Django + Tailwind CSS Complete Starter Project

A modern Django web application starter template with Tailwind CSS, complete user management system, and beautiful UI components.

🚀 Project Overview

This is a complete Django starter project that includes:

- **Modern UI**: Tailwind CSS for styling
- **User Management**: Complete authentication system
- **Responsive Design**: Mobile-first approach
- **Component Library**: Reusable UI components
- **Database Integration**: SQLite/PostgreSQL ready
- **Production Ready**: Deployment configurations

✨ Features

🎨 Modern UI/UX

- **Tailwind CSS**: Utility-first CSS framework
- **Responsive Design**: Works on all devices
- **Dark/Light Theme**: Theme switching capability
- **Component Library**: Pre-built UI components
- **Icons**: Heroicons integration
- **Animations**: Smooth transitions and animations

👥 User Management System

- **Custom User Model**: Extended user functionality
- **Authentication**: Login/Logout/Register
- **Profile Management**: User profile with avatar
- **Password Management**: Change/Reset password
- **Email Verification**: Account activation

- **Role-Based Access**: Admin and user roles

🛠️ Development Features

- **Django 4.2+**: Latest Django version
- **Hot Reload**: Automatic CSS compilation
- **Form Validation**: Client and server-side validation
- **CSRF Protection**: Security best practices
- **Debug Toolbar**: Development debugging
- **Environment Variables**: Configuration management

📁 Project Structure

```
django-tailwind-starter/
├── manage.py
├── requirements.txt
├── .env.example
├── .gitignore
├── README.md
├── static/
│   ├── css/
│   │   ├── input.css
│   │   └── output.css
│   ├── js/
│   │   └── main.js
│   └── images/
├── media/
│   └── profiles/
├── config/
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   └── asgi.py
├── apps/
│   ├── __init__.py
│   ├── accounts/
│   │   ├── __init__.py
│   │   ├── models.py
│   │   ├── views.py
│   │   ├── forms.py
│   │   ├── urls.py
│   │   ├── admin.py
│   │   ├── signals.py
│   │   └── migrations/
│   └── core/
│       ├── __init__.py
│       ├── views.py
│       ├── urls.py
│       └── utils.py
└── templates/
    ├── base.html
    ├── components/
    │   ├── navbar.html
    │   ├── footer.html
    │   ├── forms.html
    │   └── alerts.html
    ├── accounts/
    │   ├── login.html
    │   ├── register.html
    │   ├── profile.html
    │   └── password_change.html
    └── core/
        ├── home.html
        ├── about.html
        └── dashboard.html
```

🏗️ Installation & Setup

Step 1: System Requirements

```
# Python 3.8+ required
python --version

# Node.js 16+ required for Tailwind CSS
node --version
npm --version
```

Step 2: Create Project Directory

```
# Create project directory
mkdir django-tailwind-starter
cd django-tailwind-starter

# Create virtual environment
python -m venv venv

# Activate virtual environment
# Windows:
venv\Scripts\activate
# macOS/Linux:
source venv/bin/activate
```

Step 3: Install Python Dependencies

```
# Install Django and dependencies
pip install django==4.2
pip install pillow
pip install python-decouple
pip install django-crispy-forms
pip install crispy-tailwind
pip install django-allauth
pip install django-debug-toolbar

# Create requirements.txt
pip freeze > requirements.txt
```

Step 4: Install Node.js Dependencies

```
# Initialize npm
npm init -y

# Install Tailwind CSS
npm install -D tailwindcss @tailwindcss/forms @tailwindcss/typography
npm install -D @heroicons/react

# Create package.json scripts
```

Step 5: Create Django Project

```
# Create Django project
django-admin startproject config .

# Create apps directory
mkdir apps
cd apps

# Create accounts app
python ../manage.py startapp accounts

# Create core app
python ../manage.py startapp core

# Return to root
cd ..
```

🔧 Configuration Files

Step 6: Environment Configuration

Create `.env` file:

```
# .env
SECRET_KEY=your-secret-key-here
DEBUG=True
ALLOWED_HOSTS=localhost,127.0.0.1
DATABASE_URL=sqlite:///db.sqlite3
EMAIL_BACKEND=django.core.mail.backends.console.EmailBackend
```

Step 7: Django Settings

Update `config/settings.py` :

```python
import os
from pathlib import Path
from decouple import config

BASE_DIR = Path(__file__).resolve().parent.parent

# Security
SECRET_KEY = config('SECRET_KEY', default='your-secret-key')
DEBUG = config('DEBUG', default=True, cast=bool)
ALLOWED_HOSTS = config('ALLOWED_HOSTS', default='localhost,127.0.0.1').split(',')

# Applications
DJANGO_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django.contrib.sites',
]

THIRD_PARTY_APPS = [
    'crispy_forms',
    'crispy_tailwind',
    'allauth',
    'allauth.account',
    'allauth.socialaccount',
    'debug_toolbar',
]

LOCAL_APPS = [
    'apps.accounts',
    'apps.core',
]

INSTALLED_APPS = DJANGO_APPS + THIRD_PARTY_APPS + LOCAL_APPS

# Middleware
MIDDLEWARE = [
    'debug_toolbar.middleware.DebugToolbarMiddleware',
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'config.urls'

# Templates
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

# Database
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Static files
STATIC_URL = '/static/'
STATIC_ROOT = BASE_DIR / 'staticfiles'
STATICFILES_DIRS = [
    BASE_DIR / 'static',
]

# Media files
MEDIA_URL = '/media/'
```

```
MEDIA_ROOT = BASE_DIR / 'media'

# Custom User Model
AUTH_USER_MODEL = 'accounts.User'

# Crispy Forms
CRISPY_ALLOWED_TEMPLATE_PACKS = "tailwind"
CRISPY_TEMPLATE_PACK = "tailwind"

# Allauth Configuration
SITE_ID = 1
ACCOUNT_EMAIL_REQUIRED = True
ACCOUNT_EMAIL_VERIFICATION = 'mandatory'
ACCOUNT_AUTHENTICATION_METHOD = 'email'
ACCOUNT_USERNAME_REQUIRED = False

# Email Configuration
EMAIL_BACKEND = config('EMAIL_BACKEND', default='django.core.mail.backends.console.EmailBackend

# Debug Toolbar
INTERNAL_IPS = [
    '127.0.0.1',
]

# Internationalization
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_TZ = True

# Default primary key field type
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

# Login/Logout URLs
LOGIN_URL = 'account_login'
LOGIN_REDIRECT_URL = 'dashboard'
LOGOUT_REDIRECT_URL = 'home'
```

