## LIBRARY MANAGEMENT SYSTEM

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

## Borrower

- a. Id (PK)
- b. City

## Book

- a. Id (PK)
- b. BookName
- c. Author
- d. Borrowerld (It should be set as a foreign key to **Borrower** table)
- e. BorrowerName
- f. DateOfIssue
- g. City
- h. Genere
- 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 **Borrower** and use its Id to save the **Book** 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. **City** field must be auto-suggest control. If user selects existing **Borrower**, then it should use its Id in the new/existing **Book** record while saving to database.
- 2. If user enters name that doesn't exist in the **Borrower** table, then on click of Save button, it should create a new **Borrower** record in the **Borrower** table and its Id should be used in the new/existing **Book** 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.





