-6">
69         <h2 class="text-lg font-semibold text-gray-700 mb-4">Account Balances</h2>
70         <ul>
71             {% for account in accounts %}
72                 <li class="flex justify-between items-center py-2 border-b">
73                     <span class="text-gray-600">{{{ account.name }}} {{{ account.get_type_display }}}</span>
74                     <span class="font-bold text-gray-800">{{{ account.current_balance|floatformat:2 }}}</span>
75                 </li>
76             {% endfor %}
77         </ul>
78     </div>
79
80     <!-- Recent Transactions -->
81     <div class="bg-white rounded-lg shadow p-6">
82         <h2 class="text-lg font-semibold text-gray-700 mb-4">Recent Transactions</h2>
83         <table class="w-full">
84             <thead>
85                 <tr class="border-b">
86                     <th class="text-left py-2">Date</th>

```

Activate Windows
Go to Settings to activate Windows.

```

87         <th class="text-left py-2">Category</th>
88         <th class="text-right py-2">Amount</th>
89     </tr>
90 </thead>
91 <tbody>
92     {% for transaction in recent_transactions %}
93     <tr class="border-b">
94         <td class="py-2 text-gray-600">{{ transaction.date|date:"M d" }}</td>
95         <td class="py-2 text-gray-600">{{ transaction.category.name }}</td>
96         <td class="py-2 font-bold {% if transaction.type == 'INCOME' %}text-green-500{% else %}text-red-500{% endif %}">
97             ${{ transaction.amount|floatformat:2 }}
98         </td>
99     </tr>
100     {% endfor %}
101 </tbody>
102 </table>
103 </div>
104 </div>
105 </main>
106 </div>
107
108 </body>
109 </html>

```

finance/templates/finance/update_account.html

```

1  {% extends 'finance/accounts.html' %}
2
3  {% block content %}
4  <div class="container mx-auto mt-10">
5      <div class="max-w-md mx-auto bg-white p-8 rounded-lg shadow-md">
6          <h2 class="text-2xl font-bold mb-6">Update Account</h2>
7          <form method="post">
8              {% csrf_token %}
9              {{ form.as_p }}
10             <div class="mt-6 flex justify-end">
11                 <a href="{% url 'accounts' %}" class="mr-4 inline-flex justify-center py-2 px-4 border border-gray-300 rounded-md shadow-sm text-sm font-medium text-gray-700">
12                     Cancel
13                 </a>
14                 <button type="submit" class="inline-flex justify-center py-2 px-4 border border-transparent shadow-sm text-sm font-medium rounded-md text-white bg-indigo-600">
15                     Save Changes
16                 </button>
17             </div>
18         </form>
19     </div>
20 </div>
21 {% endblock %}

```

finance/templates/finance/update_budget.html

```

1  {% extends 'finance/budgets.html' %}
2
3  {% block content %}
4  <div class="container mx-auto mt-10">
5      <div class="max-w-md mx-auto bg-white p-8 rounded-lg shadow-md">
6          <h2 class="text-2xl font-bold mb-6">Update Budget</h2>
7          <form method="post">
8              {% csrf_token %}
9              {{ form.as_p }}
10             <div class="mt-6 flex justify-end">
11                 <a href="{% url 'budgets' %}" class="mr-4 inline-flex justify-center py-2 px-4 border border-gray-300 rounded-md shadow-sm text-sm font-medium text-gray-700">
12                     Cancel
13                 </a>
14                 <button type="submit" class="inline-flex justify-center py-2 px-4 border border-transparent shadow-sm text-sm font-medium rounded-md text-white bg-indigo-600">
15                     Save Changes
16                 </button>
17             </div>
18         </form>
19     </div>
20 </div>
21 {% endblock %}

```

reports/apps.py

```

1  from django.apps import AppConfig
2
3
4  class ReportsConfig(AppConfig):
5      default_auto_field = 'django.db.models.BigAutoField'
6      name = 'reports'

```

reports/urls.py

```

1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('', views.reports_view, name='reports'),
6  ]

```

reports/views.py

```

1  from django.shortcuts import render
2  from django.contrib.auth.decorators import login_required
3  from finance.models import Transaction
4  from django.db.models import Sum
5  import matplotlib.pyplot as plt
6  import io
7  import base64
8
9  @login_required
10 def reports_view(request):
11     user = request.user
12
13     # --- Chart 1: Income vs Expense (Bar Chart) ---
14     income_vs_expense = user.transactions.values('type').annotate(total=Sum('amount'))
15     income = next((item['total'] for item in income_vs_expense if item['type'] == 'INCOME'), 0)
16     expense = next((item['total'] for item in income_vs_expense if item['type'] == 'EXPENSE'), 0)
17
18     plt.figure(figsize=(6, 4))
19     plt.bar(['Income', 'Expense'], [income, expense], color=['#4CAF50', '#F44336'])
20     plt.title('Income vs Expense')
21     plt.ylabel('Amount')
22
23     buffer = io.BytesIO()
24     plt.savefig(buffer, format='png')
25     buffer.seek(0)
26     income_expense_chart = base64.b64encode(buffer.getvalue()).decode('utf-8')
27     plt.close()

```

Activate Windows
Go to Settings to activate Windows.

```

28
29     # --- Chart 2: Expense Breakdown (Pie Chart) ---
30     expense_breakdown = user.transactions.filter(type='EXPENSE').values('category__name').annotate(total=Sum('amount')).order_by('-total')
31
32     expense_labels = [item['category__name'] for item in expense_breakdown]
33     expense_totals = [item['total'] for item in expense_breakdown]
34
35     plt.figure(figsize=(6, 6))
36     if expense_totals:
37         plt.pie(expense_totals, labels=expense_labels, autopct='%1.1f%%', startangle=140)
38         plt.title('Expense Breakdown by Category')
39     else:
40         plt.text(0.5, 0.5, 'No expense data available', ha='center', va='center')
41         plt.title('Expense Breakdown by Category')
42
43     buffer = io.BytesIO()
44     plt.savefig(buffer, format='png')
45     buffer.seek(0)
46     expense_chart = base64.b64encode(buffer.getvalue()).decode('utf-8')
47     plt.close()
48
49     context = {
50         'income_expense_chart': income_expense_chart,
51         'expense_chart': expense_chart,
52     }
53
54     return render(request, 'reports/reports.html', context)

```

Activate Windows
Go to Settings to activate Windows.

```

28
29     <div class="p-4">
30         <a href="{% url 'logout' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Log-out</a>
31     </div>
32 </aside>
33
34 <main class="flex-1 p-10">
35     <div class="grid grid-cols-1 md:grid-cols-2 gap-6">
36         <div class="bg-white p-4 rounded-lg shadow flex flex-col items-center">
37             <h2 class="text-xl font-bold mb-4">Expense Breakdown</h2>
38             
39         </div>
40         <div class="bg-white p-4 rounded-lg shadow flex flex-col items-center">
41             <h2 class="text-xl font-bold mb-4">Income vs Expense</h2>
42             
43         </div>
44     </div>
45 </main>
46 </div>
47
48 </body>
49 </html>

```

Activate Windows
Go to Settings to activate Windows.

reports/templates/reports/reports.html

```

1  {% load static tailwind_tags %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Financial Report</title>
8      {% tailwind_css %}
9  </head>
10 <body class="bg-[#318f76]">
11
12     <div class="flex min-h-screen">
13         <aside class="w-60 bg-[#3a4a46] text-gray-300 flex flex-col">
14             <div class="flex items-center p-6">
15                 
16                 <span class="font-semibold text-white">{{ user.username }}</span>
17             </div>
18
19             <nav class="flex-1 px-4 py-2 space-y-2">
20                 <a href="{% url 'dashboard' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Dashboard</a>
21                 <a href="{% url 'accounts' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Accounts</a>
22                 <a href="{% url 'transactions' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Transactions</a>
23                 <a href="{% url 'categories' %}" class="block py-2 px-4 rounded bg-[#2a3633] text-white font-medium">Category</a>
24                 <a href="{% url 'budgets' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Budgets</a>
25                 <a href="{% url 'reports' %}" class="block py-2 px-4 rounded bg-[#2a3633] text-white font-medium">Reports</a>
26                 <a href="{% url 'settings' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Settings</a>
27             </nav>

```

Activate Windows
Go to Settings to activate Windows.

settings/apps.py

```

1  from django.apps import AppConfig
2
3
4  class SettingsConfig(AppConfig):
5      default_auto_field = 'django.db.models.BigAutoField'
6      name = 'settings'

```

settings/forms.py

