**JAVA CODING CHALLENGE-2**

Create APIs and tables to manage tasks and their statuses with additional functionalities like assigning tasks to users, setting due dates, and tracking progress. Here are the API requirements and business logic: (Please use Java 8 or above)

Create Task:

1. Endpoint: POST /api/tasks

* Request Body: Task object (title, description, dueDate)
* Description: Create a new task with the provided title, description, and due date.
* Update Task:

1. Endpoint: PUT /api/tasks/{taskId}

* Request Body: Task object (title, description, dueDate, status)
* Description: Update an existing task with the provided details, including the status. If the status is changed to "Completed," set the completed date.

Delete Task:

1. Endpoint: DELETE /api/tasks/{taskId}

* Description: Delete a task.

Get All Tasks:

1. Endpoint: GET /api/tasks

* Description: Get a list of all tasks.

Assign Task:

1. Endpoint: POST /api/tasks/{taskId}/assign

* Request Body: User ID
* Description: Assign a task to a specific user.

Get User's Assigned Tasks:

1. Endpoint: GET /api/users/{userId}/tasks

* Description: Get a list of tasks assigned to a specific user.

Set Task Progress:

1. Endpoint: PUT /api/tasks/{taskId}/progress

* Request Body: Progress percentage (0-100)
* Description: Set the progress percentage of a task.

Get Overdue Tasks:

1. Endpoint: GET /api/tasks/overdue

* Description: Get a list of tasks that are overdue based on the current date and the task's due date.

Get Tasks by Status:

1. Endpoint: GET /api/tasks/status/{status}

* Description: Get a list of tasks with a specific status.

Get Completed Tasks by Date Range:

1. Endpoint: GET /api/tasks/completed

* Parameters: startDate, endDate
* Description: Get a list of completed tasks within the specified date range.

Get Tasks Statistics:

1. Endpoint: GET /api/tasks/statistics

* Description: Get statistics on the total number of tasks, the number of completed tasks, and the percentage of completed tasks.

Priority-based Task Queue:

* Implement a priority queue for tasks based on their due dates and priority levels. The priority levels could be "High," "Medium," and "Low."

Design the necessary database tables to store the Task and User entities, along with any additional tables required to support the functionalities mentioned above. You can choose your preferred database and define appropriate data types and lengths for the properties.

Please create the Spring Boot application with the above APIs and database tables according to the provided requirements. Handle error responses and validations appropriately, and implement the necessary business logic for task management and assignment.