

# Product Backlog Todont

Priority: low, medium, high

## Core Site

### ID: CS001 - Look and Feel

#### Feature/User Story:

As a user, I want to use an aesthetically pleasing website that is both satisfying and easy to interact with.

#### Acceptance Criteria:

1. The site should be pleasing to look at.
2. The site should be easy to interact with while, at the same time, suit the user's level of customization.
3. The site should contain the default pages of any website (Landing Page, About Us, Contact Us, Register and Log In).

#### Tasks:

- Create a wireframe in Figma

### ID: CS002 - Creation of the site

#### Feature/User Story:

Initializing the project and tools in order to begin developing the website.

#### Acceptance Criteria:

1. The site should be able to work on the localhost using angular and Laravel.
2. The project should be saved in a version control system on the internet to increase the collaboration between the group and have back-ups in case something goes wrong.

#### Tasks:

- Create a Laravel and an Angular project in the same workspace
- Initialize the git repository in the workspace

## Core Features

### ID: CF001 - Try the app

#### Feature/User Story:

As a user, I want to try out the app before creating an account.

#### Acceptance Criteria:

1. Guests can use the app just like the user but they won't be able to save the changes unless they are logged in.
2. If the guest chooses to register, the work gets saved and sent to that user (if the user decides that he wants the changes saved).

#### Tasks:

- Use one of the storages of the browser to store the changes being made by the guest.
- Save the changes made by the guest and save them into the user that was created.

### ID: CF002 - User Authentication

#### Feature/User Story:

As a user, I want to register and log in, so I can access my personal lists and tasks.

#### Acceptance Criteria:

3. Users can register with an email and password.
4. Users can log in securely.
5. Logged-in users see only their lists and tasks.
6. Users can recover their password using their email.
7. Users can check the "remember me" checkbox so that they won't have to input their password repeatedly.

#### Tasks:

- Create user registration, login and forgot password forms in Angular.
- Implement secure login with hashed passwords. (Optional)
- Create user session management for logged-in users.
- Add "Forgot Password" functionality with email recovery.
- Create a "Remember Me" feature using cookies or tokens.

### **ID: CF003 - Switch Between To-Do and To-Don't lists**

#### **Feature/User Story:**

As a user, I want to be able to switch between the to-do and to-dont workspace, so that I don't get confused about which list belongs to which.

#### **Acceptance Criteria:**

1. Users can click a button to change their workspace from to-do or to-don't.
2. The workspace only shows the lists that belong to either the to-do or to-dont workspace.
3. The workspace is set to show todont lists by default

#### **Tasks:**

- Add a toggle button or tab for switching workspaces.
- Filter tasks dynamically based on the selected workspace.
- Test workspace switching for seamless interaction.

### **ID: CF004 - Show and Hide Lists**

#### **Feature/User Story:**

As a user, I want to be able to show or hide my lists to not flood the workspace and make it look more clean.

#### **Acceptance Criteria:**

1. Users can use the checkboxes to show or hide the lists.

#### **Tasks:**

- Make a component with a title and a checkbox that appears on the sidebar.
- Show or hide the list depending on the state of the checkbox

### **ID: CF005 - Add lists**

#### **Feature/User Story:**

As a user, I want to add new lists, so I can organize what I need to do.

#### **Acceptance Criteria:**

1. Users can input a list title.
2. The list is saved in the database and displayed in the workspace.

3. The list has an add task button

**Tasks:**

- Make a button to add new list
- Lists are stored in the database.

**ID: CF006 - Edit lists**

**Feature/User Story:**

As a user, I want to edit a list, so that I can order the tasks, rename the list, and or delete it.

**Acceptance Criteria:**

1. Users click a button on the desired list to display a menu with the options to order task, rename the list, or delete it entirely

**Tasks:**

- Add an "Edit" button for each list.
- Pre-fill an edit form with the current list details.
- Save changes to the database and update the list.

**ID: CF007 - Delete lists and the tasks belonging to the lists**

**Feature/User Story:**

As a user, I want to be able to delete lists and all its tasks.