```

1  from django import forms
2  from django.contrib.auth.forms import PasswordChangeForm
3  from django.contrib.auth.models import User
4
5  class UsernameUpdateForm(forms.ModelForm):
6      class Meta:
7          model = User
8          fields = ['username']
9          widgets = {
10              'username': forms.TextInput(attrs={'class': 'w-full max-w-sm bg-gray-200 border border-gray-300 rounded py-2 px-3 text-gray-700 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-blue-500'})
11          }
12
13  class ProfilePictureUpdateForm(forms.Form):
14      profile_picture = forms.ImageField()
15
16  class EmailUpdateForm(forms.Form):
17      email = forms.EmailField(
18          widget=forms.EmailInput(attrs={'class': 'w-full max-w-sm bg-gray-200 border border-gray-300 rounded py-2 px-3 text-gray-700 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-blue-500'})
19      )
20
21  class CustomPasswordChangeForm(PasswordChangeForm):
22      old_password = forms.CharField(
23          label="Old password",
24          widget=forms.PasswordInput(attrs={'class': 'w-full bg-gray-200 border border-gray-300 rounded py-2 px-3 text-gray-700 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-blue-500'})
25      )
26      new_password1 = forms.CharField(
27          label="New password",
28          widget=forms.PasswordInput(attrs={'class': 'w-full bg-gray-200 border border-gray-300 rounded py-2 px-3 text-gray-700 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-blue-500'})
29      )
30      new_password2 = forms.CharField(
31          label="Confirm new password",
32          widget=forms.PasswordInput(attrs={'class': 'w-full bg-gray-200 border border-gray-300 rounded py-2 px-3 text-gray-700 placeholder-gray-500 focus:outline-none focus:ring-2 focus:ring-blue-500'})
33      )
34

```

settings/urls.py

```

1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('', views.settings_view, name='settings'),
6  ]

```

settings/views.py

```

1  from django.shortcuts import render, redirect
2  from django.contrib.auth.decorators import login_required
3  from django.contrib import messages
4  from django.contrib.auth import update_session_auth_hash
5  from .forms import EmailUpdateForm, CustomPasswordChangeForm, UsernameUpdateForm, ProfilePictureUpdateForm
6
7  @login_required
8  def settings_view(request):
9      if request.method == 'POST':
10          if 'update_username' in request.POST:
11              username_form = UsernameUpdateForm(request.POST, instance=request.user)
12              if username_form.is_valid():
13                  username_form.save()
14                  messages.success(request, 'Your username has been updated!')
15                  return redirect('settings')
16          else:
17              username_form = UsernameUpdateForm(instance=request.user)
18
19          if 'update_profile_picture' in request.POST:
20              profile_picture_form = ProfilePictureUpdateForm(request.POST, request.FILES)
21              if profile_picture_form.is_valid():
22                  request.user.profile.profile_picture = profile_picture_form.cleaned_data['profile_picture']
23                  request.user.profile.save()
24                  messages.success(request, 'Your profile picture has been updated!')
25                  return redirect('settings')
26          else:
27              profile_picture_form = ProfilePictureUpdateForm()

```

```

28
29     if 'update_email' in request.POST:
30         email_form = EmailUpdateForm(request.POST)
31         if email_form.is_valid():
32             request.user.email = email_form.cleaned_data['email']
33             request.user.save()
34             messages.success(request, 'Your email has been updated!')
35             return redirect('settings')
36         else:
37             email_form = EmailUpdateForm()
38
39     if 'change_password' in request.POST:
40         password_form = CustomPasswordChangeForm(request.user, request.POST)
41         if password_form.is_valid():
42             user = password_form.save()
43             update_session_auth_hash(request, user) # Important!
44             messages.success(request, 'Your password was successfully updated!')
45             return redirect('settings')
46         else:
47             messages.error(request, 'Please correct the error below.')
48     else:
49         password_form = CustomPasswordChangeForm(request.user)
50
51     else:
52         username_form = UsernameUpdateForm(instance=request.user)
53         profile_picture_form = ProfilePictureUpdateForm()
54         email_form = EmailUpdateForm()
55         password_form = CustomPasswordChangeForm(request.user)
56
57     context = {
58         'username_form': username_form,
59
60         'profile_picture_form': profile_picture_form,
61         'email_form': email_form,
62         'password_form': password_form
63     }
64     return render(request, 'settings/settings.html', context)

```

Activate Windows
Go to Settings to activate Windows.

Activate Windows
Go to Settings to activate Windows.

settings/templates/settings/settings.html

```

1  {% load static tailwind_tags %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>User Settings</title>
8      {% tailwind_css %}
9  </head>
10 <body class="bg-[#318f76]">
11
12     <div class="flex min-h-screen">
13         <aside class="w-60 bg-[#3a4a46] text-gray-300 flex flex-col">
14             <div class="flex items-center p-6">
15                 
16                 <span class="font-semibold text-white">{{ user.username }}</span>
17             </div>
18
19             <nav class="flex-1 px-4 py-2 space-y-2">
20                 <a href="{% url 'dashboard' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Dashboard</a>
21                 <a href="{% url 'accounts' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Accounts</a>
22                 <a href="{% url 'transactions' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Transactions</a>
23                 <a href="#" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Category</a>
24                 <a href="{% url 'budgets' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Budgets</a>
25                 <a href="{% url 'reports' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Reports</a>
26                 <a href="{% url 'settings' %}" class="block py-2 px-4 rounded bg-[#2a3633] text-white font-medium">Settings</a>

```

Activate Windows
Go to Settings to activate Windows.


```

57
58     <div class="bg-gray-100 rounded-lg shadow-xl p-8 max-w-2xl">
59
60         <form method="post">
61             {% csrf_token %}
62             <div class="mb-6">
63                 <label class="block text-sm font-bold text-gray-700 mb-1">Username</label>
64                 <div class="flex items-center justify-between space-x-4">
65                     {{ username_form.username }}
66                     <button type="submit" name="update_username" class="font-semibold text-gray-700 underline whitespace-nowrap hover:text-black">Update Username</button>
67                 </div>
68                 {{ username_form.username.errors }}
69             </div>
70         </form>
71
72         <hr class="border-gray-300 my-6">
73
74         <form method="post" enctype="multipart/form-data">
75             {% csrf_token %}
76             <div class="mb-6">
77                 <label class="block text-sm font-bold text-gray-700 mb-1">Profile Picture</label>
78                 <div class="flex items-center justify-between space-x-4">
79                     {{ profile_picture_form.profile_picture }}
80                     <button type="submit" name="update_profile_picture" class="font-semibold text-gray-700 underline whitespace-nowrap hover:text-black">Update Picture</button>
81                 </div>
82                 {{ profile_picture_form.profile_picture.errors }}
83             </div>
84         </form>
85
86         <hr class="border-gray-300 my-6">

```

```

87
88     <form method="post">
89         {% csrf_token %}
90         <div class="mb-6">
91             <label class="block text-sm font-bold text-gray-700 mb-1">E-Mail Address</label>
92             <p class="text-gray-900 mb-3">{{ user.email }}</p>
93
94             <div class="flex items-center justify-between space-x-4">
95                 {{ email_form.email }}
96                 <button type="submit" name="update_email" class="font-semibold text-gray-700 underline whitespace-nowrap hover:text-black">Update E-mail</button>
97             </div>
98             {{ email_form.email.errors }}
99         </div>
100     </form>
101
102     <hr class="border-gray-300 my-6">
103
104     <form method="post">
105         {% csrf_token %}
106         <div>
107             <label class="block text-sm font-bold text-gray-700 mb-1">Password</label>
108             <div class="space-y-4">
109                 {{ password_form.old_password }}
110                 {{ password_form.old_password.errors }}
111                 {{ password_form.new_password1 }}
112                 {{ password_form.new_password1.errors }}
113                 {{ password_form.new_password2 }}
114                 {{ password_form.new_password2.errors }}
115             </div>
116             <div class="flex items-center justify-between mt-4">

```

Activate Windows
Go to Settings to activate Windows.

