Lab Project: Database Systems

DUE DATE: 11:59 PM 11TH, MAY 2023

GOAL

Enable the student to design and develop a database. In this project, the student is required to show their abilities of

- Analyzing the System Requirement
- Represent the requirement into logical design using Entity Relationship (ER) or Enhanced Entity Relationship (EER) model
- Mapping the designed model into relational schemas
- Transform the relational schemas into normalized tables
- Writing SQL statements (DDL) to create the tables including all applied integrity constraints
- Writing SQL statements (DML) to populate the initial records of each table
- Front End Development of Forms/ Reports using NodeJS & MYSQL

MUST READ INSTRUCTIONS BEFORE STARTING

- It is a group project with a maximum of three and a minimum of one student.
- CROSS-SECTIONS NOT ALLOWED
- Zero marks will be awarded to the students involved in plagiarism and late submission.
- Only one GROUP MEMBER will be responsible for all submissions on Google Classroom.
- Your submission must include all .html and .css files in ZIP format, as well as a Word document with your report and .sql files. All of these files (zip, report, and .sql) should be contained within a ZIP file.
- The name of the ZIP File should be named after the roll numbers of all group members. i.e., Name1-RollNO1_ Name2-RollNO2_Name3-RollNO2.zip. For example, contains a distinct zip folder for the PWB_front-end.zip, ProjectReport_Section.doc, and DB_Section.sql.
- The naming convention must be followed strictly.

PROJECTWORKBASE(PWB) APPLICATION

INTRODUCTION

PWB is a software tool used for project management, issue tracking, and software development. It is particularly popular among agile development teams as it allows for the creation and tracking of user stories, sprints, and other agile development methodologies. PWB also provides features for collaboration, reporting, and customization, making it a versatile tool for teams of all sizes and types. Here are some of its key features:

- Project management: It allows users to create and manage projects, including setting up projects, assigning tasks, and tracking progress.
- Issue tracking: It allows users to create and track issues or bugs, including assigning them to specific team members and adding comments.
- Customization: It can be customized to fit the specific needs of a team, including custom workflows, fields, and screens.
- Reporting: It provides a variety of reporting options, including agile boards, burndown charts, and custom reports.

PWB is a powerful and flexible tool that can help teams of all sizes and types to manage projects, track issues, and develop software. Project management is one of the core features of PWB. With PWB, you can create and manage projects with ease. project management capabilities allow you to easily create and manage projects, visualize your workflow, assign tasks, and track progress. It is a powerful tool that can help your team stay organized, collaborate more effectively, and deliver projects on time. Here are some of the key features of PWB’s project management capabilities:

1. Project setup: You can create new projects and configure them to fit your team's specific needs. You can choose from a variety of project templates or create a custom template. You can also set project permissions and configure notifications.
2. Task management: You can create tasks and assign them to team members. You can also set task priorities, and due dates, and add comments or attachments.
3. Roadmaps: It provides a roadmap feature that allows you to plan and visualize your team's work over time. You can create a roadmap that includes multiple projects and set goals or milestones.

CASE STUDY

User information has to be stored in the database in order to use PWB. There are various categories of users, each with its own set of tasks and responsibilities. Administrators have complete authority over the system and are responsible for setting it to meet the organization's requirements. The admin is responsible for managing user accounts including adding and removing users, assigning roles and permissions, setting up user groups, and configuring access controls and permissions for users and groups. creating custom reports and dashboards, and setting up integrations with other tools and systems.

A project has only one team lead/Project manager who is responsible for managing individual projects. They create and manage projects, assign issues to team members, and use reports and dashboards to track project progress. They can also examine the details and current status of their team members. Developers/ Team Members are responsible for working on the issues that have been assigned to them and updating the status of those issues as they progress. They can potentially cause new problems. Project managers create and assign tasks to team members, set task priorities, due dates, and add comments or attachments. Team members can easily view their assigned tasks and track their progress only.

Customers are another category of users. They can be external stakeholders or internal business users with access to the APP who can report problems or request new features. They can create new issues as well as track the status of issues they have already reported. Each type of user has different levels of access and permissions in PWB, depending on their role and responsibilities. Administrators have the highest level of access and control, while customers have the lowest level of access. This allows teams to work together effectively and efficiently while ensuring that sensitive information is only accessible to those who need it.

