

Capstone Project Documentation

Members: Co Lianne, Go Yasmeen, Reyes Ravi

What is your project and what does it aim to do?

Our to-do list application “GEDONE”, which is a combination of the words “get” and “done”, aims to improve productivity by keeping track of any types of tasks and reminding the user of their required completion dates. Our project helps a user create, organize, and prioritize responsibilities efficiently through listing, which helps objectives be presented in a way that is manageable. This allows one to remain calm and focused. Lists, items, and due dates can all be created in the project, making planning stress-free!

Who is the project for and why?

GEDONE is geared towards the general public, as its user-friendly interface allows anyone to create a personal to-do list. For instance, students can list their assignments and projects, while parents can write out their errands. But we believe our application is especially important for corporate settings, as it can assist groups and individuals in achieving a set number of objectives in a given time period, increasing business productivity. The project enables users to easily compartmentalize targets and goals within a broader plan, making the application useful for businesses with different departments and functions. It also allows for straightforward collaboration between team members, as tasks can be delegated amongst the group. Moreover, there is less chance for internal stakeholders to misuse valuable time trying to find information and instructions if they create to-do lists and items with our project, since they are able to follow an outlined schedule and agenda. In line with this, the application can help coordinate business resources to ensure that plans are carried out smoothly.

Process-flow or expected user journey (what does your user do given your project? What is the expected input and output of your project? In what formats?)

1. Login or Register

Individual accounts allow for a personalized experience per user. Furthermore, it makes GEDONE more practical in group settings such as the classroom or workplace.

a. LOGIN

- i. Users can enter their username and password to access and edit their to-do lists.

b. REGISTER

- i. New users have the option to make a new account.

2. GEDONE Interactive Homepage

After logging in, users are directed to the project's homepage where they can interact with their to-do lists. They can complete one or more of the following:

- a. View all lists
 - i. All created lists will be shown on the base home page.
 - ii. Each list is a button that when clicked, brings the user to the respective list page containing items / tasks.
- b. View all items in each list
 - i. All current (and completed) items / tasks will be shown on the list page.
 - ii. Each item is a button that when clicked, shows the user its title, description, and due date.
- c. Edit list and items
 - i. When clicking the item or title button, users can edit any of its contents
- d. Create a new list or item
 - i. Asks user to input a list title
 - ii. Asks user to input an item title, description, and due date
 - iii. Each object has its own 'save' and 'cancel' buttons
- e. Mark task as finished
 - i. Tick the completion box at the bottom of the item description
- f. Delete lists or items
 - i. Each object has its own 'delete' button
 - ii. Clicking the 'delete' button brings user to a confirmation page, for additional verification that they are certain about deleting the object
- g. Search for a task
 - i. Asks user to input text in a search bar

3. Logout

In the case of multiple users sharing one device, there is the option to log in and out of the account between sessions. If a user does not log out at the end of their session, they will remain logged in when re-opening GEDONE.

How does your code work? How were the files set up?

All files are in 1 project folder root, which is the set up for a Django project. A virtual environment is also activated to secure the isolation of the application code. We employed the Django object-relational mapping tool to create our data model and generate the databases that store any added list and their respective items.

The code works using a one-to-many relationship, where multiple web pages—such as to-do list and item forms—have been created through Django templates and serve to extend the base HTML page. Each HTML page is styled using CSS, which customizes the text, buttons, fonts, colors, background, and size of elements according to our code. These are all connected through the Django URL dispatcher, which sends data between pages and files, allowing the to-do list application to smoothly transition between the main page, list editing form, and item editing form.

Python view functions ensure that the application objects, which are lists and items, are returned according to user input (eg. through clicking buttons). These include viewing list contents and item descriptions, creating new lists and items, updating lists and items, and deleting lists and items.

The application can be run using `python manage.py runserver` in the command line, which directs the user to a webpage from the local host. Additionally, the application can be run with an admin extension, such that a user can access the admin interface which requires login credentials for you to see all the authorized groups and users, as well as all the items and lists.