**Acceptance Criteria:**

1. User clicks a button on the desired list and gets notified if they are sure that they want to delete the list and all the tasks within it.

**Tasks:**

- Add a "Delete" button for each list.
- Make a pop to ask the user if they are sure that they want to delete the list
- Save changes to the database and update the workspace.

**ID: CF008 - View Tasks**

**Feature/User Story:**

As a user, I want to view a list of my tasks, so I can track my progress.

**Acceptance Criteria:**

1. Users are able to see all pending tasks in a list, as well as the completed task from a dropdown element.
2. Tasks are sorted by due date by default.

**Tasks:**

- Display all pending tasks in a visually appealing list.
- Create a dropdown titled “Completed (num\_completed)” that shows completed tasks.
- Sort tasks by due date using a default sorting algorithm.

**ID: CF009 - Add a Task****Feature/User Story:**

As a user, I want to add a new task, so I can organize what I need to do.

**Acceptance Criteria:**

1. Users can input a task title.
2. For TODO tasks, users can optionally add a description, due date, tags, and if they want the tasks to repeat
3. For TODONT tasks, users can optionally add a description. The streak of the todont task is set automatically to 0.
4. The task is saved in the database and displayed in the task list.

**Tasks:**

- Create a task input form that changes depending on which category the workspace is set to (todo or todont).
- The TODO form has the title, description, due date, tags, and repeat on that can be edited.
- The TODONT form has the title, description that can be edited.
- Validate task title input.
- Save task data to the database.
- Update the task list dynamically after adding a task.

**ID: CF010 - Edit a Task****Feature/User Story:**

As a user, I want to edit a task, so I can update the details if needed.

**Acceptance Criteria:**

2. Users can edit the title, description, and due date of a task.
3. For TODO tasks, users can edit the description, due date, tags, and if they want the tasks to repeat
4. For TODONT tasks, users can edit the description. The streak of the todont task can't be changed.

**Tasks:**

- Add an "Edit" button for each task.
- Pre-fill an edit form with the current task details.
- Save changes to the database and update the task list.

**ID: CF011 - Mark a Task as Completed****Feature/User Story:**

As a user, I want to mark tasks as completed, so I can track my progress.

**Acceptance Criteria:**

1. A "Complete" button is available for each task.
2. Completed tasks move to a separate "Completed Tasks" section.
3. For TODONT tasks, when a task is completed, the task's streak goes back to zero.

**Tasks:**

- Add a "Complete" check box next to each task.
- Update task status in the database upon completion.
- Move completed tasks to a new section in the list.

**ID: CF012 - Delete a Task****Feature/User Story:**

As a user, I want to delete tasks, so I can remove tasks that are no longer needed.

**Acceptance Criteria:**

1. Users can delete tasks by clicking a "Delete" button.
2. A confirmation dialog appears before deletion.

**Tasks:**

- Add a "Delete" button for each task.
- Implement a modal confirmation dialog.
- Remove the task from the database and UI after confirmation.

## Advanced Features

### ID: AF001 - Order/Search Tasks

#### Feature/User Story:

As a user, I want to order or search my tasks, so that I can organize or find them more quickly.

#### Acceptance Criteria:

1. Users can search tasks by keywords.
2. For TODO tasks, users can sort tasks by their tags, due date or their own way.
3. For TODONT tasks, users can order tasks by their streaks.

#### Tasks:

- Add a search bar with keyword matching functionality.
- For TODO tasks, create order options for tags, due date, and custom order.
- Dynamically update the task list based on search and order criteria.

### ID: AF002 - Set Notifications

#### Feature/User Story:

As a user, I want to receive notifications for upcoming tasks, so I don't miss deadlines.

#### Acceptance Criteria:

1. Notifications are sent based on the due date, but also a daily reminder of the option can be toggled to remind you to check your todo and todont tasks.
2. Users can enable or disable notifications for due dates.

#### Tasks:

- Add a notification toggle to user settings.
- Create a system to check for upcoming deadlines.

- Send notifications via email, push, or in-app alerts.