Projects can have multiple versions, each representing a specific release or iteration of the software being developed. Each board is associated with a specific project and can be customized to show the issues that are most relevant to that project. In PWB, issues can be organized into projects, which represent the different software development projects being worked on. Sprints are usually associated with a certain project and contain a set of issues that are worked on over a fixed period of time. An "issue" is a task, bug, or feature request that must be tracked and addressed during the software development process. Within a project, issues can be created and managed. It can be assigned to individual team members as well as linked to other issues and project artifacts. Each issue is assigned to a workflow, which describes the various statuses that the issue can have, such as Open, In Progress, and Resolved. Every issue is given a unique identification and contains various details about the work that has to be done. including

1) The type of issue, such as a bug, task, feature request, or improvement

2) A brief summary or title of the issue. A more detailed description of the issue, including any relevant information or context.
3) The person or team responsible for working on the issue.
4) The current state of the issue, such as open, in progress, or resolved.
5) The level of priority assigned to the issue, such as critical, high, medium, or low.
6) The date by which the issue is expected to be completed.
7) Any comments or discussion related to the issue.
8) Any relevant files or documents attached to the issue.

By using PWB's project and issue management features, teams can collaborate more effectively and deliver high-quality software more efficiently.

A SAMPLE FRONT END

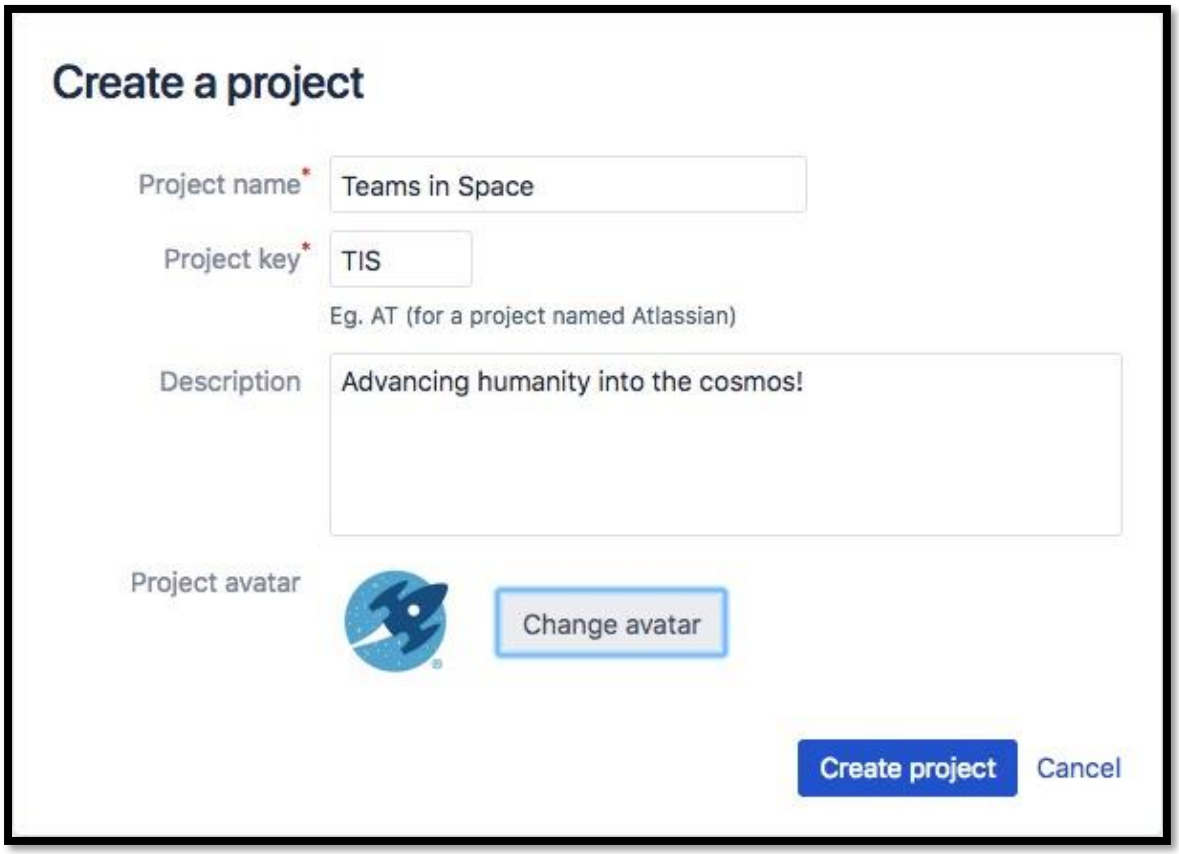
- You have to design the front end according to your creativity.
- Avoid adding static buttons and navigations.