```

117                 <p class="text-gray-900 tracking-wider text-lg"></p>
118                 <button type="submit" name="change_password" class="font-semibold text-gray-700 underline hover:text-black">Change Password</button>
119             </div>
120         </div>
121     </form>
122
123     </div>
124
125     </main>
126 </div>
127
128 </body>
129 </html>

```

Activate Windows

static/js/main.js

```

1  tailwind.config = {
2    theme: {
3      fontFamily: {
4        montserrat: ["Montserrat", "sans-serif"],
5        gotu: ["Gotu", "sans-serif"],
6      },
7    },
8  };
9
10 /**
11  * Toggles the visibility of a password field
12  * @param {string} id - The ID of the password input field
13  */
14 function togglePasswordVisibility(id) {
15   const passwordInput = document.getElementById(id);
16   const eyeIcon = document.getElementById(`${id}-eye`);
17
18   if (passwordInput.type === "password") {
19     passwordInput.type = "text";
20     eyeIcon.setAttribute("data-feather", "eye-off");
21   } else {
22     passwordInput.type = "password";
23     eyeIcon.setAttribute("data-feather", "eye");
24   }
25   feather.replace();
26 }
27

```

Activate Windows
Go to Settings to activate Windows.

```

28  /**
29   * Shows a toast notification
30   * @param {string} message - The message to display
31   * @param {string} type - The type of notification (success, error, warning, info)
32   * @param {number} duration - How long to display the notification in milliseconds
33   */
34  function showToast(message, type = "info", duration = 3000) {
35    const toast = document.createElement("div");
36    toast.className = `fixed bottom-4 right-4 px-4 py-2 rounded-md text-white shadow-lg ${
37      type === "success"
38        ? "bg-green-500"
39        : type === "error"
40        ? "bg-red-500"
41        : type === "warning"
42        ? "bg-yellow-500"
43        : "bg-blue-500"
44    }`;
45    toast.textContent = message;
46    document.body.appendChild(toast);
47
48    setTimeout(() => {
49      toast.classList.add("opacity-0", "transition-opacity", "duration-300");
50      setTimeout(() => toast.remove(), 300);
51    }, duration);
52  }
53
54  document.addEventListener("DOMContentLoaded", function () {
55    feather.replace();
56    initLoginForm();
57  });

```

Activate Windows
Go to Settings to activate Windows.

```

58
59  function initLoginForm() {
60    const loginForm = document.getElementById("login-form");
61    if (!loginForm) return;
62
63    loginForm.addEventListener("submit", async (event) => {
64      event.preventDefault();
65
66      const submitButton = loginForm.querySelector('button[type="submit"]');
67      setButtonLoading(submitButton, true);
68
69      try {
70        const payload = await submitLoginForm(loginForm);
71
72        if (!payload.success) {
73          showToast(payload.message || "Unable to sign in", "error");
74          return;
75        }
76
77        showToast(payload.message || "Signed in successfully", "success");
78
79        const redirectTarget =
80          payload.redirect || loginForm.dataset.redirect || window.location.href;
81
82        setTimeout(() => {
83          window.location.href = redirectTarget;
84        }, 800);
85      } catch (error) {
86        showToast(error.message || "Something went wrong", "error");
87      } finally {
88        setButtonLoading(submitButton, false);

```

Activate Windows
Go to Settings to activate Windows.

```

89     }
90   });
91 }
92
93 ✓ async function submitLoginForm(form) {
94   const endpoint = form.getAttribute("action") || window.location.href;
95   const formData = new FormData(form);
96
97   const response = await fetch(endpoint, {
98     method: "POST",
99     body: formData,
100     headers: {
101       "X-Requested-With": "XMLHttpRequest",
102       Accept: "application/json",
103     },
104   });
105
106   const responseText = await response.text();
107   let payload;
108
109   try {
110     payload = JSON.parse(responseText);
111   } catch (_error) {
112     payload = {
113       success: response.ok,
114       message: responseText || "Login processed",
115     };
116   }
117
118   payload.success = payload.success ?? response.ok;

```

Activate Windows
Go to Settings to activate Windows.

```

119   return payload;
120 }
121
122 ✓ function setButtonLoading(button, isLoading) {
123   if (!button) return;
124
125   if (isLoading) {
126     button.dataset.originalText = button.textContent;
127     button.textContent = "Signing in...";
128     button.disabled = true;
129     button.classList.add("opacity-75", "cursor-not-allowed");
130   } else {
131     button.textContent = button.dataset.originalText || "Sign in";
132     button.disabled = false;
133     button.classList.remove("opacity-75", "cursor-not-allowed");
134   }
135 }

```

Activate Windows
Go to Settings to activate Windows.

theme/apps.py

```

1   from django.apps import AppConfig
2
3
4   class ThemeConfig(AppConfig):
5       name = 'theme'

```

theme/static_src/src/tailwind.config.js

```

1   /** @type {import('tailwindcss').Config} */
2   module.exports = {
3     content: [
4       '../templates/**/*.html',
5       '../**/templates/**/*.html',
6       './src/**/*.js'
7     ],
8     theme: {
9       extend: {
10        colors: {
11          'custom-green-bg': '#318f76', // Main background green
12          'custom-sidebar': '#3a4a46', // Darker sidebar green-gray
13          'custom-sidebar-active': '#2a3633', // Darkest active item
14        },
15      },
16    },
17    plugins: [],
18  }

```

Activate Windows

theme/templates/base.html

```

1  {% load static tailwind_tags %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <title>Django Tailwind</title>
6      <meta charset="UTF-8">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <meta http-equiv="X-UA-Compatible" content="ie=edge">
9      {% tailwind_css %}
10 </head>
11
12 <body class="bg-gray-50 text-black font-serif leading-normal tracking-normal">
13 <div class="container mx-auto">
14     <section class="flex items-center justify-center h-screen">
15         <h1 class="text-5xl">Django + Tailwind = ❤️</h1>
16     </section>
17 </div>
18 </body>
19 </html>

```

transactions/apps.py

```

1  from django.apps import AppConfig
2
3
4  class TransactionsConfig(AppConfig):
5      default_auto_field = 'django.db.models.BigAutoField'
6      name = 'transactions'

```

transactions/forms.py

```

1  from django import forms
2  from finance.models import Transaction, Account, Category, TransactionType
3
4  class TransactionForm(forms.ModelForm):
5      class Meta:
6          model = Transaction
7          fields = ['account', 'category', 'amount', 'date', 'type', 'description']
8          widgets = {
9              'account': forms.Select(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
10             'category': forms.Select(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
11             'amount': forms.NumberInput(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm', 'type': 'text'}),
12             'date': forms.DateInput(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm', 'type': 'text'}),
13             'type': forms.Select(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'}),
14             'description': forms.TextInput(attrs={'class': 'mt-1 block w-full rounded-md border-gray-300 shadow-sm focus:border-indigo-500 focus:ring-indigo-500 sm:text-sm'})
15         }
16
17     def __init__(self, *args, **kwargs):
18         user = kwargs.pop('user', None)
19         transaction_type = kwargs.pop('transaction_type', None)
20         super().__init__(*args, **kwargs)
21
22         if user:
23             self.fields['account'].queryset = Account.objects.filter(user=user)
24             if transaction_type:
25                 self.fields['category'].queryset = Category.objects.filter(user=user, type=transaction_type)
26             else:
27                 self.fields['category'].queryset = Category.objects.filter(user=user)
28
29     class UpdateTransactionForm(TransactionForm):
30         class Meta(TransactionForm.Meta):
31             fields = ['account', 'category', 'amount', 'date', 'type', 'description']

```

Activate Windows
Go to Settings to activate Windows.

Activate Windows

transactions/urls.py

```

1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path("", views.transaction_list, name="transactions"),
6      path("create/", views.create_transaction, name="create_transaction"),
7      path("<int:transaction_id>/update/", views.update_transaction, name="update_transaction"),
8      path("<int:transaction_id>/delete/", views.delete_transaction, name="delete_transaction"),
9      path("search/", views.search_transactions_ajax, name="search_transactions_ajax"),
10 ]

```

transactions/views.py

```

1  from django.shortcuts import render, redirect, get_object_or_404
2  from django.contrib.auth.decorators import login_required
3  from django.http import JsonResponse
4  from finance.models import Transaction, Account, Category
5  from .forms import TransactionForm, UpdateTransactionForm
6
7  @login_required
8  def transaction_list(request):
9      user = request.user
10     transactions = Transaction.objects.filter(user=user).order_by('-date', '-id')
11
12     # Filter by type if provided
13     transaction_type = request.GET.get('type', '')
14     if transaction_type:
15         transactions = transactions.filter(type=transaction_type)
16
17     # Search functionality
18     search_query = request.GET.get('search', '')
19     if search_query:
20         transactions = transactions.filter(
21             description__icontains=search_query
22         ) | transactions.filter(
23             category_name__icontains=search_query
24         )
25
26     context = {
27         'transactions': transactions,
28         'transaction_type': transaction_type,
29         'search_query': search_query,
30         'show_transactions': True,
31         'action': False,
32         'is_delete': False,
33     }
34     return render(request, "transactions/transactions.html", context)
35
36 @login_required
37 def create_transaction(request):
38     if request.method == 'POST':
39         form = TransactionForm(request.POST, user=request.user)
40         if form.is_valid():
41             transaction = form.save(commit=False)
42             transaction.user = request.user
43             transaction.save()
44             return redirect('transactions')
45     else:
46         transaction_type = request.GET.get('type', '')
47         form = TransactionForm(user=request.user, transaction_type=transaction_type)
48
49     context = {
50         'form': form,
51         'action': 'Create',
52         'show_transactions': False,
53         'is_delete': False,
54     }
55     return render(request, 'transactions/transactions.html', context)

```

Activate Windows
Go to Settings to activate Windows.

Activate Windows
Go to Settings to activate Windows.

