TASK MANAGEMENT SYSTEM

1. Create a Postgre database with the following tables. Define field types based on your understanding.

Category

- a. Id (PK)
- b. Name

Task

- a. Id (PK)
- b. TaskName
- c. Assignee
- d. Categoryld (It should be set as a foreign key to **Category** table)
- e. Description
- f. DueDate
- g. Category
- h. City
- 2. Create a.NET core MVC project using repository pattern
- 3. Use Entity framework with PostgreSQL with Data first approach.
- 4. Create a screen to show the list of records as shown in first screenshot
- 5. Create Add/Edit form as shown in 2nd image
 - Add/Edit methods must be separate. Use partial view for Add and Edit forms.
 - If you are not going for nice to have features 1 & 2, use Dropdown control to populate existing **Category** and use its Id to save the **Task** record.
 - On click of Delete, record should get deleted from the database and record should get disappeared from the list on UI.

Nice To Have:

- 1. **Category** field must be auto-suggest control. If user selects existing **Category**, then it should use its Id in the new/existing **Task** record while saving to database.
- 2. If user enters name that doesn't exist in the **Category** table, then on click of Save button, it should create a new **Category** record in the **Category** table and its Id should be used in the new/existing **Task** record in Add/edit form
- 3. Paging / Pagination
- 4. Use Validations on front-end

Note: Finish the task within the specified timeframe of 5 hours, from **1:30 PM to 6:30 PM**, and establish a new folder name **Assignment** within the same project repository in

GitHub. Ensure that the assignment is **committed before 7:00 PM**; Any changes made after **7:00 PM will not be accepted and will result in the negative marking**. Share the GitHub link with the coordinator once the assignment is completed within the designated time.