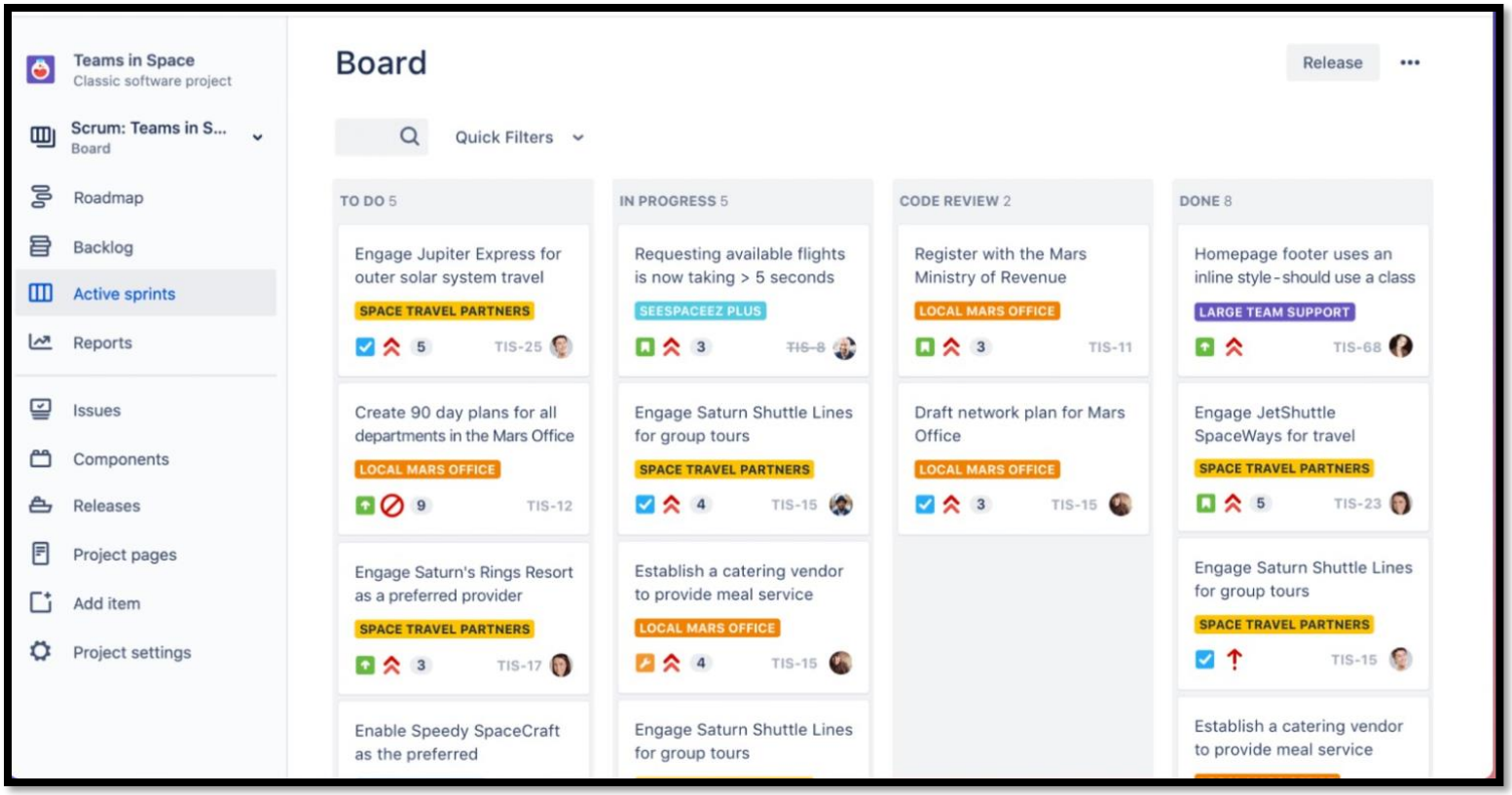
Your interface should be user-friendly and provide an efficient experience for managing projects, issues, and workflows. Here are some key features of the interface:

- **Navigation Bar:** The navigation bar is located at the top of the interface and provides quick access to various areas of the application, including projects, issues, boards, reports, and user management.
- **Projects:** The Projects section displays a list of all the projects that the user has access to. Each project displays its name, key, and a summary of its current status. Clicking on a project takes the user to the project overview page
- **Issues:** The Issues section displays a list of all the issues that the user has access to. Users can filter and search for issues based on a variety of criteria, including issue type, status, priority, and assignee.
- **Workflows:** The Workflows section allows users to define custom workflows for their projects, specifying the steps that an issue must go through before it is considered complete.
- **Boards:** The Boards section allows users to create and visual representation of the work that needs to be done, as well as its current status.