```

56
57 @login_required
58 def update_transaction(request, transaction_id):
59     transaction = get_object_or_404(Transaction, id=transaction_id, user=request.user)
60     if request.method == 'POST':
61         form = UpdateTransactionForm(request.POST, instance=transaction, user=request.user)
62         if form.is_valid():
63             form.save()
64             return redirect('transactions')
65         else:
66             form = UpdateTransactionForm(instance=transaction, user=request.user)
67
68     context = {
69         'form': form,
70         'transaction': transaction,
71         'action': 'Update',
72         'show_transactions': False,
73         'is_delete': False,
74     }
75     return render(request, 'transactions/transactions.html', context)
76
77 @login_required
78 def delete_transaction(request, transaction_id):
79     transaction = get_object_or_404(Transaction, id=transaction_id, user=request.user)
80     if request.method == 'POST':
81         transaction.delete()
82         return redirect('transactions')
83     context = {
84         'transaction': transaction,

```

Activate Windows
Go to Settings to activate Windows.

```

85         'show_transactions': False,
86         'action': False,
87         'is_delete': True,
88     }
89     return render(request, 'transactions/transactions.html', context)
90
91 @login_required
92 def search_transactions_ajax(request):
93     """AJAX endpoint for searching transactions"""
94     user = request.user
95     transactions = Transaction.objects.filter(user=user).order_by('-date', '-id')
96
97     # Filter by type if provided
98     transaction_type = request.GET.get('type', '')
99     if transaction_type:
100         transactions = transactions.filter(type=transaction_type)
101
102     # Search functionality
103     search_query = request.GET.get('search', '')
104     if search_query:
105         transactions = transactions.filter(
106             description__icontains=search_query
107         ) | transactions.filter(
108             category__name__icontains=search_query
109         )
110
111     # Render the table rows as HTML
112     html = ''
113     if transactions.exists():
114         for transaction in transactions:

```

Activate Windows
Go to Settings to activate Windows.

```

115         color_class = "text-green-500" if transaction.type == "INCOME" else "text-red-500"
116         sign = '+' if transaction.type == 'INCOME' else '-'
117         date_str = transaction.date.strftime('%b %d, %Y')
118         description = transaction.description or '-'
119
120         html += f'<tr class="border-b hover:bg-gray-50" data-type="{transaction.type}">\n'
121         html += f'<td class="py-2 px-4 text-gray-600">{date_str}</td>\n'
122         html += f'<td class="py-2 px-4 text-gray-600">{transaction.account.name}</td>\n'
123         html += f'<td class="py-2 px-4 text-gray-600">{transaction.category.name}</td>\n'
124         html += f'<td class="py-2 px-4 text-gray-600">{description}</td>\n'
125         html += f'<td class="py-2 px-4 font-bold text-right {color_class}">{sign}${transaction.amount:.2f}</td>\n'
126         html += f'<td class="py-2 px-4 text-center">\n'
127         html += f'<div class="flex gap-3 justify-center items-center">\n'
128         html += f'<a href="/transactions/{transaction.id}/update/" class="px-4 py-2 bg-blue-600 text-white rounded-lg hover:bg-blue-700 transition-all shadow-md hover:sha'
129         html += f'<a href="/transactions/{transaction.id}/delete/" class="px-4 py-2 bg-red-600 text-white rounded-lg hover:bg-red-700 transition-all shadow-md hover:sha'
130         html += f'</div></td></tr>\n'
131     else:
132         html = '<tr><td colspan="6" class="text-center text-gray-500 py-8">No transactions found. <a href="/transactions/create/" class="text-blue-600 hover:underline">Crea
133
134     return JsonResponse({'html': html, 'count': transactions.count()})

```

Activate Windows

transactions/templates/transactions/transactions.html

```

1  {% load static tailwind_tags %}
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>{% if action %}{{ action }} Transaction{% elif show_transactions %}Transactions{% elif is_delete %}Delete Transaction{% endif %}</title>
8      {% tailwind_css %}
9  </head>
10 <body class="bg-[#318f76]">
11     <div class="flex min-h-screen">
12         <!-- Sidebar -->
13         <aside class="w-60 bg-[#3a4a46] text-gray-300 flex flex-col sticky top-0 h-screen">
14             <div class="flex items-center p-6">
15                 
16                 <span class="font-semibold text-white">{{ user.username }}</span>
17             </div>
18
19             <nav class="flex-1 px-4 py-2 space-y-2">
20                 <a href="{% url 'dashboard' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Dashboard</a>
21                 <a href="{% url 'accounts' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Accounts</a>
22                 <a href="{% url 'transactions' %}" class="block py-2 px-4 rounded bg-[#2a3633] text-white font-medium">Transactions</a>
23                 <a href="{% url 'categories' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Category</a>
24                 <a href="{% url 'budgets' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Budgets</a>
25                 <a href="{% url 'reports' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Reports</a>
26                 <a href="{% url 'settings' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Settings</a>
27
28             </nav>
29
30             <div class="p-4">
31                 <a href="{% url 'logout' %}" class="block py-2 px-4 rounded hover:bg-[#2a3633]/50">Log-out</a>
32             </div>
33         </aside>
34
35         <!-- Main Content -->
36         <main class="flex-1 p-10 overflow-y-auto">
37
38             <!-- Transaction List View -->
39             {% if show_transactions %}
40             <div class="flex justify-between items-center mb-6">
41                 <h1 class="text-3xl font-bold text-white">Transactions</h1>
42                 <a href="{% url 'create_transaction' %}" class="px-4 py-2 bg-blue-600 text-white rounded hover:bg-blue-700">Add Transaction</a>
43             </div>
44
45             <!-- Filters -->
46             <div class="bg-white rounded-lg shadow p-4 mb-6">
47                 <div class="flex items-center gap-3">
48                     <input type="text" id="searchInput" value="{{ search_query }}" placeholder="Search by description or category..." class="p-2 rounded w-64 border flex-1">
49                     <button type="button" id="filterAll" class="px-4 py-2 bg-gray-200 rounded hover:bg-gray-300">All</button>
50                     <button type="button" id="filterIncome" class="px-4 py-2 bg-green-200 rounded hover:bg-green-300">Income</button>
51                     <button type="button" id="filterExpense" class="px-4 py-2 bg-red-200 rounded hover:bg-red-300">Expense</button>
52                     <button type="button" id="searchButton" class="px-4 py-2 bg-blue-600 text-white rounded hover:bg-blue-700">Search</button>
53                 </div>
54             </div>
55
56             <!-- Transactions Table -->
57             <div class="bg-white rounded-lg shadow p-6">

```



```

57     <h2 class="text-lg font-semibold text-gray-700 mb-4">All Transactions</h2>
58
59     <div class="overflow-x-auto">
60         <table class="w-full table-auto" id="transactionTable">
61             <thead>
62                 <tr class="border-b">
63                     <th class="text-left py-2 px-4">Date</th>
64                     <th class="text-left py-2 px-4">Account</th>
65                     <th class="text-left py-2 px-4">Category</th>
66                     <th class="text-left py-2 px-4">Description</th>
67                     <th class="text-right py-2 px-4">Amount</th>
68                     <th class="text-center py-2 px-4 w-48">Actions</th>
69                 </tr>
70             </thead>
71             <tbody id="transactionTableBody">
72                 {% for transaction in transactions %}
73                 <tr class="border-b hover:bg-gray-50" data-type="{% transaction.type %}">
74                     <td class="py-2 px-4 text-gray-600 whitespace-nowrap">{{ transaction.date|date:'M d, Y' }}</td>
75                     <td class="py-2 px-4 text-gray-600">{{ transaction.account.name }}</td>
76                     <td class="py-2 px-4 text-gray-600">{{ transaction.category.name }}</td>
77                     <td class="py-2 px-4 text-gray-600">{{ transaction.description|default:'' }}</td>
78                     <td class="py-2 px-4 font-bold text-right whitespace-nowrap {% if transaction.type == 'INCOME' %}text-green-500{% else %}text-red-500{% endif %}">
79                         {% if transaction.type == 'INCOME' %}{% else %}{% endif %}${{ transaction.amount|floatformat:2 }}
80                     </td>
81                     <td class="py-3 px-4 text-center whitespace-nowrap">
82                         <div class="flex gap-2 justify-center items-center">
83                             <a href="{% url 'update_transaction' transaction.id %}" class="inline-block px-4 py-2 bg-red-200 rounded hover:bg-red-400 font-semibold">
84                                 Edit
85                             </a>
86                             <a href="{% url 'delete_transaction' transaction.id %}" class="inline-block px-4 py-2 bg-green-200 rounded hover:bg-green-400 font-semibold">
87                                 Delete
88                             </a>

```

