**Problem Statement**:

**Design a Bowling Game:**

Game will consist of players and their scorecards which can be obtained at any point of the game. During the game, players and their scores will be maintained and shown by the system and winner will be declared at the end of the game.

Following are rules to be followed:

• In a single lane only 3 players are allowed to play. For ex: If there are 4 lanes (A, B, C and D) than each lane can have 3 players allocated for playing game.

• Game will consist of ten sets per player and in each set, each player has two opportunities to knock down ten pins (bottles in simple term)

so, in total a player has twenty chances for all ten sets.

* The score for a set is the total number of pins knocked down, plus bonuses for strikes and spares.
* A spare is when the player knocks down all ten pins in two tries. If there is spare the player gets 5 bonus points.
* A strike is when the player knocks down all ten pins on his/her first try. If there is a strike the player gets 10 bonus points.
* In the final set, a player who rolls a spare or a strike is allowed to roll the extra balls to complete the set. However, only a maximum of three balls can be rolled in the final set.

**Assumptions**:

You are free to use random number generator or any other way to simulate number of pins knocked down per chance/opportunity.

Feel free to make any other assumptions as well. All the assumptions should be documented in README file and also can be added as a description in the function performing the business logic.

**Application Design**:

Create a java, spring or spring boot application to implement following scenarios:

1. Build REST APIs to create, update or modify game with following details:

* **Player management:** Player details
* **Scoreboard management:** Current Score, Allocated lane, Total Score, Total Strikes, Current Strikes, Missed Strikes etc.

2. Create a REST API to start the game. This API should accept all the required input like number of players and their name and should return the start game id.

3. Build REST API, to get the score card of a player at any instance of the game i.e. after start of the game.

3. Store all the rules either in configuration file or Database (recommended)

4. Try to avoid hardcoding

5. Write JUnit test cases

6. Code should be push to any of GitHub, Bitbucket, GitLab

7. Code must be modular and proper use of OOPs and design pattern