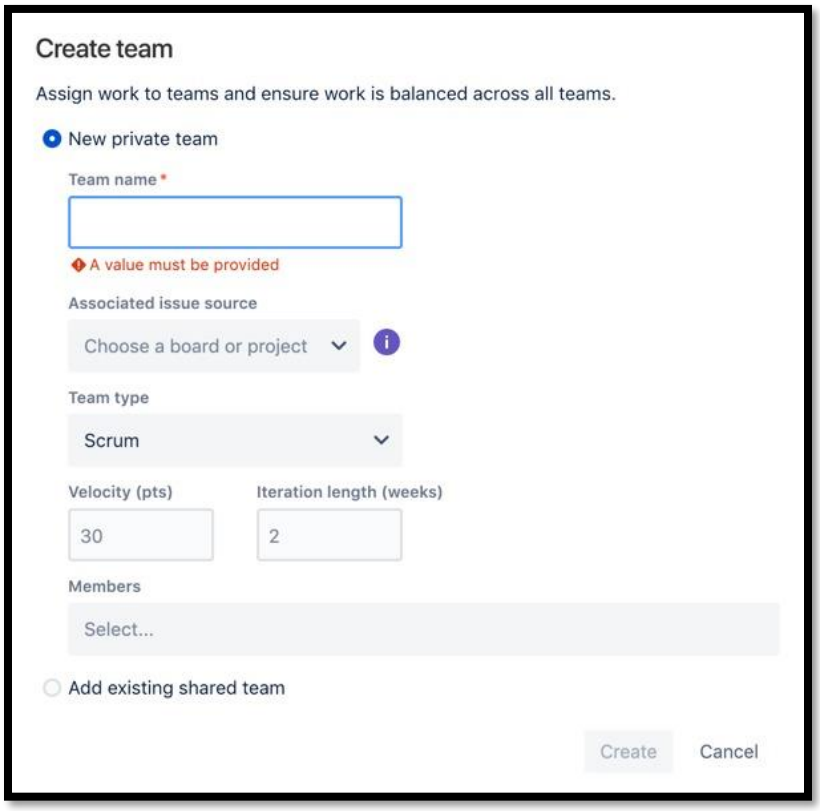
- **Reports:** The Reports section provides users with a variety of reports and metrics that allow them to track the progress of their projects and identify areas where improvements can be made.
- **User Management:** The User Management section allows administrators to manage user accounts, permissions, and access levels.

1. Start with the login page, every user type has a different portal. The login page typically consists of a simple form that prompts users to enter their username and password. Once users have entered their credentials, they are authenticated against the user database and granted access to the application if their credentials are valid. If the user enters incorrect credentials, they will be prompted to try again or reset their password.
2. Let's talk about the admin portal. For example, Admin can Add, or delete a project. Here is the sample interface for that feature in the admin portal.

