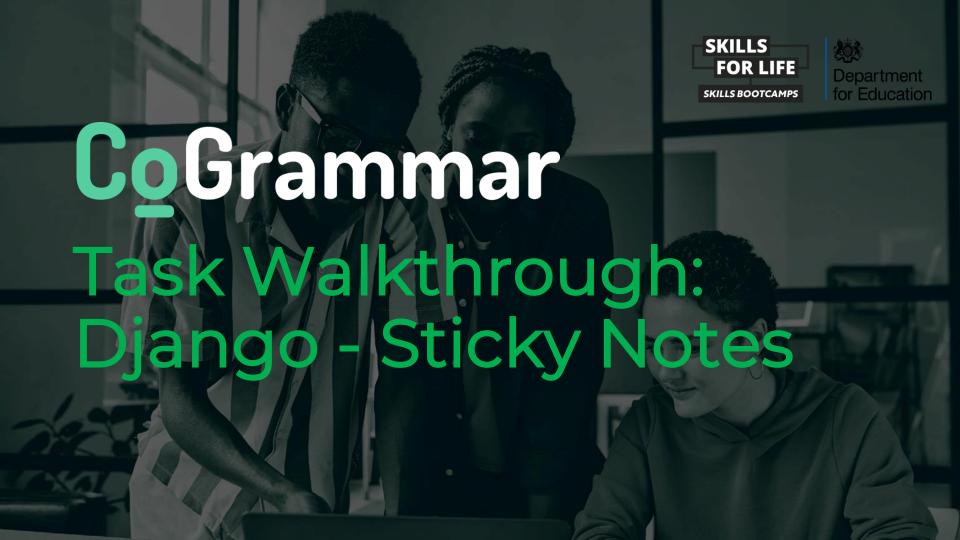
# Welcome to this CoGrammar Task Walkthrough: Task 24/26

The session will start shortly...

Questions? Drop them in the chat.







#### Software Engineering Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly.
   (Fundamental British Values: Mutual Respect and Tolerance)
- No question is daft or silly ask them!
- There are Q&A sessions throughout this session, should you wish to ask any follow-up questions.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: <u>Questions</u>

#### Software Engineering Session Housekeeping cont.

- For all non-academic questions, please submit a query: www.hyperiondev.com/support
- Report a safeguarding incident: www.hyperiondev.com/safeguardreporting
- We would love your **feedback** on lectures: <u>Feedback on Lectures</u>

#### Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:



Ian Wyles Designated Safeguarding Lead



Simone Botes



Nurhaan Snyman



Rafig Manan

Scan to report a safeguarding concern



or email the Designated Safeguarding Lead: Ian Wyles safeguarding@hyperiondev.com



Ronald Munodawafa



## Learning Outcomes

- Recall the core components of Django (models, views, templates, URLs).
- Describe the relationship between models, views, and templates in a CRUD app.
- Apply the CRUD functionality and MVT pattern to create a workout APP.
- Design and develop a complete Django application with full CRUD functionality.



#### Polls

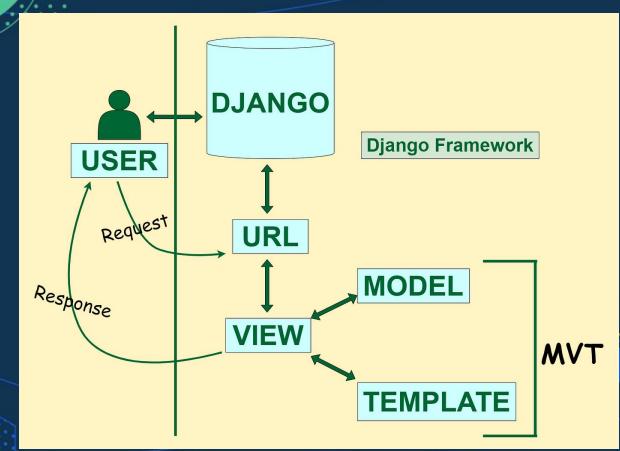
- · Refer to the polls section to vote for you option.
- 1. How confident are you in setting up a Django project and app?
  - a) Not confident at all.
  - b) Somewhat confident.
  - c) Very confident.