```

87                                 Delete
88                             </a>
89                         </div>
90                     </td>
91                 </tr>
92                 {% empty %}
93                 <tr>
94                     <td colspan="6" class="text-center text-gray-500 py-8">No transactions found. <a href="{% url 'create_transaction' %}" class="text-blue-500">Create Transaction</a>
95                 </tr>
96                 {% endfor %}
97             </tbody>
98         </table>
99     </div>
100 </div>
101 {% endif %}
102
103 <!-- Create/Edit Transaction Form -->
104 {% if action %}
105 <div class="max-w-3xl mx-auto">
106     <h1 class="text-4xl font-bold text-white mb-8 drop-shadow-lg">{{ action }} Transaction</h1>
107
108     <div class="bg-white rounded-xl shadow-2xl overflow-hidden">
109         <div class="p-6">
110             <form method="post">
111                 {% csrf_token %}
112                 <div class="space-y-4">
113                     <div>
114                         <label class="block text-sm font-medium text-gray-700 mb-1">Account</label>
115                         {{ form.account }}
116                         {{ form.account.errors }}

```

```

117         </div>
118         <div>
119             <label class="block text-sm font-medium text-gray-700 mb-1">Type</label>
120             {{ form.type }}
121             {{ form.type.errors }}
122         </div>
123         <div>
124             <label class="block text-sm font-medium text-gray-700 mb-1">Category</label>
125             {{ form.category }}
126             {{ form.category.errors }}
127         </div>
128         <div>
129             <label class="block text-sm font-medium text-gray-700 mb-1">Amount</label>
130             {{ form.amount }}
131             {{ form.amount.errors }}
132         </div>
133         <div>
134             <label class="block text-sm font-medium text-gray-700 mb-1">Date</label>
135             {{ form.date }}
136             {{ form.date.errors }}
137         </div>
138         <div>
139             <label class="block text-sm font-medium text-gray-700 mb-1">Description</label>
140             {{ form.description }}
141             {{ form.description.errors }}
142         </div>
143     </div>
144     <div class="mt-6 flex gap-3">
145         <button type="submit" class="flex-1 px-8 py-4 bg-blue-600 text-white rounded-lg hover:bg-blue-700 font-bold text-lg transition-all duration-200">Transaction
146         {{ action }}

```

```

147     </button>
148     <a href="{% url 'transactions' %}" class="flex-1 px-8 py-4 bg-gray-600 text-white rounded-lg hover:bg-gray-700 font-bold text-lg transition-all duration-200">Cancel
149     </a>
150 </div>
151 </div>
152 </form>
153 </div>
154 </div>
155 </div>
156 {% endif %}
157
158 <!-- Delete Transaction View -->
159 {% if is_delete %}
160 <div class="max-w-3xl mx-auto">
161     <h1 class="text-4xl font-bold text-white mb-8 drop-shadow-lg">Delete Transaction</h1>
162
163     <div class="bg-white rounded-xl shadow-2xl overflow-hidden">
164         <!-- Warning Banner -->
165         <div class="bg-gradient-to-r from-red-50 to-orange-50 border-1.4 border-red-500 p-6">
166             <div class="flex items-start">
167                 <div class="flex-shrink-0">
168                     <svg class="h-6 w-6 text-red-500" fill="none" viewBox="0 0 24 24" stroke="currentColor">
169                         <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M12 9v2m0 4h.01m-6.938 4h13.856c1.54 0 2.502-1.667 1.732-3L13.7 13.7" />
170                     </svg>
171                 </div>
172                 <div class="ml-3">
173                     <h3 class="text-lg font-semibold text-red-800">Confirm Deletion</h3>
174                     <p class="mt-1 text-sm text-red-700">Are you sure you want to delete this transaction? This action cannot be undone.</p>
175                 </div>
176             </div>

```

```

177         </div>
178
179         <!-- Transaction Details -->
180         <div class="p-6">
181             <div class="bg-gray-50 rounded-lg p-6 border border-gray-200">
182                 <h3 class="text-lg font-bold text-gray-800 mb-4 flex items-center">
183                     <svg class="w-5 h-5 mr-2 text-gray-600" fill="none" viewBox="0 0 24 24" stroke="currentColor">
184                         <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9 12h6m-6 4h6m2 2 0 0 12-2v5a2 2 0 0 12-2h5.586a1 1 0 0 1.586 1 0 0 1 0 1.586" />
185                     </svg>
186                     Transaction Details
187                 </h3>
188                 <div class="space-y-3">
189                     <div class="flex justify-between items-center py-2 border-b border-gray-200">
190                         <span class="text-gray-600 font-medium">Date:</span>
191                         <span class="text-gray-900">{{ transaction.date|date:"M d, Y" }}</span>
192                     </div>
193                     <div class="flex justify-between items-center py-2 border-b border-gray-200">
194                         <span class="text-gray-600 font-medium">Account:</span>
195                         <span class="text-gray-900">{{ transaction.account.name }}</span>
196                     </div>
197                     <div class="flex justify-between items-center py-2 border-b border-gray-200">
198                         <span class="text-gray-600 font-medium">Category:</span>
199                         <span class="text-gray-900">{{ transaction.category.name }}</span>
200                     </div>
201                     <div class="flex justify-between items-center py-2 border-b border-gray-200">
202                         <span class="text-gray-600 font-medium">Type:</span>
203                         <span class="px-3 py-1 rounded-full text-sm font-medium {% if transaction.type == 'INCOME' %}bg-green-100 text-green-800{% else %}bg-red-100 text-red-800{% endif %}">
204                             {{ transaction.get_type_display }}
205                         </span>
206                     </div>
207                 </div>
208                 <div class="flex justify-between items-center py-2 border-b border-gray-200">
209                     <span class="text-gray-600 font-medium">Amount:</span>
210                     <span class="text-2xl font-bold {% if transaction.type == 'INCOME' %}text-green-600{% else %}text-red-600{% endif %}">
211                         ${{ transaction.amount|floatformat:2 }}
212                     </span>
213                 </div>
214                 <div class="py-2">
215                     <span class="text-gray-600 font-medium block mb-1">Description:</span>
216                     <p class="text-gray-900 bg-white p-3 rounded border border-gray-200">{{ transaction.description }}</p>
217                 </div>
218                 <div class="text-gray-900 font-medium">{{ transaction.get_type_display }}</div>
219             </div>
220         </div>
221
222         <form method="post" class="mt-8">
223             <div class="flex gap-4">
224                 <button type="submit" class="flex-1 px-8 py-4 bg-red-600 text-white rounded-lg hover:bg-red-700 font-bold text-lg transition-all duration-200 shadow-lg hover:shadow-xl">
225                     Delete Transaction
226                 </button>
227                 <a href="{% url 'transactions' %}" class="flex-1 px-8 py-4 bg-gray-600 text-white rounded-lg hover:bg-gray-700 font-bold text-lg transition-all duration-200 shadow-lg hover:shadow-xl">
228                     Cancel
229                 </a>
230             </div>
231         </form>
232     </div>
233 </div>
234 </div>
235 </div>
236 </div>

```

```

237         {% endif %}
238
239     </main>
240 </div>
241
242 <script>
243     // Only run script if we're on the transactions list page
244     if (document.getElementById('transactionTableBody')) {
245         let currentType = '{{ transaction_type|default:"" }}';
246
247         function performSearch() {
248             const searchQuery = document.getElementById('searchInput').value;
249             const url = new URL('{% url "search_transactions_ajax" %}', window.location.origin);
250
251             if (searchQuery) {
252                 url.searchParams.append('search', searchQuery);
253             }
254             if (currentType) {
255                 url.searchParams.append('type', currentType);
256             }
257
258             fetch(url)
259                 .then(response => response.json())
260                 .then(data => {
261                     document.getElementById('transactionTableBody').innerHTML = data.html;
262                 })
263                 .catch(error => {
264                     console.error('Error:', error);
265                 });
266         }

```

Activate Windows
Go to Settings to activate Windows.