Step 8: Tailwind CSS Configuration

Create `tailwind.config.js` :

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: [
    './templates/**/*.html',
    './apps/**/*.py',
    './apps/**/*.html',
  ],
  theme: {
    extend: {
      colors: {
        'primary': {
          50: '#eff6ff',
          500: '#3b82f6',
          600: '#2563eb',
          700: '#1d4ed8',
        },
        'secondary': {
          50: '#f8fafc',
          500: '#64748b',
          600: '#475569',
          700: '#334155',
        }
      },
      fontFamily: {
        'sans': ['Inter', 'system-ui', 'sans-serif'],
      },
    },
  },
  plugins: [
    require('@tailwindcss/forms'),
    require('@tailwindcss/typography'),
  ],
  darkMode: 'class',
}
```

Create `static/css/input.css` :

```
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
  html {
    font-family: 'Inter', system-ui, sans-serif;
  }
}

@layer components {
  .btn {
    @apply px-4 py-2 rounded-lg font-medium transition-colors duration-200 focus:outline-none f
  }

  .btn-primary {
    @apply bg-primary-600 text-white hover:bg-primary-700 focus:ring-primary-500;
  }

  .btn-secondary {
    @apply bg-gray-200 text-gray-900 hover:bg-gray-300 focus:ring-gray-500;
  }

  .btn-danger {
    @apply bg-red-600 text-white hover:bg-red-700 focus:ring-red-500;
  }

  .form-input {
    @apply block w-full px-3 py-2 border border-gray-300 rounded-md shadow-sm placeholder-gray-
  }

  .card {
    @apply bg-white rounded-lg shadow-sm border border-gray-200 overflow-hidden;
  }

  .card-header {
    @apply px-6 py-4 border-b border-gray-200 bg-gray-50;
  }

  .card-body {
    @apply px-6 py-4;
  }
}
```

Update `package.json`:

```
{
  "scripts": {
    "build-css": "tailwindcss -i ./static/css/input.css -o ./static/css/output.css --watch",
    "build-css-prod": "tailwindcss -i ./static/css/input.css -o ./static/css/output.css --minify
  }
}
```

📊 Models

Step 9: Create User Model

`apps/accounts/models.py`:

```python
from django.contrib.auth.models import AbstractUser
from django.db import models
from django.urls import reverse
from PIL import Image

class User(AbstractUser):
    email = models.EmailField(unique=True)
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    bio = models.TextField(max_length=500, blank=True)
    avatar = models.ImageField(
        upload_to='profiles/',
        default='profiles/default.jpg',
        blank=True
    )
    phone = models.CharField(max_length=20, blank=True)
    birth_date = models.DateField(null=True, blank=True)
    is_verified = models.BooleanField(default=False)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    USERNAME_FIELD = 'email'
    REQUIRED_FIELDS = ['username', 'first_name', 'last_name']

    class Meta:
        verbose_name = 'User'
        verbose_name_plural = 'Users'
        ordering = ['-created_at']

    def __str__(self):
        return f"{self.first_name} {self.last_name}"

    def get_absolute_url(self):
        return reverse('accounts:profile', kwargs={'pk': self.pk})

    def get_full_name(self):
        return f"{self.first_name} {self.last_name}".strip()

    def get_short_name(self):
        return self.first_name

    def save(self, *args, **kwargs):
        super().save(*args, **kwargs)
        # Resize avatar if it exists
        if self.avatar:
            img = Image.open(self.avatar.path)
            if img.height > 300 or img.width > 300:
                output_size = (300, 300)
                img.thumbnail(output_size)
                img.save(self.avatar.path)

class Profile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    website = models.URLField(blank=True)
    github = models.URLField(blank=True)
    linkedin = models.URLField(blank=True)
    twitter = models.URLField(blank=True)
    location = models.CharField(max_length=100, blank=True)
    company = models.CharField(max_length=100, blank=True)
    job_title = models.CharField(max_length=100, blank=True)

    def __str__(self):
        return f"{self.user.get_full_name()}'s Profile"
```

Step 10: Create Profile Signal

`apps/accounts/signals.py` :

```
from django.db.models.signals import post_save
from django.dispatch import receiver
from django.contrib.auth import get_user_model
from .models import Profile

User = get_user_model()

@receiver(post_save, sender=User)
def create_user_profile(sender, instance, created, **kwargs):
    if created:
        Profile.objects.create(user=instance)

@receiver(post_save, sender=User)
def save_user_profile(sender, instance, **kwargs):
    instance.profile.save()
```

Step 11: Register Signals

`apps/accounts/apps.py` :

```
from django.apps import AppConfig

class AccountsConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'apps.accounts'

    def ready(self):
        import apps.accounts.signals
```

🎨 Templates

Step 12: Base Template

`templates/base.html` :

```
<!DOCTYPE html>
<html lang="en" class="h-full">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}Django Tailwind Starter{% endblock %}</title>

    <!-- Tailwind CSS -->
    {% load static %}
    <link href="{% static 'css/output.css' %}" rel="stylesheet">

    <!-- Inter Font -->
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&display=

    <!-- Heroicons -->
    <script src="https://unpkg.com/heroicons@2.0.18/24/outline/index.js" type="module"></script

    {% block extra_css %}{% endblock %}
</head>
<body class="h-full bg-gray-50">
    <!-- Navigation -->
    {% include 'components/navbar.html' %}

    <!-- Main Content -->
    <main class="min-h-screen">
        <!-- Messages -->
        {% if messages %}
            {% for message in messages %}
                <div class="alert alert-{{ message.tags }} max-w-7xl mx-auto px-4 sm:px-6 lg:px-
                    {% include 'components/alerts.html' %}
                </div>
            {% endfor %}
        {% endif %}

        <!-- Page Content -->
        {% block content %}{% endblock %}
    </main>

    <!-- Footer -->
    {% include 'components/footer.html' %}

    <!-- JavaScript -->
    <script src="{% static 'js/main.js' %}"></script>
    {% block extra_js %}{% endblock %}
</body>
</html>
```

Step 13: Navigation Component

`templates/components/navbar.html` :