#### **ID: AF003 - Drag and Drop for Task Reordering**

##### **Feature/User Story:**

As a user, I want to reorder tasks with drag and drop, so I can prioritize visually.

##### **Acceptance Criteria:**

1. Users can drag tasks to reorder them.
2. The new order is saved automatically.

##### **Tasks:**

- Implement drag-and-drop functionality using a library or custom code.
- Save the updated task order in the database.
- Ensure smooth animations and transitions during dragging.

## **UI/UX Enhancements**

#### **ID: UX002 - Responsive Design**

##### **Feature/User Story:**

As a user, I want the app to work on both mobile and desktop, so I can access it anywhere.

##### **Acceptance Criteria:**

1. The app is fully responsive and adjusts to different screen sizes.

##### **Tasks:**

- Use CSS media queries to create a responsive layout.
- Test responsiveness on multiple devices and screen sizes.
- Adjust touch interactions for mobile users.

## **Settings Menu**

#### **ID: SM001 - Dark Mode**

##### **Feature/User Story:**

As a user, I want a dark mode option, so I can use the app comfortably at night.

##### **Acceptance Criteria:**



1. Users can toggle between light and dark modes.

**Tasks:**

- Implement a dark mode theme for the app.
- Add a toggle button in the user settings or header.
- Save user preference in local storage or database.

**ID: SM002 - Key bind customization**

**Feature/User Story:**

As a user, I want to have my personal key binds to be able to facilitate navigation through my workspace.

**Acceptance Criteria:**

1. Users can edit their key binds.
2. The new settings are saved automatically

**Tasks:**

- Implement drag-and-drop functionality using a library or custom code.
- Save the updated task order in the database.
- Ensure smooth animations and transitions during dragging.

## Cloud Service

**ID: CS001 - Import the project into AWS**

**Feature/User Story:**

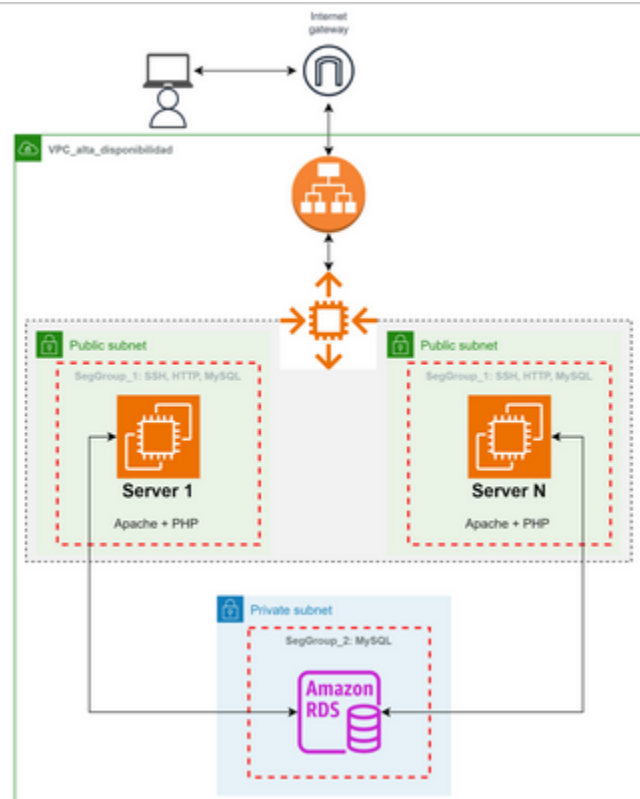
As a user, I want to access the site through the domain instead of installing it locally.

**Acceptance Criteria:**

1. Users can access the site via a domain.
2. All the components of the site should be in an EC2 (except the database, which should be stored in the RDS)

**Tasks:**

- Create the following diagram:



- Use Route 53 to create the domain.

## Sprints:

**Sprint1:** Planning, Initialization of the project, tools, and Git

**Sprint2:** Core site pages, core features for the app, and user authorization

**Sprint3:** Project migration to cloud (AWS), adding unfinished features.

**Sprint4:** Changing the AWS structure to be like the image above. Fixing errors and completing the documentation + presentation.