

Document: Django Exercise 2.1

1. Why Django is Popular Among Web Developers:

Django is widely popular due to its "batteries-included" approach, meaning it provides many built-in features that simplify web development. Its robust Model-View-Template (MVT) architecture helps in the rapid development of complex, database-driven websites. Furthermore, Django's strong emphasis on security, scalability, and the "Don't Repeat Yourself" (DRY) principle makes it an ideal choice for professional and large-scale applications.

2. Companies Using Django:

1. **Instagram** - Social media platform; uses Django to handle massive user data and enhance scalability.
2. **Spotify** - Music streaming service; employs Django for content management and a responsive interface.
3. **Pinterest** - Visual discovery and social media platform; leverages Django for scalability and fast prototyping.
4. **Mozilla** - Browser and internet software; uses Django for various internal tools and projects.
5. **NASA** - Space research organization; utilizes Django for handling large datasets and secure applications.

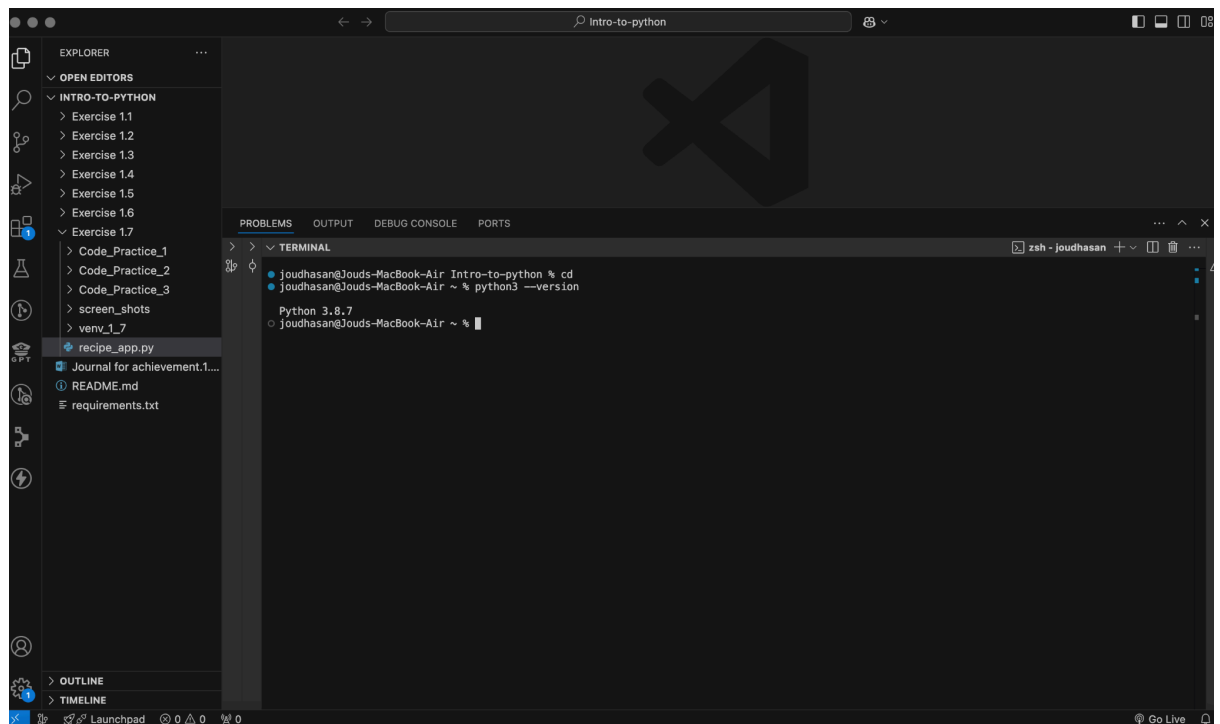
3. Scenario Analysis:

- **Developing a web application with multiple users:** Yes, Django is a good fit here because it has robust built-in tools for user authentication, session management, and database interaction, essential for managing multiple users.
- **Fast deployment with ongoing changes:** Yes, Django's MVT architecture and modular structure allow for rapid deployment and iterative development, making it easy to incorporate changes.
- **Basic application without database access or file operations:** No, Django might be too heavy for this scenario since it includes many built-in features for database management that are unnecessary for a basic application. A simpler framework like Flask might be more suitable.

