

Task-by-Task Guide

If you'd like a little more support while completing this project, explore this step-by-step resource to get additional hints and resources to help you along each task of this project.

Step 1: Setting Up Your Project

In this step, you will create a new GitHub repository that you will be working on for this project. This is important because you will be showcasing your work to future employers using this GitHub repository. After you create your repository, you will clone it and create a local branch to start working on your project.

You can refer to the resources section below for more information on how to create and clone GitHub repositories.

It's also important for you to set up your local development environment by installing the required tools. You can refer to the Setup section of your portfolio project for more information about configuring your local development environment.

After you set up your environment, it's time for you to create your brand new Django app and start building your website! Do not forget to commit your code regularly to your local branch as well as your GitHub repository to prevent code loss. For more guidance on committing and pushing your changes, you may refer to the Git resources below.

Step 2: Scaffolding the Admin Interface and Importing Data

In this step, you will create your Django admin interface. The purpose of this step is to create a dashboard that users will be using to update the book information data. This dashboard will contain the views and the links that users need to create, read, update, and delete book information. You may refer to the resources section below for more guidance on how to implement the Admin interface in Django.

Once you've created your dashboard shell, it's time for you to create your data models. To do that, you will first go back to the company's [spreadsheet](#) to extract business entities and determine their relationships. After you extract these entities and their fields, you can start adding data model classes to your application and configure the relationships between these models.

Step 3: Migrating Data from Spreadsheets to Django

After building the data layer of your app, it's time for you to import data from the company's [spreadsheet](#) into your app's database. To do that, you will prepare your data for import. This might involve a variety of tasks depending on the way you have implemented your data models and the way you want to use to import your data into the system. These tasks could include splitting the data or exporting it to a different format before uploading it to the database. You may refer to the

resources below for some tips on how to import and export data.

After completing and testing the admin interface and uploading your data, it's time for you to move on to implementing your report view.

Step 4: Designing Your Report

To start working on your report, you will create a report view and add it to your website. The report view should first retrieve the data from the database and format it in a way that is suitable for display. This step might involve techniques such as data grouping, sorting, filtering, and others. The techniques you use could vary depending on the way you're planning to implement your reports. You may refer to the below resources on how to prepare your data for reports.

Once your data is ready for display, you can display your data on the webpage using the chart.js library. To do that, you will need to reference the library and its dependencies in your view. You can also add additional buttons and link to your reports view to sort, filter, or group data. For some guidance on using charts.js, you may refer to the resources below.

Step 5: Publishing Your Project

When you complete and test your work, make sure to commit any changes left to the local branch, then push those changes to your remote branch on GitHub. After uploading your changes, you will merge the branch you have been working on with the main branch on your GitHub repository. Additionally, you can deploy your project to the web to allow future employers to access your work easily.

Resources:

[Clone a repository in GitHub](#)

[Git basic branching & Merging.](#)

[Git Commit Documentation.](#)

[Django Admin](#)

[Build Data Models with Django.](#)

[Git Commit Documentation](#)

[Import data from spreadsheets into Django](#)

[Chart.js with Django](#)

[Git Commit Documentation.](#)

[Git Push Documentation.](#)

[Github Pull Requests.](#)