User Accounts

We will see how to manage users, Django already ships with a robust user authentication system, plus the implementation is also easy.

New Project

Let's start with a blank slate, create a new project called 'users_project' and within the project create two apps 'pages' and 'users'

'pages' app will serve our regular home page, while 'users' app will handle user management.

```
$ mkdir users
$ cd users
$ pipenv install django
$ pipenv shell
$ django-admin startproject users_project .
$ python manage.py startapp pages
$ python manage.py startapp users
```

Register the app in project settings -

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'pages.apps.PagesConfig',  # new
    'users.apps.UsersConfig'  # new
]
```

And finally complete our migrations

```
$ python manage.py migrate
```

Basic Setup

Let's add our templates and add bootstrap for our basic styling.

As Django provides all the necessary models and urls, we can simply create our views and templates.

User Signup

We don't require any views for login, authenticate & logout, we just use the default Django functionality, however we need to create a view for user sign up.

We define a custom flow to sign up users -

1. Get user sign up data
2. Create user
3. Authenticate and login the user
4. Redirect to our home page

It's not necessary to always follow this work flow, you can always create a custom work flow depending on the project requirements.

```
from django.shortcuts import render, redirect
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import authenticate, login


def sign_up(request):
    # Check request method
    if request.method == 'POST':
        form = UserCreationForm(request.POST)
        # Check for form validity
        if form.is_valid():
            # Save user
            form.save()
            # Get user credentials
            username = form.cleaned_data.get('username')
            raw_password = form.cleaned_data.get('password1')
            # Authenticate user
            user = authenticate(request, username=username,
password=raw_password)
            if user is not None:
                print(f'User: {username}\tPassword: {raw_password}')
                # Login user
                login(request, user)
                return redirect('home')
    else:
        form = UserCreationForm()
    context = {
        'form': form
    }
    return render(request, 'registration/signup.html', context)
```

Rest we plug in all the built-in urls and views offered by Django.