```
<nav class="bg-white shadow">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
        <div class="flex justify-between h-16">
            <div class="flex">
                <div class="flex-shrink-0 flex items-center">
                    <a href="{% url 'home' %}" class="text-xl font-bold text-gray-900">
                        Django Starter
                    </a>
                </div>
                <div class="hidden sm:ml-6 sm:flex sm:space-x-8">
                    <a href="{% url 'home' %}" class="border-transparent text-gray-500 hover:bor
                        Home
                    </a>
                    {% if user.is_authenticated %}
                        <a href="{% url 'dashboard' %}" class="border-transparent text-gray-500
                            Dashboard
                        </a>
                    {% endif %}
                </div>
            </div>
            <div class="hidden sm:ml-6 sm:flex sm:items-center">
                {% if user.is_authenticated %}
                    <div class="ml-3 relative">
                        <div class="flex items-center space-x-4">
                            <span class="text-sm text-gray-700">{{ user.get_full_name }}</span>
                            <img class="h-8 w-8 rounded-full" src="{{ user.avatar.url }}" alt="
                            <a href="{% url 'accounts:profile' user.pk %}" class="text-gray-500
                                Profile
                            </a>
                            <a href="{% url 'account_logout' %}" class="text-gray-500 hover:text
                                Logout
                            </a>
                        </div>
                    </div>
                {% else %}
                    <div class="flex items-center space-x-4">
                        <a href="{% url 'account_login' %}" class="text-gray-500 hover:text-gray
                            Sign in
                        </a>
                        <a href="{% url 'account_signup' %}" class="btn btn-primary text-sm">
                            Sign up
                        </a>
                    </div>
                {% endif %}
            </div>
        </div>
    </div>
</nav>
```

Step 14: Home Page

`templates/core/home.html`:

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

{% block title %}Home - Django Tailwind Starter{% endblock %}

{% block content %}
<div class="relative bg-white overflow-hidden">
    <!-- Hero Section -->
    <div class="max-w-7xl mx-auto">
        <div class="relative z-10 pb-8 bg-white sm:pb-16 md:pb-20 lg:max-w-2xl lg:w-full lg:pb-
            <main class="mt-10 mx-auto max-w-7xl px-4 sm:mt-12 sm:px-6 md:mt-16 lg:mt-20 lg:px-
                <div class="sm:text-center lg:text-left">
                    <h1 class="text-4xl tracking-tight font-extrabold text-gray-900 sm:text-5xl
                        <span class="block xl:inline">Django +</span>
                        <span class="block text-primary-600 xl:inline">Tailwind CSS</span>
                    </h1>
                    <p class="mt-3 text-base text-gray-500 sm:mt-5 sm:text-lg sm:max-w-xl sm:mx
                        A modern, production-ready Django starter template with Tailwind CSS, co
                    </p>
                    <div class="mt-5 sm:mt-8 sm:flex sm:justify-center lg:justify-start">
                        <div class="rounded-md shadow">
                            {% if user.is_authenticated %}
                                <a href="{% url 'dashboard' %}" class="w-full flex items-center
                                    Go to Dashboard
                                </a>
                            {% else %}
                                <a href="{% url 'account_signup' %}" class="w-full flex items-ce
                                    Get Started
                                </a>
                            {% endif %}
                        </div>
                        <div class="mt-3 sm:mt-0 sm:ml-3">
                            <a href="#features" class="w-full flex items-center justify-center
                                Learn More
                            </a>
                        </div>
                    </div>
                </div>
            </main>
        </div>
    </div>

    <!-- Features Section -->
```

```
<div id="features" class="py-12 bg-white">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
        <div class="lg:text-center">
            <h2 class="text-base text-primary-600 font-semibold tracking-wide uppercase">Fea
            <p class="mt-2 text-3xl leading-8 font-extrabold tracking-tight text-gray-900 sr
                Everything you need to get started
            </p>
        </div>

        <div class="mt-10">
            <div class="space-y-10 md:space-y-0 md:grid md:grid-cols-2 md:gap-x-8 md:gap-y-
                <div class="relative">
                    <div class="absolute flex items-center justify-center h-12 w-12 rounded-
                        <svg class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="curren
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-wid
                        </svg>
                    </div>
                    <p class="ml-16 text-lg leading-6 font-medium text-gray-900">Fast Devel
                    <p class="mt-2 ml-16 text-base text-gray-500">
                        Built with modern tools and best practices for rapid development.
                    </p>
                </div>

                <div class="relative">
                    <div class="absolute flex items-center justify-center h-12 w-12 rounded-
                        <svg class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="curren
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-wid
                        </svg>
                    </div>
                    <p class="ml-16 text-lg leading-6 font-medium text-gray-900">Secure Aut
                    <p class="mt-2 ml-16 text-base text-gray-500">
                        Complete user management with secure authentication and authorizati
                    </p>
                </div>

                <div class="relative">
                    <div class="absolute flex items-center justify-center h-12 w-12 rounded-
                        <svg class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="curren
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-wid
                        </svg>
                    </div>
                    <p class="ml-16 text-lg leading-6 font-medium text-gray-900">Beautiful
                    <p class="mt-2 ml-16 text-base text-gray-500">
```

```
                            Modern and responsive design with Tailwind CSS components.
                        </p>
                    </div>

                    <div class="relative">
                        <div class="absolute flex items-center justify-center h-12 w-12 rounded
                            <svg class="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="curren
                                <path stroke-linecap="round" stroke-linejoin="round" stroke-wid
                            </svg>
                        </div>
                        <p class="ml-16 text-lg leading-6 font-medium text-gray-900">Production
                        <p class="mt-2 ml-16 text-base text-gray-500">
                            Configured for deployment with security and performance best practi
                        </p>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock %}
```

🔗 URLs Configuration

Step 15: Main URLs

`config/urls.py` :

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('apps.core.urls')),
    path('accounts/', include('apps.accounts.urls')),
    path('accounts/', include('allauth.urls')),
]

if settings.DEBUG:
    import debug_toolbar
    urlpatterns = [
        path('__debug__/', include(debug_toolbar.urls)),
    ] + urlpatterns

    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
    urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

Step 16: Core URLs

`apps/core/urls.py` :

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.HomeView.as_view(), name='home'),
    path('dashboard/', views.DashboardView.as_view(), name='dashboard'),
    path('about/', views.AboutView.as_view(), name='about'),
]
```

🚀 Running the Project

Step 17: Database Setup

```
# Create and apply migrations
python manage.py makemigrations
python manage.py migrate

