



# Frameworks Assignment

## Building a Web Application with Django and a Database

### How-To Guide

The Frameworks Assignment represents 12% of the overall course grade.

### Instructions

Develop a web application that leverages the Django framework integrated with a PostgreSQL database. This integration aims to ensure efficient data management, from storing user messages to capturing intricate project details. By combining modern HTML, CSS powered by Bootstrap, and dynamic JavaScript functionalities, the end goal is to deliver a captivating user experience.

### Project Details

#### 1. Application Structure

- Backend Framework: Django
- Database: PostgreSQL
- Frontend: HTML, CSS, JavaScript

#### 2. Core Features

- User Management
  - Registration and Login System
  - Allows users to update their personal and contact details.
  - Email-based recovery for forgotten passwords.

- **Data Storage**

- Store details such as project name, description, start and end dates, stakeholders, and status.
- Inbox functionality where users can send, receive, and archive messages.
- A system where users can store and categorize their data.

### **3. User Interface**

- Bootstrap-driven responsive layouts ensuring compatibility across devices of varying screen sizes.
- HTML templates for consistent page structures, such as headers, footers, and sidebars.
- JavaScript-driven content that reacts to user inputs, such as real-time form validation, pop-up notifications, and collapsible sections.

### **4. Security Features**

- Implement data encryption methodologies to ensure sensitive data, especially user passwords, are stored securely.
- Implement user roles and permissions to restrict access to specific functionalities or data based on the user's role.

### **5. Extension Capabilities**

- Modular design allowing for future integrations like third-party APIs, additional database systems, or incorporating additional frontend frameworks.

### **Development Strategy**

This assignment, while structured, offers freedom to decide on specifics, explore integrations, and experiment with design. It's an opportunity not just to showcase proficiency in web development but to express creativity and vision. With any large project it is a good idea to break it into phases. The phases below can be used as a guide.

Phase 1 - Setting up the Django environment.

Phase 2 - Designing the database schema, considering tables for users, projects, messages, and contacts.

Phase 3 - Building core backend functionalities such as user management and data storage operations.

Phase 4 - Crafting the frontend using HTML templates, enhancing the look with Bootstrap, and integrating JavaScript-driven functionalities.

Phase 5 - Rigorous testing for functionality, responsiveness, and security.

Phase 6 - Ensure your supplication is successfully hosted via a hosting provider such as Render.

## Preparing Your Submission

- Compress all your project files into a single ZIP file, including the Django app files and the deployment configuration files.
- Submit the ZIP file along with clear instructions on how to deploy and access the web app on your chosen hosting provider.
- Ensure that your code is properly structured and follows best practices.
- Implement additional features or design enhancements to showcase your creativity and skills.
- Use comments including docstrings to explain each non-trivial function, and a README file for an overview.
- Make sure your Django app is compatible with your chosen hosting providers requirements and configurations.
- Ensure that your web app is accessible and fully functional on the provided URL.
- Double-check the deployment instructions and any necessary configuration details to ensure a smooth deployment process.
- Include any additional details or considerations specific to the deployment in your submission.
- Verify the hosted web app's functionality and appearance after deployment to ensure it matches your local development version.

# How You Will Be Assessed

The following rubric describes how the essay will be assessed:

<b>1. Demonstrate an understanding and apply the key concepts of Django.</b>	<b>2. Show clear understanding of Database integration in Django.</b>	<b>3. Show clear understanding of integrating Authentication and Authorization with Django</b>	<b>4. Demonstrate clean code structure, including HTML templates, Bootstrap and JavaScript</b>	<b>5. Show evidence of a hosted Django app that is fully functional and accessible</b>
<b>Distinction Criteria</b>				
Demonstrates strong ability to apply the key concepts of Django.	Demonstrates strong ability to apply key concepts and principles of Database integration in Django.	Demonstrates a strong understanding of integrating Authentication and Authorization with Django.	Demonstrates a strong ability to create a clean code structure, including HTML templates, Bootstrap and JavaScript	Demonstrates strong evidence of a hosted Django app that is fully functional and accessible
<b>Merit Criteria</b>				
Able to apply the key concepts of Django.	Able to apply key concepts and principles of Database integration in Django.	Demonstrates a good understanding of integrating Authentication and Authorization with Django.	A very good level of ability to create a clean code structure, including HTML templates, Bootstrap and JavaScript	Demonstrates a good evidence of a hosted Django app that is fully functional and accessible
<b>Pass Criteria</b>				
Adequate use of the key concepts of Django.	Adequate use of the key concepts of Database integration in Django.	Makes adequate use of integrating Authentication and Authorization with Django.	Adequate evidence for creating a clean code structure, including HTML templates, Bootstrap and JavaScript	Adequate evidence of a hosted Django app that is fully functional and accessible
<b>Unsatisfactory Criteria</b>				
Response is partial or tangential. Requires greater depth, level of detail and discussion.	Response is partial or tangential. Requires greater depth, level of detail and discussion.	Inadequate application of course learning integrating Authentication and Authorization with Django.	Inadequate use for creating a clean code structure, including HTML templates, Bootstrap and JavaScript	Unsatisfactory evidence of a hosted Django app that is fully functional and accessible
<b>Clear Fail Criteria</b>				

Little evidence of the concepts of Django.	Little evidence of Database integration in Django.	Little evidence of ability to apply key course concepts in practice.	Very little evidence of for creating a clean code structure, including HTML templates, Bootstrap and JavaScript	Little evidence of a hosted Django app that is fully functional and accessible
No Attempt Criteria				
No submission	No submission	No submission	No submission	No submission