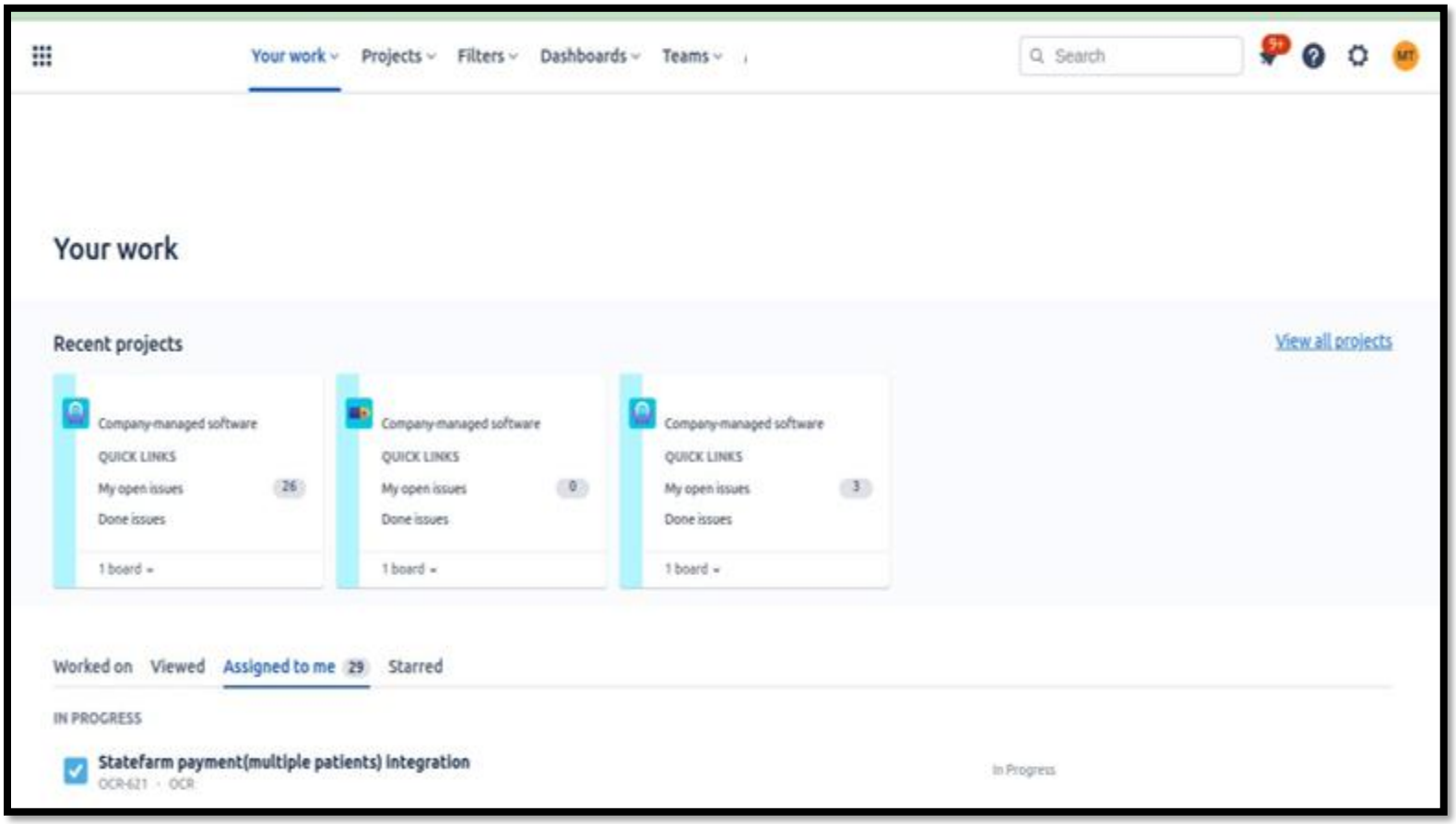




There are many sub-teams under a project. For example, an AI Team, a QA team, a management team, and so on. All the teams on a particular project work under one team lead. An employee or developer may work on multiple projects under different teams. While creating a team, available a drop-down list should be available for the admin to add that employee or worker under that project within a specific team.



3. Every team member can see only his/her project. Status of the current working projects, assign projects, previous projects, current team, and so on. Design the pages according to your understanding and creativity. Add the information of all the other roles that a team member could have after the above case study analysis.



DELIVERABLES

ERD

The entity-relationship (ER) data model allows us to describe the data involved in a real-world enterprise in terms of ENTITIES and their relationships and is widely used to develop an initial database design.

RELATIONAL SCHEMA

The relational schema allows us to map the aforementioned ERD to the database design.

DATABASE

You are required to design a database that meets the user's needs and implement this using MYSQL in .sql file(s).

WEBSITE

A fully functional app requires an interface for users to interact with. Create a web interface using HTML, and use PHP to connect this interface with your database.

- FORMS AND REPORTS on the Website

PROJECT REPORT(USING MS WORD)

Contain all flow of your project in detail and must contain:

- Contain your ERD
- Table description
- Schema diagram

EVALUATION METRIC

PROJECT FUNCTIONALITY	**DELIVERABLES**	**MARKS (100)**
SQL	ERD	15
	Table description	5
	Schema diagram	5
	Table creation and insertion in .sql file	15
	All necessary constraints on DDL	7
HTML INTERFACE	Login page	5
	The working navigation bar should contain all the necessary information (describe above)	5
	Separate portal for different users	7
	All other necessary pages	8
NODEJS FILES (CONTAIN DB CONNECTION AND WORKING QUERIES)	All necessary reports against each portal	10
	Buttons contain queries and display in a proper format	8

WORD REPORT	Contain all details description of your project	10
BONUS	Bootstrap, Triggers, Query optimization, visualization	10

A BRIEF OVERVIEW OF THE TECHNOLOGY

Front end: HTML (optional CSS, JavaScript, Bootstrap)

1. HTML: HTML is used to create and save web documents. E.g., Notepad/Notepad++.
2. CSS: Create an attractive Layout. (optional)
3. Javascript: Used for dynamic interactions with user (optional)

Back end: NodeJS, MYSQL