# Create superuser
python manage.py createsuperuser

# Load sample data (optional)
python manage.py loaddata fixtures/sample_data.json
```

Step 18: Build CSS

```
# Build Tailwind CSS (in separate terminal)
npm run build-css

# Or build for production
npm run build-css-prod
```

Step 19: Start Development Server

```
# Run Django development server
python manage.py runserver

# Access application at http://127.0.0.1:8000/
```

📱 Views and Forms

Step 20: Core Views

`apps/core/views.py` :

```python
from django.shortcuts import render
from django.views.generic import TemplateView
from django.contrib.auth.mixins import LoginRequiredMixin
from django.contrib.auth import get_user_model

User = get_user_model()

class HomeView(TemplateView):
    template_name = 'core/home.html'

class DashboardView(LoginRequiredMixin, TemplateView):
    template_name = 'core/dashboard.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['total_users'] = User.objects.count()
        context['recent_users'] = User.objects.order_by('-created_at')[:5]
        return context

class AboutView(TemplateView):
    template_name = 'core/about.html'
```

Step 21: Account Views

`apps/accounts/views.py`:

```python
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth.decorators import login_required
from django.contrib.auth.mixins import LoginRequiredMixin
from django.contrib import messages
from django.views.generic import DetailView, UpdateView
from django.urls import reverse_lazy
from django.contrib.auth import get_user_model
from .forms import UserUpdateForm, ProfileUpdateForm
from .models import Profile

User = get_user_model()

class ProfileView(LoginRequiredMixin, DetailView):
    model = User
    template_name = 'accounts/profile.html'
    context_object_name = 'profile_user'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['is_own_profile'] = self.object == self.request.user
        return context

class ProfileUpdateView(LoginRequiredMixin, UpdateView):
    model = User
    form_class = UserUpdateForm
    template_name = 'accounts/profile_edit.html'

    def get_object(self):
        return self.request.user

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        if self.request.POST:
            context['profile_form'] = ProfileUpdateForm(
                self.request.POST,
                instance=self.request.user.profile
            )
        else:
            context['profile_form'] = ProfileUpdateForm(
                instance=self.request.user.profile
            )
        return context
    def form_valid(self, form):
        context = self.get_context_data()
        profile_form = context['profile_form']

        if profile_form.is_valid():
            form.save()
            profile_form.save()
            messages.success(self.request, 'Your profile has been updated successfully!')
            return redirect('accounts:profile', pk=self.request.user.pk)
        else:
            return self.form_invalid(form)

@login_required
def profile_view(request, pk):
    user = get_object_or_404(User, pk=pk)
    context = {
        'profile_user': user,
        'is_own_profile': user == request.user
    }
    return render(request, 'accounts/profile.html', context)
```

Step 22: Account Forms

`apps/accounts/forms.py` :

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import get_user_model
from .models import Profile

User = get_user_model()

class CustomUserCreationForm(UserCreationForm):
    email = forms.EmailField(required=True)
    first_name = forms.CharField(max_length=30, required=True)
    last_name = forms.CharField(max_length=30, required=True)

    class Meta:
        model = User
        fields = ('email', 'username', 'first_name', 'last_name', 'password1', 'password2')

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for fieldname in ['email', 'username', 'first_name', 'last_name', 'password1', 'passwor
            self.fields[fieldname].widget.attrs['class'] = 'form-input'
            self.fields[fieldname].widget.attrs['placeholder'] = self.fields[fieldname].label

class UserUpdateForm(forms.ModelForm):
    class Meta:
        model = User
        fields = ['first_name', 'last_name', 'email', 'bio', 'avatar', 'phone', 'birth_date']
        widgets = {
            'birth_date': forms.DateInput(attrs={'type': 'date'}),
            'bio': forms.Textarea(attrs={'rows': 4}),
        }

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for fieldname in self.fields:
            if fieldname != 'avatar':
                self.fields[fieldname].widget.attrs['class'] = 'form-input'

class ProfileUpdateForm(forms.ModelForm):
    class Meta:
        model = Profile
        fields = ['website', 'github', 'linkedin', 'twitter', 'location', 'company', 'job_title

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for fieldname in self.fields:
            self.fields[fieldname].widget.attrs['class'] = 'form-input'
            self.fields[fieldname].required = False
```

Step 23: Account URLs

`apps/accounts/urls.py`：

```
from django.urls import path
from . import views

app_name = 'accounts'

urlpatterns = [
    path('profile/<int:pk>/', views.ProfileView.as_view(), name='profile'),
    path('profile/edit/', views.ProfileUpdateView.as_view(), name='profile_edit'),
]
```

🎨 Additional Templates

Step 24: Dashboard Template

`templates/core/dashboard.html`：

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

{% block title %}Dashboard - Django Tailwind Starter{% endblock %}

{% block content %}
<div class="min-h-screen bg-gray-50 py-8">
    <div class="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
        <!-- Header -->
        <div class="md:flex md:items-center md:justify-between">
            <div class="flex-1 min-w-0">
                <h2 class="text-2xl font-bold leading-7 text-gray-900 sm:text-3xl sm:truncate">
                    Welcome back, {{ user.first_name }}!
                </h2>
                <p class="mt-1 text-sm text-gray-500">
                    Here's what's happening with your account today.
                </p>
            </div>
            <div class="mt-4 flex md:mt-0 md:ml-4">
                <a href="{% url 'accounts:profile_edit' %}" class="btn btn-primary">
                    Edit Profile
                </a>
            </div>
        </div>

        <!-- Stats -->
        <div class="mt-8">
            <div class="grid grid-cols-1 gap-5 sm:grid-cols-2 lg:grid-cols-3">
                <!-- Profile Completion -->
                <div class="card">
                    <div class="card-body">
                        <div class="flex items-center">
                            <div class="flex-shrink-0">
                                <div class="w-8 h-8 bg-green-500 rounded-full flex items-center
                                    <svg class="w-5 h-5 text-white" fill="currentColor" viewBox
                                        <path fill-rule="evenodd" d="M16.707 5.293a1 1 0 010 1.
                                    </svg>
                                </div>
                            </div>
                            <div class="ml-5 w-0 flex-1">
                                <dl>
                                    <dt class="text-sm font-medium text-gray-500 truncate">Prof
                                    <dd class="text-lg font-medium text-gray-900">85%</dd>
                                </dl>
                            </div>
                        </div>
                    </div>
                </div>

