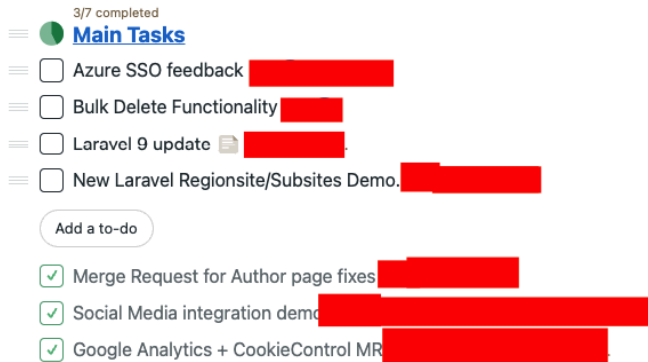




Project Title: TaskHive – A Team-Based Task Management System



User Functionality



Registration & Login (AJAX-based)

- **Fields during registration:**

- Name *
- Email (Email have all kind of Validations) *
- Password & Confirm Password (Both Have Validation like strong password, Match Password) *
- Country (Country Dropdown , Take 3 Country Fix) *
- Languages Known (Multi-select: Hindi, Gujarati, English) (Checkbox) *

Note : Client Side Validation on Keyup, keychange and blur

- **Registration Flow:**

- Registration is submitted via AJAX.
- On successful registration, the user is redirected to the login page.

- **Login Flow:**

- AJAX-based login using email and password.
- On successful login, users are redirected to their dashboard.

User Dashboard (Add and list in single form)

- **Add Task Form (AJAX submission): (Add To Do button Like thats see above Screenshot) Model Form.**

- Fields:
 - Task Title *
 - Completion Date *
 - Description *
 - File Upload *
 - Assign To: One of two predefined Super Admins (shown in a fixed dropdown) *
- Default task status: Pending
- A user can add **multiple tasks** (One-to-Many relationship between User and Task)

Note : Client Side Validation on Keyup, keychange and blur

- **Task List View (User Side):**

- Display all tasks created by the user.
- Each task includes:
 - Checkbox to mark completion
 - Task Name
 - Assigned Super Admin's Name
 -
- **Completion Flow:**
 - Tasks are displayed with an unchecked checkbox (status: Pending)
 - When the user checks the checkbox, an AJAX request marks the task as Completed

Admin Functionality

Login

- Uses the **same login form** as users (role-based redirect after login)

Admin Dashboard

- Displays a list of all **completed tasks (Using Ajax)**
 - Each record includes:
 - Task Name
 - Completion Status
 - User Name (who completed the task)
 - Optionally, filter tasks by user or date
-

Technical Features (Note)

- **Django**
- **AJAX** for asynchronous registration, login, task submission, and status update
- **Django ORM** for relational mappings
- **Custom Admin Panel** (optional, for managing tasks and users)
- **Two predefined Super Admins** are seeded/fixed in the database
- **One-to-Many Relationship** between User and Task models
- **File handling** via FileField in Django model
- Each Files Must be Required in Each Form.
- All Form Field is Required.
- Make a Video Of Working Frontend. With explanation . How it worked.