

Full Stack Backend Developer Python Task

Section 1: Python and Web Frameworks (Django/Flask)

Task 1: Simple Web Application

- **Description:** Build a basic web application that allows users to register and manage a list of tasks.
- **Requirements:**
 - Use Django or Flask to create the application.
 - Implement user registration and login.
 - Create CRUD (Create, Read, Update, Delete) operations for tasks.
 - Use an in-memory database (e.g., SQLite in-memory mode).

Section 2: Frontend Development

Task 2: Basic Frontend Implementation

- **Description:** Enhance the above application with a basic frontend.
- **Requirements:**
 - Use HTML5, CSS3, JavaScript (ES6+) and template engines like Jinja for the front end.
 - Implement a simple interface. Design it according to your creativity.
 - Ensure the frontend communicates with the backend API for CRUD operations.

Section 3: API and Web Services

Task 3: Basic RESTful API Development

- **Description:** Create basic RESTful APIs for the application.
- **Requirements:**

- Develop endpoints for user registration, login, and task management.
- Ensure the API follows REST principles.
- Implement basic error handling and response codes.

Bonus Task

Completing the following bonus tasks can increase your chance of qualifying for the next round.

- A user can assign tasks to other registered users.
- Ensure only the task creator or assigned users can update the task.
- Only the task owner can delete the task.
- Show the last updated status: Who and when was the task last updated?

Submission Instructions

- Provide a public GitHub repository link with instructions on how to set up and run the application.
- Commit frequently with proper commit messages.
- Include a README file explaining the project structure and any assumptions made.
- Ensure all necessary dependencies are listed in a requirements.txt file.
- Host the web app in any service you choose and share the link after completing the task.