                <!-- Member Since -->
                <div class="card">
                    <div class="card-body">
                        <div class="flex items-center">
                            <div class="flex-shrink-0">
                                <div class="w-8 h-8 bg-blue-500 rounded-full flex items-center
                                    <svg class="w-5 h-5 text-white" fill="currentColor" viewBox
                                        <path fill-rule="evenodd" d="M6 2a1 1 0 00-1 1v1H4a2 2
                                    </svg>
                                </div>
                            </div>
                            <div class="ml-5 w-0 flex-1">
                                <dl>
                                    <dt class="text-sm font-medium text-gray-500 truncate">Memb
                                    <dd class="text-lg font-medium text-gray-900">{{ user.creat
                                </dl>
                            </div>
                        </div>
                    </div>
                </div>

                <!-- Total Users -->
                <div class="card">
                    <div class="card-body">
                        <div class="flex items-center">
                            <div class="flex-shrink-0">
                                <div class="w-8 h-8 bg-purple-500 rounded-full flex items-cente
                                    <svg class="w-5 h-5 text-white" fill="currentColor" viewBox
                                        <path d="M9 6a3 3 0 11-6 0 3 3 0 016 0zM17 6a3 3 0 11-6
                                    </svg>
                                </div>
                            </div>
                            <div class="ml-5 w-0 flex-1">
                                <dl>
                                    <dt class="text-sm font-medium text-gray-500 truncate">Tota
                                    <dd class="text-lg font-medium text-gray-900">{{ total_user
                                </dl>
```

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

<!-- Profile Section -->
<div class="mt-8 grid grid-cols-1 gap-8 lg:grid-cols-2">
    <!-- Profile Info -->
    <div class="card">
        <div class="card-header">
            <h3 class="text-lg leading-6 font-medium text-gray-900">Profile Information
        </div>
        <div class="card-body">
            <div class="flex items-center space-x-4">
                <img class="h-16 w-16 rounded-full object-cover" src="{{ user.avatar.ur
                <div>
                    <h4 class="text-lg font-medium text-gray-900">{{ user.get_full_name
                    <p class="text-sm text-gray-500">{{ user.email }}</p>
                    {% if user.bio %}
                        <p class="mt-2 text-sm text-gray-700">{{ user.bio }}</p>
                    {% endif %}
                </div>
            </div>
            <div class="mt-6">
                <a href="{% url 'accounts:profile' user.pk %}" class="btn btn-secondary"
                    View Full Profile
                </a>
            </div>
        </div>
    </div>

    <!-- Recent Users -->
    <div class="card">
        <div class="card-header">
            <h3 class="text-lg leading-6 font-medium text-gray-900">Recent Users</h3>
        </div>
        <div class="card-body">
            <div class="space-y-4">
                {% for recent_user in recent_users %}
                    <div class="flex items-center space-x-3">
                        <img class="h-8 w-8 rounded-full" src="{{ recent_user.avatar.ur
```

```
                                <div class="flex-1 min-w-0">
                                    <p class="text-sm font-medium text-gray-900 truncate">
                                        {{ recent_user.get_full_name }}
                                    </p>
                                    <p class="text-sm text-gray-500 truncate">
                                        Joined {{ recent_user.created_at|timesince }} ago
                                    </p>
                                </div>
                            </div>
                        {% endfor %}
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock %}
```

Step 25: Profile Template

`templates/accounts/profile.html`:

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

{% block title %}{{ profile_user.get_full_name }} - Profile{% endblock %}

{% block content %}
<div class="min-h-screen bg-gray-50 py-8">
    <div class="max-w-4xl mx-auto px-4 sm:px-6 lg:px-8">
        <!-- Profile Header -->
        <div class="card">
            <div class="card-body">
                <div class="sm:flex sm:items-center sm:justify-between">
                    <div class="sm:flex sm:space-x-5">
                        <div class="flex-shrink-0">
                            <img class="mx-auto h-20 w-20 rounded-full object-cover" src="{{ pro
                        </div>
                        <div class="mt-4 text-center sm:mt-0 sm:pt-1 sm:text-left">
                            <p class="text-sm font-medium text-gray-600">Welcome back,</p>
                            <p class="text-xl font-bold text-gray-900 sm:text-2xl">{{ profile_us
                            <p class="text-sm font-medium text-gray-600">{{ profile_user.email }
                            {% if profile_user.profile.job_title %}
                                <p class="text-sm text-gray-500">{{ profile_user.profile.job_tit
                            {% endif %}
                        </div>
                    </div>
                    {% if is_own_profile %}
                        <div class="mt-5 flex justify-center sm:mt-0">
                            <a href="{% url 'accounts:profile_edit' %}" class="btn btn-primary">
                                Edit Profile
                            </a>
                        </div>
                    {% endif %}
                </div>
            </div>
        </div>

        <!-- Profile Content -->
        <div class="mt-8 grid grid-cols-1 gap-8 lg:grid-cols-3">
            <!-- Main Info -->
            <div class="lg:col-span-2">
                <div class="card">
                    <div class="card-header">
                        <h3 class="text-lg leading-6 font-medium text-gray-900">About</h3>
                    </div>
                    <div class="card-body">
                        {% if profile_user.bio %}
                            <p class="text-gray-700">{{ profile_user.bio }}</p>
                        {% else %}
                            <p class="text-gray-500 italic">No bio available.</p>
                        {% endif %}

                        <div class="mt-6 grid grid-cols-1 gap-4 sm:grid-cols-2">
                            {% if profile_user.phone %}
                                <div>
                                    <dt class="text-sm font-medium text-gray-500">Phone</dt>
                                    <dd class="mt-1 text-sm text-gray-900">{{ profile_user.phone
                                </div>
                            {% endif %}
                            {% if profile_user.birth_date %}
                                <div>
                                    <dt class="text-sm font-medium text-gray-500">Birthday</dt>
                                    <dd class="mt-1 text-sm text-gray-900">{{ profile_user.birt
                                </div>
                            {% endif %}
                            {% if profile_user.profile.location %}
                                <div>
                                    <dt class="text-sm font-medium text-gray-500">Location</dt>
                                    <dd class="mt-1 text-sm text-gray-900">{{ profile_user.prof
                                </div>
                            {% endif %}
                            {% if profile_user.profile.company %}
                                <div>
                                    <dt class="text-sm font-medium text-gray-500">Company</dt>
                                    <dd class="mt-1 text-sm text-gray-900">{{ profile_user.prof
                                </div>
                            {% endif %}
                        </div>
                    </div>
                </div>
            </div>

