**Documentation of OKR:**

Index:

1. Added Add Project button to add new projects,
2. Created Project create modal,
3. Sending Project data (Name, description, assignee, department) to API to save project,
4. Populated users in the Assign People dropdown to assign them to projects,
5. Designed Project-Cards in the Index page,
6. Enabled scrolling of layout.

Details:

1. Created getCookie() function to get ComId and UserId,
2. Created a container for Project Assignees,
3. Populating assignees in Project Assignees field from saved assignees + Showing all company users to edit the project assignees,
4. Created MyTemlateDTO in Core,
5. Created populatePAssignees() function to populate the list,
6. Created PsaveAssignee() function to save project assignees,
7. Created countTasks() function and other properties to show Project name, Total Estimated Time, Due Date, Completed Tasks, Delayed Tasks, Completed Tasks (No Delay) fields,
8. Created getTabData() function to show Milestone, Task and Subtask name, assignees, Due Date, Est. Time, Est. Date, Delay, Progress and Action fields,
9. Created assigneeFormatter, progressBarFormatter and action field custom formatter for customizations on Tabulator fields,
10. Modified MilestoneListViewDTO (EstDate, SEst, EstTime1, Delay, CompleteDate, Progress) to handle the getTabData() function’s GetIncludeT() method in API,
11. Added EstTime1 and CompleteDate fields in Core.Models,
12. Modified GetAllInlcudeT (CompleteDate, Status, DueDate, Type, EstTime1) method in DataAccess.Repositories,
13. Added Add Milestone Button and Off-canvas with milestone name, description, Assignees, Add Task fields,
14. Created a container for Milestone assignees that shows assignees who are assigned to the project,
15. Created populateMaddAssignees() function to populate the assignees in the Milestone assignees container,
16. Created addTaskCard() function to add new Tasks when Add Task button in the Off-canvas is pressed and Delete button to delete the task-card,
17. Created a container to show Task assignees for every individual tasks, the task assignee list is based on Milestone assignees,
18. Created addSubtask() function to add new Subtasks under the parent task when Add Subtask button is pressed and generate fields for subtask name, Due Date, assignees, Estimated time and Delete button to delete the subtask-card,
19. Created MytaskDTO in DataAccess,
20. Created populateTaskaddAssignees() function to populate the task assignee list from the milestone assignees,
21. Created a container to show Subtask assignees for every individual subtask, the subtask assignee list is based on its parent Task assignees,
22. Created toggleDatePicker() function to get the date field for Due Date,
23. Created MySubtaskDTO in DataAccess,
24. Created populateSubtaskAssignees() function to populate the subtask assignee list from parent task assignees,
25. Created initializeSelect2() function for every Select2 field in Task and Subtask card,
26. Created createTemplate() function to save the milestone along with all its tasks and subtasks,
27. Modified SubTaskDTOs() and CreateTemplateT() function to save subtask with assignees,
28. Created an off-canvas with milestone name, assignees, description field to edit milestone from action column,
29. Separate assignee field in edit milestone that shows existing assignees + project assignees to edit it,
30. Created TemplateAssignees() method in DataAccess.Repositories to get the milestone’s data with saved assignees,
31. Created getData() function to load the edit off-canvas depending on its type,
32. Created UpdateMilestone() function to send the updated milestone data to the API,
33. Modified UpdateTemplate() method in Application layer to handle update,
34. Created UpdateWithAssignee() method in DataAccess to handle update milestone,
35. Created an off-canvas with task name, assignees, description field to edit task from action column,
36. Separate assignee field in edit task that shows existing assignees + milestone assignees to edit it,
37. Created TemplateAssignees() method in DataAccess.Repositories to get the tasks’s data with saved assignees,
38. Created TaskAssignees() method to get the template id and tasks information with assignee,
39. Created GetTemplateAssignees() API to get the existing assignees in the milestone,
40. Created UpdateTask() function to send updated data to the API,
41. Modified UpdateTask() method in Application to handle edit,
42. Created UpdateWithAssignee() method in DataAccess to update the task,
43. Created an off-canvas with subtask name, assignees, description, due date, estimated time field to edit task from action column,
44. Separate assignee field in edit task that shows existing assignees + task assignees to edit it,
45. Modified TaskAssignees() method to get the task id and tasks information with assignee,
46. Created GetWithAssignees () method in DataAccess.Repositories to get the subtask’s data with saved assignees,
47. Created UpdateSubtask () function to send updated data to the API,
48. Modified UpdateSubTask () method in Application to handle edit,
49. Created UpdateWithAssignee () method in DataAccess to update the subtask,

MyTask: