**🛒 1. Shopmart – E-commerce Website**

**How to say:**

“This is an online shopping website. I used Django, Python, MySQL, HTML, CSS, and JavaScript.

In this project, users can register, login, browse products, add to cart, and place orders.

I created product models, views, and templates. I used MySQL to store user and product data. The UI is responsive and user-friendly.”

**👨‍💼 2. Prime – Employee Management System**

**How to say:**

“This is a web application for managing employees in a company. I used Django and MySQL.

There are two login types – Employee and Team Leader. After login, dashboard shows employee details.

I created leave management, attendance tracking, approval system using Django views and templates.

Admin or Team Leader can search and update employee info.”

**❤️ 3. Heart Rate Monitoring System (API-based Project)**

**How to say:**

“This project is an API-based system to record and manage heart rate data. I used Django REST Framework and Postman.

I created REST APIs for Create, Read, Update, Delete (CRUD) heart rate records.

I used MySQL as the database and tested API using Postman. This is a backend-only project, no frontend.”

**🤖 Bonus: Real-time Needle Detection (Company Project)**

**How to say:**

“In my previous company, I worked on a machine learning project using YOLO and OpenCV.

It detects needle movement in garment machines – like pick-up, stitching, and drop time.

This helps in tracking machine performance. I used MongoDB to save real-time data and Python for backend.”

**✅ EC2 (Elastic Compute Cloud)**

“EC2 is a virtual server in the cloud from AWS.  
It helps us run websites and applications just like our local system.”

**✅ Linux**

“Linux is an operating system like Windows.  
It is open-source and mostly used in servers because it's secure and fast.”

**✅ Ubuntu**

“Ubuntu is one type of Linux OS.  
It is easy to use and mostly used for web development in cloud servers.”

**✅ Nginx**

“Nginx is a web server.  
It receives requests from users (like browser) and forwards them to the backend app (like Django).”

(Bonus if they ask: It's also used for load balancing and serving static files.)

**✅ Gunicorn**

“Gunicorn is a Python WSGI server.  
It runs the Django app and connects it to Nginx for production deployment.”

**✅ SSH (Secure Shell)**

“SSH is used to connect to the cloud server securely from our local computer using command line.”

**✅ Virtualenv**

“Virtualenv is used to create a separate Python environment for each project.  
It keeps all libraries isolated.”

**✅ MySQL**

“MySQL is a relational database.  
It stores data like users, products, orders in tables with rows and columns.”

**✅ Postman**

“Postman is a tool to test APIs.  
We can send requests like GET, POST, PUT to our backend and check the response.”

**✅ Django All Files – One-Line Purpose**

| **🔹 File / Folder** | **🔸 One-Line Explanation** |
| --- | --- |
| manage.py | Django project command-line tool to run server, migrations, etc. |
| settings.py | Main configuration file for project settings (DB, apps, static files). |
| urls.py | Maps URLs to views (routing system). |
| wsgi.py | Entry point for WSGI servers (like Gunicorn) to serve the Django app. |
| asgi.py | Entry point for ASGI servers (for async apps like WebSockets). |
| \_\_init\_\_.py | Marks the folder as a Python package. |

**✅ Inside App Folder (e.g., myapp/)**

| **🔹 App File** | **🔸 One-Line Explanation** |
| --- | --- |
| models.py | Defines database models (tables) using Django ORM. |
| views.py | Contains the logic to handle requests and return responses. |
| urls.py | App-level URL routing (optional, but good practice). |
| admin.py | Registers models for Django admin panel. |
| apps.py | Stores app configuration info. |
| forms.py | Defines Django forms for input handling (optional). |
| tests.py | Write test cases for app features. |
| migrations/ | Stores migration files to track database schema changes. |
| templates/ | Stores HTML files for frontend rendering. |
| static/ | Stores static files like CSS, JavaScript, and images. |