            <!-- Sidebar -->
            <div>
                <div class="card">
                    <div class="card-header">
                        <h3 class="text-lg leading-6 font-medium text-gray-900">Links</h3>
                    </div>
                    <div class="card-body">
                        <div class="space-y-3">
                            {% if profile_user.profile.website %}
                                <a href="{{ profile_user.profile.website }}" target="_blank" cla
                                    <svg class="mr-2 h-4 w-4" fill="currentColor" viewBox="0 0 2
                                        <path fill-rule="evenodd" d="M12.586 4.586a2 2 0 112.828
                                    </svg>
                                    Website
                                </a>
                            {% endif %}
                            {% if profile_user.profile.github %}
                                <a href="{{ profile_user.profile.github }}" target="_blank" clas
                                    <svg class="mr-2 h-4 w-4" fill="currentColor" viewBox="0 0 2
                                        <path fill-rule="evenodd" d="M10 0C4.477 0 0 4.484 0 10
                                    </svg>
                                    GitHub
                                </a>
                            {% endif %}
                            {% if profile_user.profile.linkedin %}
                                <a href="{{ profile_user.profile.linkedin }}" target="_blank" cl
                                    <svg class="mr-2 h-4 w-4" fill="currentColor" viewBox="0 0 2
                                        <path fill-rule="evenodd" d="M16.338 16.338H13.67V12.16
                                    </svg>
                                    LinkedIn
                                </a>
                            {% endif %}
                            {% if profile_user.profile.twitter %}
                                <a href="{{ profile_user.profile.twitter }}" target="_blank" cla
                                    <svg class="mr-2 h-4 w-4" fill="currentColor" viewBox="0 0 2
                                        <path d="M6.29 18.251c7.547 0 11.675-6.253 11.675-11.675
                                    </svg>
                                    Twitter
                                </a>
                            {% endif %}
                        </div>

                        {% if not profile_user.profile.website and not profile_user.profile.gith
                            <p class="text-gray-500 italic text-sm">No links available.</p>
                        {% endif %}
                    </div>
                </div>
```

```
                <!-- Account Info -->
                <div class="mt-8 card">
                    <div class="card-header">
                        <h3 class="text-lg leading-6 font-medium text-gray-900">Account</h3>
                    </div>
                    <div class="card-body">
                        <div class="space-y-3">
                            <div>
                                <dt class="text-sm font-medium text-gray-500">Member since</dt>
                                <dd class="mt-1 text-sm text-gray-900">{{ profile_user.created_
                            </div>
                            <div>
                                <dt class="text-sm font-medium text-gray-500">Last updated</dt>
                                <dd class="mt-1 text-sm text-gray-900">{{ profile_user.updated_
                            </div>
                            <div>
                                <dt class="text-sm font-medium text-gray-500">Status</dt>
                                <dd class="mt-1">
                                    {% if profile_user.is_verified %}
                                        <span class="inline-flex items-center px-2.5 py-0.5 rou
                                            Verified
                                        </span>
                                    {% else %}
                                        <span class="inline-flex items-center px-2.5 py-0.5 rou
                                            Unverified
                                        </span>
                                    {% endif %}
                                </dd>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
{% endblock %}
```

Step 26: Profile Edit Template

`templates/accounts/profile_edit.html`:

```
{% extends 'base.html' %}
{% load static %}
{% load crispy_forms_tags %}

{% block title %}Edit Profile - Django Tailwind Starter{% endblock %}

{% block content %}
<div class="min-h-screen bg-gray-50 py-8">
    <div class="max-w-3xl mx-auto px-4 sm:px-6 lg:px-8">
        <!-- Header -->
        <div class="mb-8">
            <h1 class="text-2xl font-bold text-gray-900">Edit Profile</h1>
            <p class="mt-1 text-sm text-gray-600">Update your profile information and settings.<
        </div>

        <form method="post" enctype="multipart/form-data" class="space-y-8">
            {% csrf_token %}

            <!-- Personal Information -->
            <div class="card">
                <div class="card-header">
                    <h3 class="text-lg leading-6 font-medium text-gray-900">Personal Informatio
                    <p class="mt-1 text-sm text-gray-500">Update your personal details.</p>
                </div>
                <div class="card-body space-y-6">
                    <div class="grid grid-cols-1 gap-6 sm:grid-cols-2">
                        <div>
                            {{ form.first_name|as_crispy_field }}
                        </div>
                        <div>
                            {{ form.last_name|as_crispy_field }}
                        </div>
                    </div>

                    <div>
                        {{ form.email|as_crispy_field }}
                    </div>

                    <div>
                        {{ form.bio|as_crispy_field }}
                    </div>

                    <div class="grid grid-cols-1 gap-6 sm:grid-cols-2">
                        <div>
                            {{ form.phone|as_crispy_field }}
                        </div>
                        <div>
                            {{ form.birth_date|as_crispy_field }}
                        </div>
                    </div>

                    <div>
                        <label class="block text-sm font-medium text-gray-700 mb-2">Profile Pic
                        <div class="flex items-center space-x-6">
                            <div class="shrink-0">
                                <img class="h-16 w-16 object-cover rounded-full" src="{{ user.a
                            </div>
                            <div class="flex-1">
                                {{ form.avatar }}
                                <p class="mt-1 text-sm text-gray-500">JPG, GIF or PNG. 1MB max.
                            </div>
                        </div>
                    </div>
                </div>
            </div>

            <!-- Professional Information -->
            <div class="card">
                <div class="card-header">
                    <h3 class="text-lg leading-6 font-medium text-gray-900">Professional Informa
                    <p class="mt-1 text-sm text-gray-500">Update your work and professional det
                </div>
                <div class="card-body space-y-6">
                    <div class="grid grid-cols-1 gap-6 sm:grid-cols-2">
                        <div>
                            {{ profile_form.company|as_crispy_field }}
                        </div>
                        <div>
                            {{ profile_form.job_title|as_crispy_field }}
                        </div>
                    </div>

                    <div>
                        {{ profile_form.location|as_crispy_field }}
                    </div>
                </div>
```

```
            </div>

            <!-- Social Links -->
            <div class="card">
                <div class="card-header">
                    <h3 class="text-lg leading-6 font-medium text-gray-900">Social Links</h3>
                    <p class="mt-1 text-sm text-gray-500">Add your social media and website lin
                </div>
                <div class="card-body space-y-6">
                    <div class="grid grid-cols-1 gap-6 sm:grid-cols-2">
                        <div>
                            {{ profile_form.website|as_crispy_field }}
                        </div>
                        <div>
                            {{ profile_form.github|as_crispy_field }}
                        </div>
                    </div>

