Project Title	Cricket Score Site
Technologies	MERN
Domain	Industry
Project Level	Easy

Table

Contents

**4.** **Submission requirements:**

1. Problem Statement:

The initiative involves streaming online real-time cricket match scores. Many people choose to observe the results online if they don't have access to television or cable at home or if they are travelling. The system features an administrator account where the administrator regularly updates the scores. Users can access their accounts at any time to view their scores. For customers to be informed in advance about the matches, the system also offers a timetable of all the games that will be played. As a result, users can monitor live cricket scores.

2. Features:

- Admin login
- Add-Update score details
- View Score Details (for Users)
- Manage Score Details

2.1. Report Generation:

- Can generate all type reports based on different categories i.e.- Batting, Bowling, Strike rate, wickets, etc.
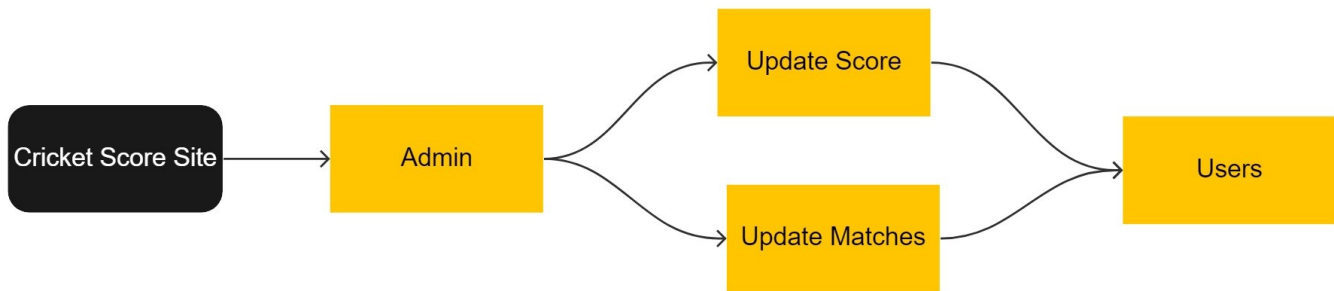
NOTE: Score should be updated in backend data base

3. Project Evaluation metrics:

3.1. Code:

- You are supposed to write code in a modular fashion ● Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works similarly in every environment (operating system).
- You have to maintain your code on GitHub.
- You must keep your GitHub repo public so anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

3.2. Database:
MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

3.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

3.4. Deployment:
Deploy the application on your preferred service.

3.5. Solutions Design:
You have to submit complete solution design strategies in High-level Document (HLD), Lowlevel Document (LLD), and Wireframe documents**.**

3.6. System Architecture:
You have to submit a system architecture design in your wireframe document and architecture document.
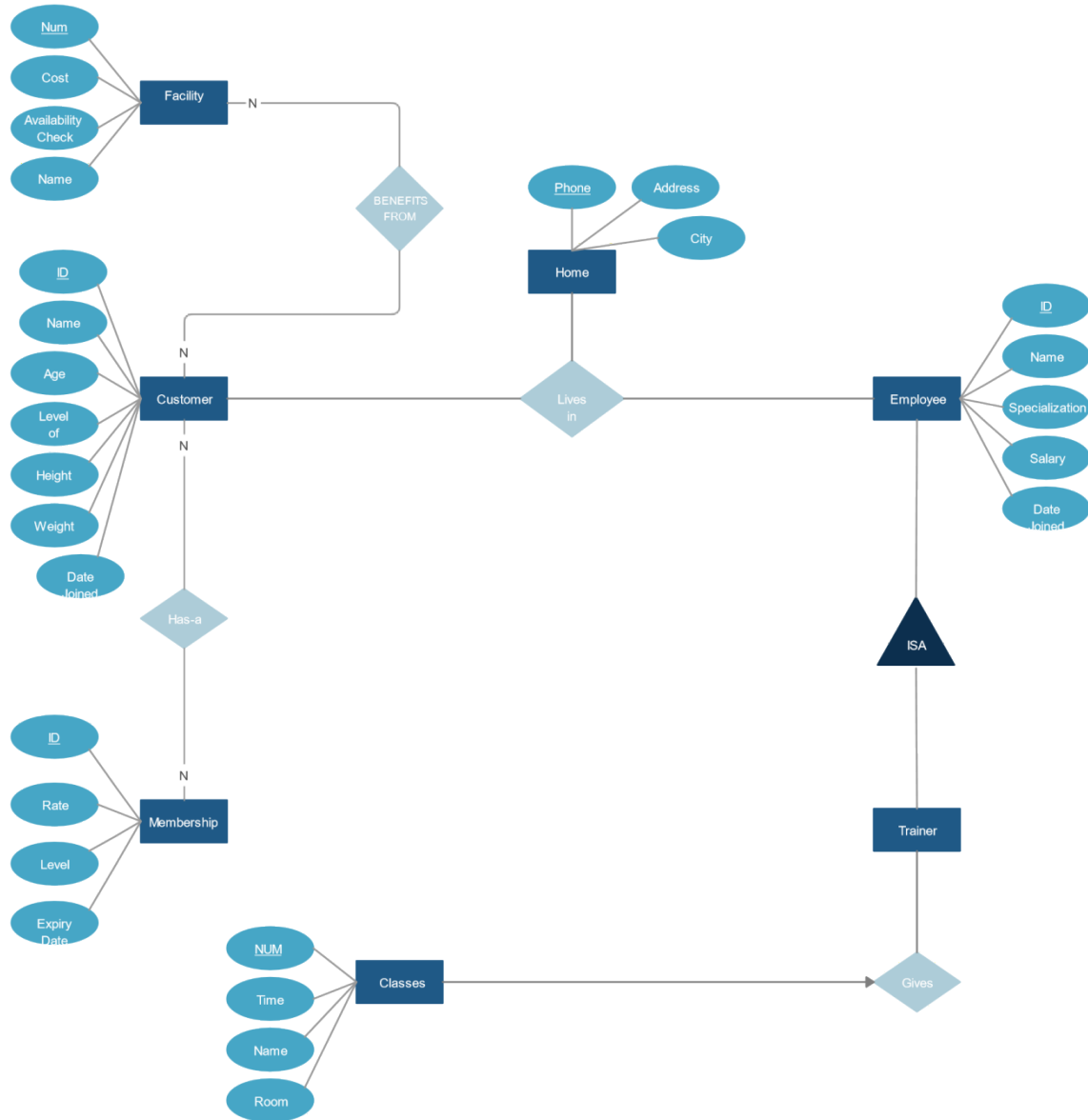
3.7. Optimization of solutions:
Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

4. Submission requirements:

4.1. High-level Document:
You have to create a high-level document design for your project. You can reference the HLD form below the link.

Num
Cost
Availability Check
Name
Facility
N
BENEFITS FROM

Phone Address
City
Home

ID
Name
Age
Level of
Height
Weight
Date Joined
Customer
N
N

Lives in

ID
Name
Specialization
Salary
Date Joined
Employee

Has-a

ID
Rate
Level
Expiry Date
Membership
N

ISA

Trainer

NUM
Time
Name
Room
Classes

Gives

Sample link: <u>HLD Document Link</u>

1.1. Low-level document:

You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: <u>LLD Document Link</u>

1.2. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: <u>Architecture sample link</u>

1.3. Wireframe:

 You have to create a Wireframe document design for your project; refer to the Wireframe fromthe link below.

Demo link: Wireframe Document Link

1.4. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submissionof your project.

Demo link: Project code sample link

1.5. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.Demo link: DPR sample link

4.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link.