## Task Walkthrough: Django

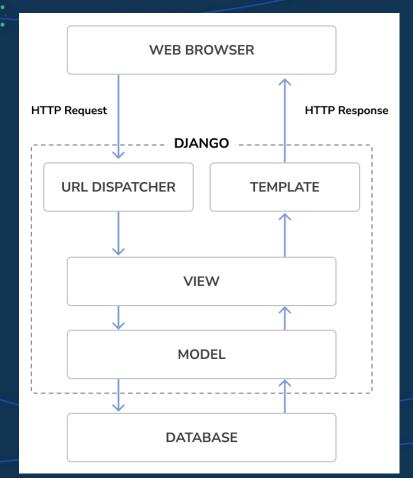


### MVI



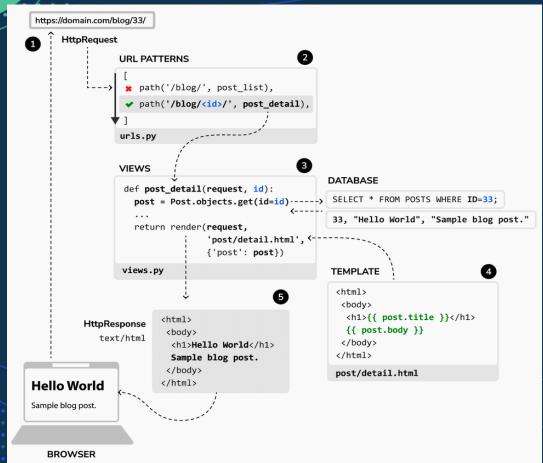
CoGrammar

## ANGO PROCESS FLOW





### HOW THE CODE CONNECTS





## Task 24







#### **Practical task**

#### Part 1

Follow these steps:

- Use the design techniques and principles you have learnt in the bootcamp to design your sticky notes application. Please remember to submit your diagrams alongside your project. Feel free to review the Software Design task if you need a refresher. You will need to submit at least the following:
  - Use case diagram
  - o Sequence diagram
  - Class diagram
- 2. Create a new Django project called sticky\_notes.
- Install your app in settings.py. Configure your Django project settings to include the newly created app.
- Design and define the model for your sticky notes application, including fields for the title and content of each note.
- Implement and run database migrations to ensure that your model is reflected in the database.
- Develop views that handle the CRUD functionality for your sticky notes application, allowing users to create, read, update, and delete notes.



- 7. Build forms that facilitate the input and editing of sticky notes content.
- 8. Set up URL patterns to map the different views of your sticky notes application. Ensure that the URLs of your sticky notes app are appropriately included in the main **urls.py** file.
- 9. Create a base HTML template that can be extended by other templates in your application.
- 10. Design HTML templates for displaying sticky notes content, integrating them with your views.
- Style your sticky notes application by adding CSS to enhance the visual appeal. If you are familiar with Bootstrap you are welcome to use it to speed up your CSS development.
- Configure your Django project to collect static files, including CSS, into a designated directory.
- 13. Rerun migrations to ensure that any changes are reflected in the database, and start the development server to test your application.
- 14. Please **exclude** the **venv** folder when you submit your Django project; it is unnecessary, as the mentor reviewing your work can generate it. It is good practice to always delete generated files since they may make it difficult to set up your app on other machines and they take up a lot of storage space.



#### Part 2

Create a text document, markdown file, or PDF called research\_answers to record your findings for the following:

- Conduct research to understand how HTTP applications preserve the state
  of an application across multiple request-response cycles, especially
  concerning user authentication and session management. Write a
  paragraph describing your findings.
- Investigate and document the procedures for performing Django database migrations to a server-based relational database like MariaDB. Write a paragraph describing your findings.



## Summary of the Task Steps

- 1. Set Up the Django Project and App:
  - Create a Django project: django-admin startproject projectname.
- 2. Define Models:
  - o Create a model in models.py (e.g., Post for a blog).
- 3. Create Views:
  - o Write view functions in views.py for CRUD operation.
- 4. Build Templates
  - Create HTML templates in a templates folder.
- 5. Map URLs
  - Map the url in the application urls.py file



## Task 26









#### **Practical task**

In this task, you will further develop your sticky notes application from the previous task and test it. During this task, you are welcome to refactor your application and improve it if you feel there are any features you wish to improve upon. Please follow these steps:

- Design and implement tests to ensure they cover the use cases of the sticky notes application.
- 2. Run the tests you've created to validate the correctness of your implementation.
- 3. Ensure that you also manually test your application to ensure that there are no obvious bugs.



- Your completed sticky notes task manager application should be saved in a folder in your task folder. In this folder, you will need to include all the components of the project, including:
  - All your design diagrams
  - Unit tests
  - o Project directory for the final app, including the source code
- Please exclude the venv folder; it is unnecessary as the mentor reviewing your work can generate it. It's good practice to always delete generated files since they may make it difficult to set up your app on other machines and they take up a lot of storage space.
- Add your completed sticky notes application to your GitHub account as part
  of your developer portfolio. Ensure your repository is public and remember
  to exclude the virtual environment folder before uploading. Share a link to
  your sticky notes repository in a text file called sticky\_github.txt.



### Polls

- · Refer to the polls section to vote for you option.
- 1. Can you explain the MVT (Model-View-Template) pattern in Django?
  - a) No, I'm still unclear.
  - b) I understand the basics but need more practice.
  - c) Yes, I can explain it clearly.



## **Questions and Answers**



Thank you for attending