                    <div class="grid grid-cols-1 gap-6 sm:grid-cols-2">
                        <div>
                            {{ profile_form.linkedin|as_crispy_field }}
                        </div>
                        <div>
                            {{ profile_form.twitter|as_crispy_field }}
                        </div>
                    </div>
                </div>
            </div>

            <!-- Form Actions -->
            <div class="flex justify-end space-x-3">
                <a href="{% url 'accounts:profile' user.pk %}" class="btn btn-secondary">
                    Cancel
                </a>
                <button type="submit" class="btn btn-primary">
                    Save Changes
                </button>
            </div>
        </form>
    </div>
</div>
{% endblock %}
```

Step 27: Additional Components

Create `templates/components/alerts.html` :

```
{% if message.tags == 'success' %}
    <div class="rounded-md bg-green-50 p-4">
        <div class="flex">
            <div class="flex-shrink-0">
                <svg class="h-5 w-5 text-green-400" viewBox="0 0 20 20" fill="currentColor">
                    <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000 16zm3.707-9.293a1
                </svg>
            </div>
            <div class="ml-3">
                <p class="text-sm font-medium text-green-800">{{ message }}</p>
            </div>
        </div>
    </div>
{% elif message.tags == 'error' %}
    <div class="rounded-md bg-red-50 p-4">
        <div class="flex">
            <div class="flex-shrink-0">
                <svg class="h-5 w-5 text-red-400" viewBox="0 0 20 20" fill="currentColor">
                    <path fill-rule="evenodd" d="M10 18a8 8 0 100-16 8 8 0 000 16zM8.707 7.293a1
                </svg>
            </div>
            <div class="ml-3">
                <p class="text-sm font-medium text-red-800">{{ message }}</p>
            </div>
        </div>
    </div>
{% elif message.tags == 'warning' %}
    <div class="rounded-md bg-yellow-50 p-4">
        <div class="flex">
            <div class="flex-shrink-0">
                <svg class="h-5 w-5 text-yellow-400" viewBox="0 0 20 20" fill="currentColor">
                    <path fill-rule="evenodd" d="M8.257 3.099c.765-1.36 2.722-1.36 3.486 0l5.58
                </svg>
            </div>
            <div class="ml-3">
                <p class="text-sm font-medium text-yellow-800">{{ message }}</p>
            </div>
        </div>
    </div>
{% else %}
    <div class="rounded-md bg-blue-50 p-4">
        <div class="flex">
            <div class="flex-shrink-0">
```

```
                <svg class="h-5 w-5 text-blue-400" viewBox="0 0 20 20" fill="currentColor">
                    <path fill-rule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 
                </svg>
            </div>
            <div class="ml-3">
                <p class="text-sm font-medium text-blue-800">{{ message }}</p>
            </div>
        </div>
    </div>
{% endif %}
```

Create `templates/components/footer.html` :

```
<footer class="bg-white border-t border-gray-200">
    <div class="max-w-7xl mx-auto py-12 px-4 sm:px-6 lg:px-8">
        <div class="grid grid-cols-1 md:grid-cols-4 gap-8">
            <div class="col-span-1 md:col-span-2">
                <div class="flex items-center">
                    <span class="text-xl font-bold text-gray-900">Django Starter</span>
                </div>
                <p class="mt-4 text-gray-600 text-sm">
                    A modern Django starter template with Tailwind CSS, complete user management
                </p>
            </div>

            <div>
                <h3 class="text-sm font-semibold text-gray-400 tracking-wider uppercase">Navigat
                <ul class="mt-4 space-y-4">
                    <li><a href="{% url 'home' %}" class="text-base text-gray-500 hover:text-gra
                    <li><a href="{% url 'about' %}" class="text-base text-gray-500 hover:text-gr
                    {% if user.is_authenticated %}
                        <li><a href="{% url 'dashboard' %}" class="text-base text-gray-500 hover
                    {% endif %}
                </ul>
            </div>

            <div>
                <h3 class="text-sm font-semibold text-gray-400 tracking-wider uppercase">Account
                <ul class="mt-4 space-y-4">
                    {% if user.is_authenticated %}
                        <li><a href="{% url 'accounts:profile' user.pk %}" class="text-base text
                        <li><a href="{% url 'account_logout' %}" class="text-base text-gray-500
                    {% else %}
                        <li><a href="{% url 'account_login' %}" class="text-base text-gray-500 h
                        <li><a href="{% url 'account_signup' %}" class="text-base text-gray-500
                    {% endif %}
                </ul>
            </div>
        </div>

        <div class="mt-8 pt-8 border-t border-gray-200">
            <p class="text-base text-gray-400 text-center">
                &copy; 2025 Django Starter. Built with Django and Tailwind CSS.
            </p>
        </div>
```

```
      </div>
  </footer>
```

Step 28: JavaScript and Additional Files

Create `static/js/main.js` :

```
// Theme Toggle Functionality
document.addEventListener('DOMContentLoaded', function() {
    // Theme toggle
    const themeToggle = document.getElementById('theme-toggle');
    const html = document.documentElement;

    if (themeToggle) {
        themeToggle.addEventListener('click', function() {
            if (html.classList.contains('dark')) {
                html.classList.remove('dark');
                localStorage.setItem('theme', 'light');
            } else {
                html.classList.add('dark');
                localStorage.setItem('theme', 'dark');
            }
        });
    }

    // Check for saved theme preference or default to light
    const savedTheme = localStorage.getItem('theme');
    if (savedTheme === 'dark') {
        html.classList.add('dark');
    }

    // Mobile menu toggle
    const mobileMenuButton = document.getElementById('mobile-menu-button');
    const mobileMenu = document.getElementById('mobile-menu');

    if (mobileMenuButton && mobileMenu) {
        mobileMenuButton.addEventListener('click', function() {
            mobileMenu.classList.toggle('hidden');
        });
    }

    // Auto-hide alerts after 5 seconds
    const alerts = document.querySelectorAll('.alert');
    alerts.forEach(function(alert) {
        setTimeout(function() {
            alert.style.opacity = '0';
            setTimeout(function() {
                alert.remove();
            }, 300);
        }, 5000);
    });

    // File input preview
    const avatarInput = document.querySelector('input[type="file"][name="avatar"]');
    if (avatarInput) {
        avatarInput.addEventListener('change', function(e) {
            const file = e.target.files[0];
            if (file) {
                const reader = new FileReader();
                reader.onload = function(e) {
                    const preview = document.querySelector('img[alt*="' + document.querySelector
                    if (preview) {
                        preview.src = e.target.result;
                    }
                };
                reader.readAsDataURL(file);
            }
        });
    }
});