- **Building an application from scratch with full control over functionality:** No, if you want deep control, Django's structured "Django way" might feel restrictive. A lightweight framework like Flask would offer more freedom for custom configurations.
- **Starting a large project with additional support needs:** Yes, Django is suitable here because of its large community, extensive documentation, and pre-built functionalities, which make it easier to find resources and support.

4. Python Installation and Version Check:

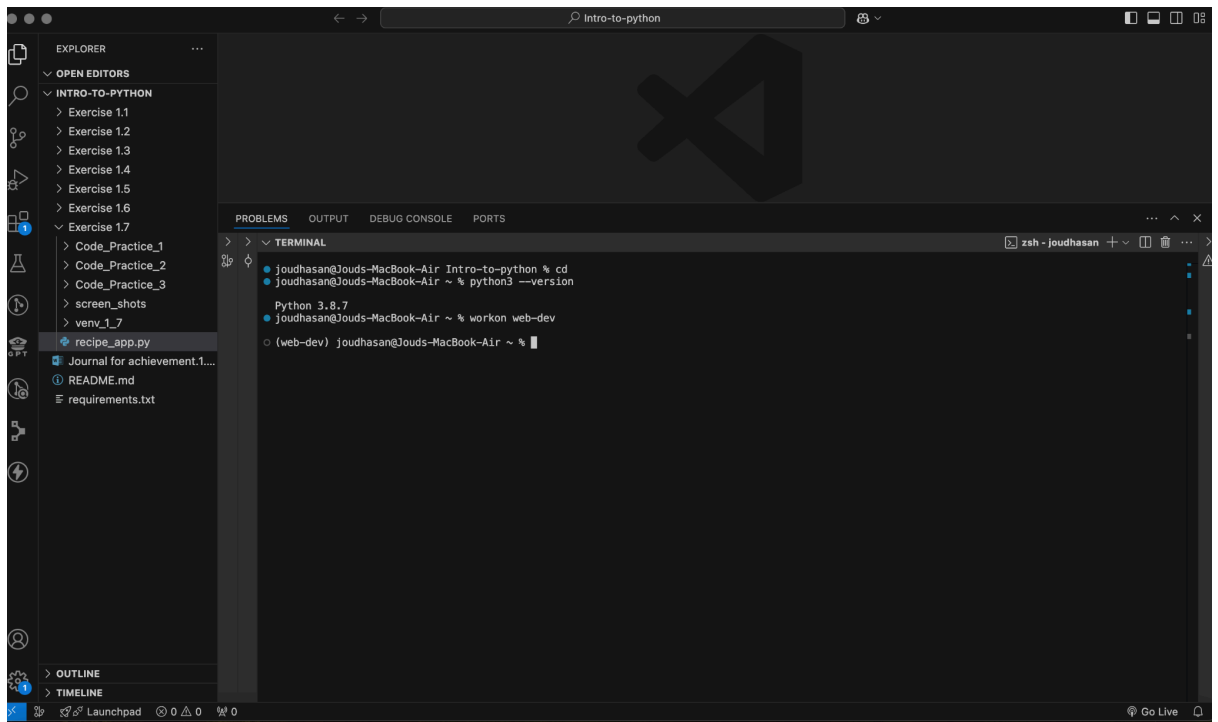
- **Command:** `python3 --version`
- **Screenshot:**



5. Virtual Environment Setup (achievement2-practice):

- **Commands:**
 - `mkvirtualenv achievement2-practice`
 - `workon achievement2-practice`

- **Screenshot:**



6. Django Installation Verification:

- **Commands:**

- `pip install django`
- `django-admin --version`

- **Screenshot:**

