

Essential Tools for Your Project

1. Development Tools

- **Python** – Main programming language for Django.
- **Django** – Web framework for building the application.
- **Django REST Framework (DRF)** – If you need to create APIs.
- **Visual Studio Code (VS Code) / PyCharm** – Code editors for development.
- **Git & GitHub/GitLab/Bitbucket** – Version control and collaboration.

2. Database Management

- **SQLite** (default in Django, good for small projects)
- **PostgreSQL / MySQL** (better for production and scalability)
- **Django ORM** – For interacting with the database using Python.

3. Security & Authentication

- **Django Allauth** – For user authentication (email/password, social login).
- **Django Middleware** – For security protections like CSRF and XSS.
- **Django Environ** – To manage environment variables securely.

4. Frontend Tools (if applicable)

- **HTML, CSS, JavaScript** – Basic frontend technologies.
- **Bootstrap / Tailwind CSS** – For responsive UI design.
- **React.js or Vue.js** – If you want a more dynamic frontend.

5. Deployment & Hosting

- **Gunicorn / uWSGI** – For running Django on a server.
- **NGINX / Apache** – As a web server.
- **Heroku / AWS / DigitalOcean** – For cloud deployment.
- **Docker** – If you need containerized deployment.

6. Additional Libraries & Features

- **Celery & Redis** – For background tasks (e.g., email notifications).
- **Django Crispy Forms** – For better form handling.
- **Django Debug Toolbar** – Helps with debugging during development.
- **Stripe / PayPal SDK** – If you need online payments.

Would you like recommendations for a specific feature in your project? 😊

You said:

hey chat gpt i am student of final year my project is property rental management system and i donot know anything about python and django now what are the basic o f paython and django basic structure i should prepared for viva

ChatGPT said:

Memory updated

No worries! Since you're new to Python and Django, I'll break down the **basics** you need to understand for your **viva** and project.

1. Python Basics (Before Learning Django)

Python is a beginner-friendly language. Here are some fundamental concepts:

a) Variables & Data Types

```
python
CopyEdit
name = "John"      # String
age = 25           # Integer
height = 5.8       # Float
is_student = True  # Boolean
```

b) Control Flow (if-else, loops)

```
python
CopyEdit
# If-else statement
if age > 18:
    print("Adult")
else:
    print("Minor")

# Loop example
for i in range(5):
    print(i)
```

c) Functions

```
python
CopyEdit
def greet(name):
    return f"Hello, {name}!"

print(greet("Alice"))
```

d) Object-Oriented Programming (OOP)

Django is built around OOP concepts, so understand **classes and objects**.

```
python
CopyEdit
class Car:
    def __init__(self, brand, model):
        self.brand = brand
        self.model = model

    def show_info(self):
        return f"{self.brand} - {self.model}"

car1 = Car("Toyota", "Corolla")
print(car1.show_info())
```

2. Django Basics (Framework for Web Development)

a) Django Project Structure

When you create a Django project, it generates this structure:

```
bash
CopyEdit
/property_rental_management_system # Main project folder
|-- manage.py                     # Command-line utility
|-- settings.py                   # Project configurations
|-- urls.py                       # URL routing
|-- wsgi.py / asgi.py             # For deploying Django app
|-- apps/                         # Django apps (features)
|   |-- models.py                 # Defines database tables
|   |-- views.py                  # Handles logic
|   |-- templates/                # HTML files
|   |-- static/                   # CSS, JS, images
```

b) Creating a Django Project & App

Run these commands:

```
sh
CopyEdit
# Install Django
pip install django

# Create a new Django project
django-admin startproject property_rental_management_system

# Move into project folder
cd property_rental_management_system
```

```
# Create an app (e.g., for handling properties)
python manage.py startapp rentals
```

c) Models (Database Table)

In `models.py`, define a model (table) for properties:

```
python
CopyEdit
from django.db import models

class Property(models.Model):
    title = models.CharField(max_length=100)
    description = models.TextField()
    price = models.DecimalField(max_digits=10, decimal_places=2)
    location = models.CharField(max_length=100)

    def __str__(self):
        return self.title
```

Then, run:

```
sh
CopyEdit
python manage.py makemigrations
python manage.py migrate
```

This will create the database table.

d) Views (Logic)

In `views.py`, create a function to show properties:

```
python
CopyEdit
from django.shortcuts import render
from .models import Property

def property_list(request):
    properties = Property.objects.all()
    return render(request, 'property_list.html', {'properties': properties})
```

e) URL Configuration

In `urls.py`, connect the view to a URL:

```
python
CopyEdit
from django.urls import path
from .views import property_list

urlpatterns = [
```

```
    path('properties/', property_list, name='property_list'),  
]
```

f) Running the Server

```
sh  
CopyEdit  
python manage.py runserver
```

Now, visit `http://127.0.0.1:8000/properties/` in your browser.

3. Important Topics for Viva

Here's what you should be prepared to answer:

- ◆ **What is Django?** → It's a Python framework for web development.
 - ◆ **What are Models, Views, and Templates (MVT Architecture)?**
 - ◆ **How does Django handle databases?** (Django ORM)
 - ◆ **How do you define URLs in Django?** (Using `urls.py`)
 - ◆ **How do you create a Django model and apply migrations?**
 - ◆ **How do you run a Django server?** (`python manage.py runserver`)
 - ◆ **What is Django Admin Panel?** (Use `python manage.py createsuperuser` to access it)
 - ◆ **How does Django ensure security?** (CSRF protection, authentication)
-

Next Steps

1. **Try running Django on your PC**
2. **Write a simple view, model, and URL configuration**
3. **Explore Django Admin Panel**
4. **Understand how Django templates work**

Since this is your **final year project**, I recommend **learning by building**. Start with small steps, and I'm here to help whenever you need guidance! 😊🚀