// Utility functions
function showNotification(message, type = 'info') {
    const notification = document.createElement('div');
    notification.className = `fixed top-4 right-4 p-4 rounded-lg shadow-lg z-50 ${
        type === 'success' ? 'bg-green-500' :
        type === 'error' ? 'bg-red-500' :
        type === 'warning' ? 'bg-yellow-500' : 'bg-blue-500'
    } text-white`;
    notification.textContent = message;

    document.body.appendChild(notification);

    setTimeout(() => {
        notification.remove();
    }, 3000);
}

// Form validation helpers
function validateForm(form) {
    const requiredFields = form.querySelectorAll('[required]');
    let isValid = true;
    requiredFields.forEach(field => {
        if (!field.value.trim()) {
            field.classList.add('border-red-500');
            isValid = false;
        } else {
            field.classList.remove('border-red-500');
        }
    });

    return isValid;
}
```

Step 29: Admin Configuration

Update `apps/accounts/admin.py` :

```python
from django.contrib import admin
from django.contrib.auth.admin import UserAdmin as BaseUserAdmin
from django.contrib.auth import get_user_model
from .models import Profile

User = get_user_model()

class ProfileInline(admin.StackedInline):
    model = Profile
    can_delete = False
    verbose_name_plural = 'Profile'

class UserAdmin(BaseUserAdmin):
    inlines = (ProfileInline,)
    list_display = ('email', 'first_name', 'last_name', 'is_verified', 'is_staff', 'created_at'
    list_filter = ('is_staff', 'is_superuser', 'is_active', 'is_verified', 'created_at')
    search_fields = ('email', 'first_name', 'last_name')
    ordering = ('-created_at',)

    fieldsets = BaseUserAdmin.fieldsets + (
        ('Personal info', {'fields': ('bio', 'avatar', 'phone', 'birth_date')}),
        ('Status', {'fields': ('is_verified',)}),
        ('Important dates', {'fields': ('created_at', 'updated_at')}),
    )

    readonly_fields = ('created_at', 'updated_at')

    add_fieldsets = (
        (None, {
            'classes': ('wide',),
            'fields': ('email', 'username', 'first_name', 'last_name', 'password1', 'password2'
        }),
    )

admin.site.unregister(User)
admin.site.register(User, UserAdmin)

@admin.register(Profile)
class ProfileAdmin(admin.ModelAdmin):
    list_display = ('user', 'company', 'job_title', 'location')
    list_filter = ('company', 'location')
    search_fields = ('user__email', 'user__first_name', 'user__last_name', 'company')
```

Step 30: Additional Settings and Configuration

Create `.gitignore` :

```
# Django
*.log
*.pot
*.pyc
__pycache__/
local_settings.py
db.sqlite3
db.sqlite3-journal
media/

# Virtual Environment
venv/
env/
ENV/

# IDE
.vscode/
.idea/
*.swp
*.swo

# OS
.DS_Store
.DS_Store?
._*
.Spotlight-V100
.Trashes
ehthumbs.db
Thumbs.db

# Node.js
node_modules/
npm-debug.log*
yarn-debug.log*
yarn-error.log*

# Static files
staticfiles/
static/css/output.css

# Environment variables
.env
.env.local
```

```
.env.development.local
.env.test.local
.env.production.local

# Coverage reports
htmlcov/
.coverage
.coverage.*
coverage.xml
*.cover
.hypothesis/
.pytest_cache/
```

Create `requirements.txt`:

```
Django==4.2.7
Pillow==10.1.0
python-decouple==3.8
django-crispy-forms==2.1
crispy-tailwind==0.5.0
django-allauth==0.57.0
django-debug-toolbar==4.2.0
```

Create `.env.example`:

```
# Django Settings
SECRET_KEY=your-secret-key-here
DEBUG=True
ALLOWED_HOSTS=localhost,127.0.0.1

# Database
DATABASE_URL=sqlite:///db.sqlite3

# Email Settings
EMAIL_BACKEND=django.core.mail.backends.console.EmailBackend
EMAIL_HOST=smtp.gmail.com
EMAIL_PORT=587
EMAIL_USE_TLS=True
EMAIL_HOST_USER=your-email@gmail.com
EMAIL_HOST_PASSWORD=your-app-password

# Static/Media Files
STATIC_URL=/static/
MEDIA_URL=/media/

# Security (for production)
SECURE_SSL_REDIRECT=False
SECURE_PROXY_SSL_HEADER=None
```

Step 31: Deployment Configuration

Create `Procfile` for Heroku:

```
web: gunicorn config.wsgi:application
release: python manage.py migrate
```

Create `runtime.txt` :

```
python-3.11.0
```

Update `requirements.txt` for production:

```
Django==4.2.7
Pillow==10.1.0
python-decouple==3.8
django-crispy-forms==2.1
crispy-tailwind==0.5.0
django-allauth==0.57.0
django-debug-toolbar==4.2.0
gunicorn==21.2.0
whitenoise==6.6.0
psycopg2-binary==2.9.9
dj-database-url==2.1.0
```

Step 32: Final Setup Commands

```
# Create directories
mkdir -p media/profiles
mkdir -p static/images

# Add default profile image
# Download a default avatar image and place it at media/profiles/default.jpg

# Make migrations
python manage.py makemigrations accounts
python manage.py makemigrations

# Apply migrations
python manage.py migrate

# Collect static files
python manage.py collectstatic --noinput

# Create superuser
python manage.py createsuperuser

# Build CSS (in separate terminal)
npm run build-css

# Run server
python manage.py runserver
```

🎉 Features Summary

This complete Django + Tailwind CSS starter project includes:

✅ User Management

- Custom User model with extended fields
- Profile management with social links
- Avatar upload with automatic resizing
- Email verification and authentication
- Password reset functionality

✅ Modern UI/UX

- Tailwind CSS with custom components
- Responsive design for all devices
- Beautiful forms with Crispy Forms
- Alert notifications system
- Dark/Light theme support

✅ Development Features

- Hot reload for CSS compilation
- Debug toolbar for development
- Environment variable configuration
- Comprehensive admin interface
- Form validation and error handling

✅ Production Ready

- Security best practices
- Static file handling
- Media file management
- Database migrations
- Deployment configurations

🚀 Quick Start

1. Clone or download the project

2. Install Python and Node.js dependencies
3. Configure environment variables
4. Run migrations and create superuser
5. Build CSS and start development server

This starter project provides a solid foundation for building modern Django applications with beautiful UI components and comprehensive user management!