```

267
268     // Search input with debounce
269     let searchTimeout;
270     document.getElementById('searchInput').addEventListener('input', function() {
271         clearTimeout(searchTimeout);
272         searchTimeout = setTimeout(performSearch, 300);
273     });
274
275     // Search button
276     document.getElementById('searchButton').addEventListener('click', performSearch);
277
278     // Filter buttons
279     document.getElementById('filterAll').addEventListener('click', function() {
280         currentType = '';
281         updateFilterButtons();
282         performSearch();
283     });
284
285     document.getElementById('filterIncome').addEventListener('click', function() {
286         currentType = 'INCOME';
287         updateFilterButtons();
288         performSearch();
289     });
290
291     document.getElementById('filterExpense').addEventListener('click', function() {
292         currentType = 'EXPENSE';
293         updateFilterButtons();
294         performSearch();
295     });
296

```

Activate Windows
Go to Settings to activate Windows.

```

297         function updateFilterButtons() {
298             // Reset all buttons
299             document.getElementById('filterAll').classList.remove('bg-gray-400', 'font-semibold');
300             document.getElementById('filterAll').classList.add('bg-gray-200');
301             document.getElementById('filterIncome').classList.remove('bg-green-400', 'font-semibold');
302             document.getElementById('filterIncome').classList.add('bg-green-200');
303             document.getElementById('filterExpense').classList.remove('bg-red-400', 'font-semibold');
304             document.getElementById('filterExpense').classList.add('bg-red-200');
305
306             if (currentType === '') {
307                 document.getElementById('filterAll').classList.remove('bg-gray-200');
308                 document.getElementById('filterAll').classList.add('bg-gray-400', 'font-semibold');
309             } else if (currentType === 'INCOME') {
310                 document.getElementById('filterIncome').classList.remove('bg-green-200');
311                 document.getElementById('filterIncome').classList.add('bg-green-400', 'font-semibold');
312             } else if (currentType === 'EXPENSE') {
313                 document.getElementById('filterExpense').classList.remove('bg-red-200');
314                 document.getElementById('filterExpense').classList.add('bg-red-400', 'font-semibold');
315             }
316         }
317
318         updateFilterButtons();
319     }
320 

```

Activate Windows

users/apps.py

```

1 from django.apps import AppConfig
2
3
4 class UsersConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'users'

```

users/urls.py

```

1 from django.urls import path
2 from django.shortcuts import redirect
3 from . import views
4
5 urlpatterns = [
6     path('', lambda request: redirect('login')),
7     path('login/', views.login_view, name='login'),
8     path('signup/', views.signup_view, name='signup'),
9     path('logout/', views.logout_view, name='logout'),
10 ]

```

users/views.py

```

1 from django.shortcuts import render, redirect
2 from django.contrib.auth.models import User
3 from django.contrib.auth import authenticate, login, logout
4
5 def login_view(request):
6     if request.method == "POST":
7         username = request.POST.get("username")
8         password = request.POST.get("password")
9
10        user = authenticate(request, username=username, password=password)
11
12        if user:
13            login(request, user)
14            return redirect("dashboard")
15        else:
16            return render(request, "login.html", {"error": "Invalid username or password"})
17
18        return render(request, "login.html")
19
20 def signup_view(request):
21     if request.method == "POST":
22         username = request.POST.get("username")
23         email = request.POST.get("email")
24         password = request.POST.get("password")
25         confirm_password = request.POST.get("confirm-password")
26
27         if password != confirm_password:

```

Activate Windows
Go to Settings to activate Windows.

```

28         return render(request, "signup.html", {"error": "Passwords do not match"})
29
30         if User.objects.filter(username=username).exists():
31             return render(request, "signup.html", {"error": "Username already taken"})
32
33         if User.objects.filter(email=email).exists():
34             return render(request, "signup.html", {"error": "Email already registered"})
35
36         user = User.objects.create_user(username=username, email=email, password=password)
37         login(request, user)
38         return redirect("dashboard")
39
40     return render(request, "signup.html")
41
42 def logout_view(request):
43     logout(request)
44     return redirect("login")
45
46 def home_view(request):
47     return render(request, "home.html")

```

Activate Windows
Go to Settings to activate Windows.

users/templates/login.html

```

1  {% load static %}
2
3  <!DOCTYPE html>
4  <html lang="en" class="h-full bg-gray-50">
5      <head>
6          <meta charset="UTF-8" />
7          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8          <title>Financeo - Login</title>
9          <script src="https://cdn.tailwindcss.com"></script>
10         <script src="https://cdn.jsdelivr.net/npm/feather-icons/dist/feather.min.js"></script>
11         <script defer src="{% static 'script.js' %}"></script>
12
13     <style>
14         @import url("https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap");
15         body {
16             font-family: "Inter", sans-serif;
17         }
18     </style>
19 </head>
20
21 <body class="h-full bg-gradient-to-b from-green-600 to-green-900">
22     <div class="min-h-full flex flex-col justify-center py-12 sm:px-6 lg:px-8">
23         <div class="sm:mx-auto sm:w-full sm:max-w-md">
24             <div class="bg-white py-8 px-4 shadow sm:rounded-lg sm:px-10">
25                 <h2 class="mt-2 text-center text-3xl font-extrabold text-gray-900">
26                     Financeo

```

Activate Windows
Go to Settings to activate Windows.

```

27     </h2>
28     <p class="mt-2 mb-5 text-center text-sm text-gray-600">
29         Your personal finance management tool
30     </p>
31
32     {% if error %}
33     <div class="mb-4 bg-red-100 text-red-700 px-3 py-2 rounded text-sm">
34         {{ error }}
35     </div>
36     {% endif %}
37
38     <form method="POST">
39         {% csrf_token %}
40
41         <!-- Email -->
42         <div class="relative mt-4">
43             <div
44                 class="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none"
45             >
46                 <i data-feather="user" class="h-5 w-5 text-gray-400"></i>
47             </div>
48
49             <input
50                 name="username"
51                 type="text"
52                 placeholder=" "
53                 required
54                 class="peer w-full border-b-2 border-gray-300 focus:border-green-600 outline-none py-2 pl-10 text-gray-900"
55             />
56

```

Activate Windows
Go to Settings to activate Windows.

```

57     <label
58       class="absolute left-10 -top-3.5 text-gray-500 text-sm transition-all peer-placeholder-shown:top-2 peer-placeholder-shown:text-base peer-focus:-top-3.5 peer-
59     >
60       Username
61     </label>
62   </div>
63
64   <div class="relative mt-6">
65     <div
66       class="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none"
67     >
68       <i data-feather="lock" class="h-5 w-5 text-gray-400"></i>
69     </div>
70
71     <input
72       name="password"
73       type="password"
74       placeholder=" "
75       required
76       class="peer w-full border-b-2 border-gray-300 focus:border-green-600 outline-none py-2 pl-10 text-gray-900"
77     />
78
79     <label
80       class="absolute left-10 -top-3.5 text-gray-500 text-sm transition-all peer-placeholder-shown:top-2 peer-placeholder-shown:text-base peer-focus:-top-3.5 peer-
81     >
82       Password
83     </label>
84   </div>
85
86   <!-- Submit -->

```

Activate Windows
Go to Settings to activate Windows.

```

87     <button
88       type="submit"
89       class="w-full mt-8 py-2 px-4 rounded-md text-sm font-medium text-white bg-gradient-to-r from-green-600 to-green-700 hover:from-green-700 hover:to-green-800 tr
90     >
91       Sign in
92     </button>
93   </form>
94
95   <!-- Signup Link -->
96   <div class="mt-6 text-center">
97     <p class="text-sm text-gray-600">
98       Don't have an account?
99     <a
100       href="{% url 'signup' %}"
101       class="font-medium text-green-700 hover:text-green-600"
102     >
103       Sign up
104     </a>
105   </p>
106 </div>
107 </div>
108 </div>
109 </div>
110
111 <script>
112   feather.replace();
113 </script>
114 </body>
115 </html>

```

Activate Windows
Go to Settings to activate Windows.

users/templates/signup.html

```

1  <!DOCTYPE html>
2  <html lang="en" class="h-full bg-gray-50">
3  {% load static %}
4  <head>
5      <meta charset="UTF-8" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Financeo - Sign Up</title>
8      <script src="https://cdn.tailwindcss.com"></script>
9      <script src="https://cdn.jsdelivr.net/npm/feather-icons/dist/feather.min.js"></script>
10     <script src="https://unpkg.com/feather-icons"></script>
11     <script defer src="{% static 'js/main.js' %}"></script>
12     <style>
13         @import url("https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=swap");
14     </style>
15     body {
16         font-family: "Inter", sans-serif;
17     }
18 </style>
19 </head>
20
21 <body class="h-full bg-gradient-to-b from-green-600 to-green-900">
22     <div class="min-h-full flex flex-col justify-center py-12 sm:px-6 lg:px-8">
23         <div class="sm:mx-auto sm:w-full sm:max-w-md">
24             <div class="bg-white py-8 px-4 shadow sm:rounded-lg sm:px-10">
25                 <div class="mb-4">
26                     <a

```

Activate Windows
Go to Settings to activate Windows.

```

27         href="{% url 'login' %}"
28         class="inline-flex items-center text-sm text-gray-600 hover:text-primary-600 transition-colors"
29     >
30         <i data-feather="arrow-left" class="h-4 w-4 mr-1"></i>
31         Back to Login
32     </a>
33 </div>
34 <h2 class="mt-2 text-center text-3xl font-extrabold text-gray-900">
35     Create your account
36 </h2>
37 <p class="mt-2 mb-5 text-center text-sm text-gray-600">
38     Start managing your finances today
39 </p>
40
41 <form class="space-y-6" action="{% url 'signup' %}" method="POST">
42     {% csrf_token %}
43     <div class="relative">
44         <div
45             class="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none"
46         >
47             <i data-feather="user" class="h-5 w-5 text-gray-400"></i>
48         </div>
49         <input
50             id="username"
51             name="username"
52             type="text"
53             placeholder=" "
54             autocomplete="username"
55             required
56             class="peer w-full border-b-2 border-gray-300 focus:border-primary-500 outline-none py-2 pl-10 text-gray-900"

```

Activate Windows
Go to Settings to activate Windows.


```

57     />
58     <label
59       for="username"
60       class="absolute left-10 -top-3.5 text-gray-500 text-sm transition-all peer-placeholder-shown:top-2 peer-placeholder-shown:text-gray-400 peer-placeholder-shown:transition-all"
61     >
62       Username
63     </label>
64   </div>
65
66   <div class="relative">
67     <div
68       class="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none"
69     >
70       <i data-feather="lock" class="h-5 w-5 text-gray-400"></i>
71     </div>
72     <input
73       id="password"
74       name="password"
75       type="password"
76       placeholder=" "
77       autocomplete="new-password"
78       required
79       class="peer w-full border-b-2 border-gray-300 focus:border-primary-500 outline-none py-2 pl-10 text-gray-900"
80     />
81     <label
82       for="password"
83       class="absolute left-10 -top-3.5 text-gray-500 text-sm transition-all peer-placeholder-shown:top-2 peer-placeholder-shown:text-gray-400 peer-placeholder-shown:transition-all"
84     >
85       Password
86     </label>

```

Activate Windows
Go to Settings to activate Windows.

```

87     <div
88       class="absolute inset-y-0 right-0 pr-3 flex items-center cursor-pointer"
89       onclick="togglePasswordVisibility('password')"
90     >
91       <i
92         data-feather="eye"
93         class="h-5 w-5 text-gray-400"
94         id="password-eye"
95       ></i>
96     </div>
97   </div>
98
99   <div class="relative">
100     <div
101       class="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none"
102     >
103       <i data-feather="lock" class="h-5 w-5 text-gray-400"></i>
104     </div>
105     <input
106       id="confirm-password"
107       name="confirm-password"
108       type="password"
109       placeholder=" "
110       autocomplete="new-password"
111       required
112       class="peer w-full border-b-2 border-gray-300 focus:border-primary-500 outline-none py-2 pl-10 text-gray-900"
113     />
114     <label
115       for="confirm-password"
116       class="absolute left-10 -top-3.5 text-gray-500 text-sm transition-all peer-placeholder-shown:top-2 peer-placeholder-shown:text-gray-400 peer-placeholder-shown:transition-all"

```

Activate Windows
Go to Settings to activate Windows.

```

117         >
118         Confirm Password
119     </label>
120     <div>
121         class="absolute inset-y-0 right-0 pr-3 flex items-center cursor-pointer"
122         onclick="togglePasswordVisibility('confirm-password')"
123     >
124         <i>
125             data-feather="eye"
126             class="h-5 w-5 text-gray-400"
127             id="confirm-password-eye"
128         ></i>
129     </div>
130     {% if error %}
131     <div class="text-center">
132         <p class="text-red-500 text-sm mb-4">{{ error }}</p>
133     </div>
134     {% endif %}
135 </div>
136
137 <div>
138     <button
139         type="submit"
140         class="w-full flex justify-center py-2 px-4 rounded-md text-sm font-medium text-white bg-gradient-to-r from-primary-500 to-secondary-500 hover:from-primary-600 to-secondary-600">
141     >
142     Create Account
143 </button>
144 </div>
145 </form>
146

```

Activate Windows
Go to Settings to activate Windows.

```

147     <div class="mt-6 text-center">
148         <p class="text-sm text-gray-600">
149             Already have an account?
150         <a
151             href="{% url 'login' %}"
152             class="font-medium text-primary-600 hover:text-primary-500">
153         >
154             Sign in
155         </a>
156     </p>
157 </div>
158 </div>
159 </div>
160 </div>
161
162 <script>
163     feather.replace();
164
165     function togglePasswordVisibility(inputId) {
166         const input = document.getElementById(inputId);
167         const eyeIcon = document.getElementById(inputId + "-eye");
168
169         if (input.type === "password") {
170             input.type = "text";
171             eyeIcon.setAttribute("data-feather", "eye-off");
172         } else {
173             input.type = "password";
174             eyeIcon.setAttribute("data-feather", "eye");
175         }
176         feather.replace();
177

```

Activate Windows
Go to Settings to activate Windows.

```

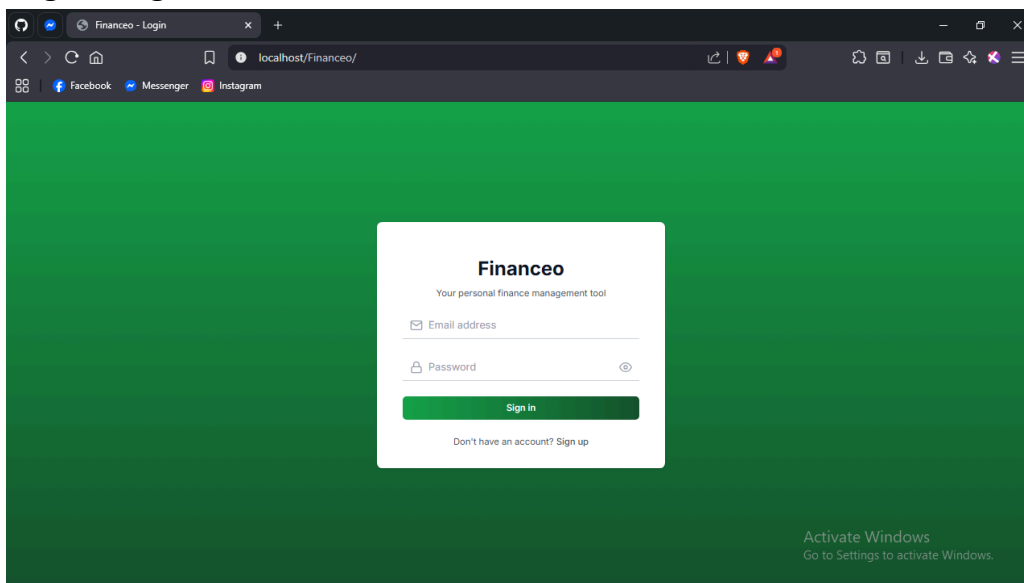
177     }
178 </script>
179 </body>
180 </html>

```

Activate Windows

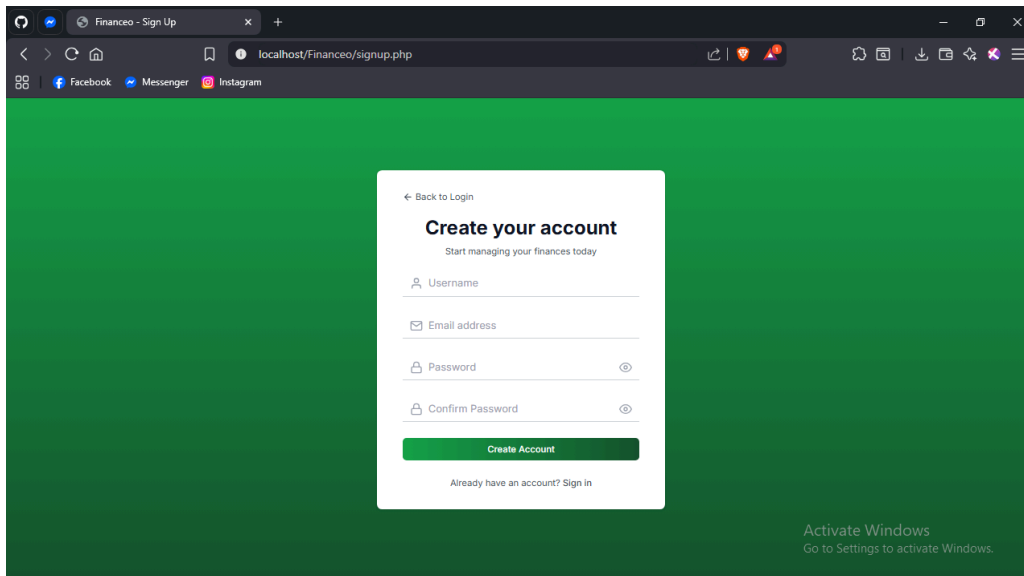
SCREENSHOTS OF THE SYSTEM

Log-in Page



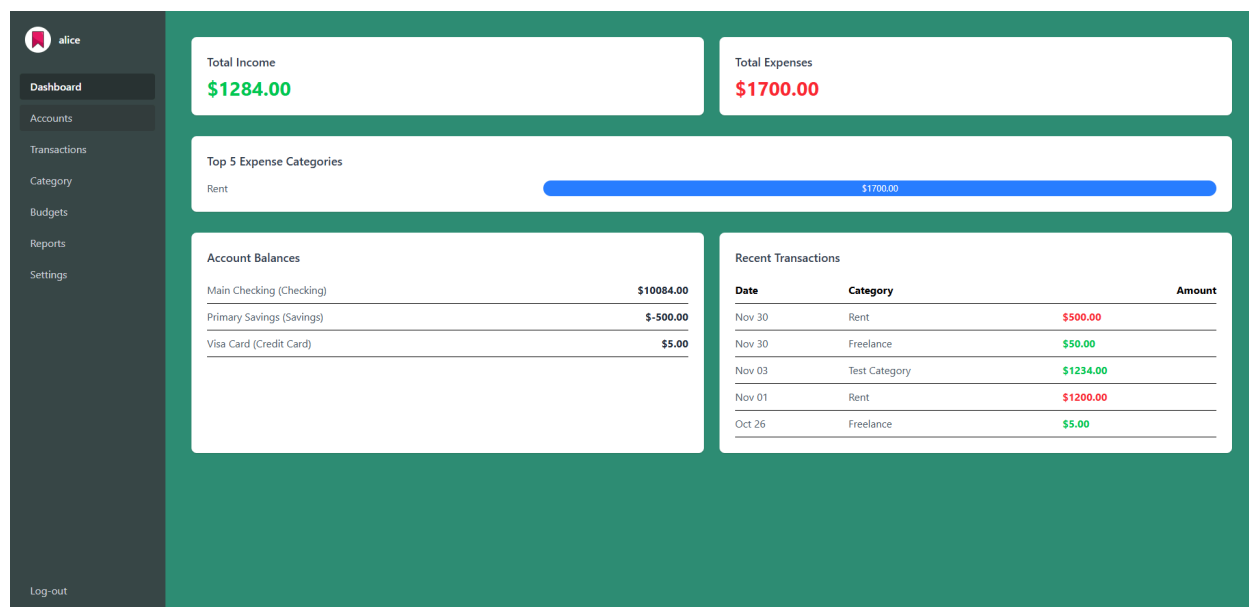
A screenshot of a web browser displaying the Financeo login page. The browser's address bar shows 'localhost/Financeo/'. The page has a dark green background. In the center, there is a white login form titled 'Financeo' with the subtitle 'Your personal finance management tool'. The form contains two input fields: 'Email address' and 'Password'. Below these fields is a green 'Sign in' button. At the bottom of the form, it says 'Don't have an account? Sign up'. In the bottom right corner of the page, there is a Windows watermark that says 'Activate Windows Go to Settings to activate Windows.'

Sign-up Page



A screenshot of a web browser displaying the Financeo sign-up page. The browser's address bar shows 'localhost/Financeo/signup.php'. The page has a dark green background. In the center, there is a white sign-up form titled 'Create your account' with the subtitle 'Start managing your finances today'. The form includes a link '← Back to Login' at the top left. It contains four input fields: 'Username', 'Email address', 'Password', and 'Confirm Password'. Below these fields is a green 'Create Account' button. At the bottom of the form, it says 'Already have an account? Sign in'. In the bottom right corner of the page, there is a Windows watermark that says 'Activate Windows Go to Settings to activate Windows.'

Dashboard



Accounts

alice

Dashboard

Accounts

Transactions

Category

Budgets

Reports

Settings

Log-out

Add New Account

Name:

Type:

Checking

Current balance:

0.0

Add Account

Your Accounts

ACCOUNT NAME	TYPE	BALANCE		
Main Checking	Checking	\$10084.00	Update	Delete
Primary Savings	Savings	\$-500.00	Update	Delete
Visa Card	Credit Card	\$5.00	Update	Delete

Transactions

alice

Dashboard

Accounts

Transactions

Category

Budgets

Reports

Settings

Log-out

Transactions

Add Transaction

Search by description or category...

All

Income

Expense

Search

All Transactions

Date	Account	Category	Description	Amount	Actions	
Nov 30, 2025	Primary Savings	Rent	-	-\$500.00	Edit	Delete
Nov 30, 2025	Main Checking	Freelance	test add	+\$50.00	Edit	Delete
Nov 03, 2025	Main Checking	Test Category	-	+\$1234.00	Edit	Delete
Nov 01, 2025	Main Checking	Rent	Monthly Rent	-\$1200.00	Edit	Delete
Oct 26, 2025	Visa Card	Freelance	-	+\$5.00	Edit	Delete

Add Transaction

alice

Dashboard

Accounts

Transactions

Category

Budgets

Reports

Settings

Log-out

Create Transaction

Account

Type

Category

Amount

Date

mm/dd/yyyy

Description

Create Transaction

Cancel

Category

alice

Dashboard

Accounts

Transactions

Category

Budgets

Reports

Settings

Log-out

Categories

+ Add Category

Groceries	Expense	Update	Delete
Rent	Expense	Update	Delete
Restaurants	Expense	Update	Delete
Utilities	Expense	Update	Delete
Freelance	Income	Update	Delete
Salary	Income	Update	Delete
Test Category	Income	Update	Delete

Add Category

alice

Dashboard

Accounts

Transactions

Category

Budgets

Reports

Settings

Log-out

Create Category

Category Name

Category name

Type

Expense

Create Category

Cancel

Budgets

alice

Dashboard

Accounts

Transactions

Category

Budgets

Reports

Settings

Log out

Add New Budget

Category:

Amount:

Start date:
mm/dd/yyyy

End date:
mm/dd/yyyy

Add Budget

Your Budgets

CATEGORY	AMOUNT	START DATE	END DATE		
Groceries	\$500.00	Nov. 1, 2025	Nov. 30, 2025	Update	Delete
Restaurants	\$150.00	Nov. 1, 2025	Nov. 30, 2025	Update	Delete

Reports

alice

Dashboard

Accounts

Transactions

Category

Budgets

Reports

Settings

Log out

Expense Breakdown

Expense Breakdown by Category

100.0%

Rent

Income vs Expense


Income vs Expense

Amount

Income

Expense

Settings

alice

Dashboard

Accounts

Transactions


Category

Budgets

Reports

Settings

Log out

alice

Username

alice

[Update Username](#)

Profile Picture

Choose File No file chosen

[Update Picture](#)

E-Mail Address

alice@example.com

New E-Mail Address

[Update E-mail](#)

Password

Current Password

New Password

Confirm New Password

[Change Password](#)