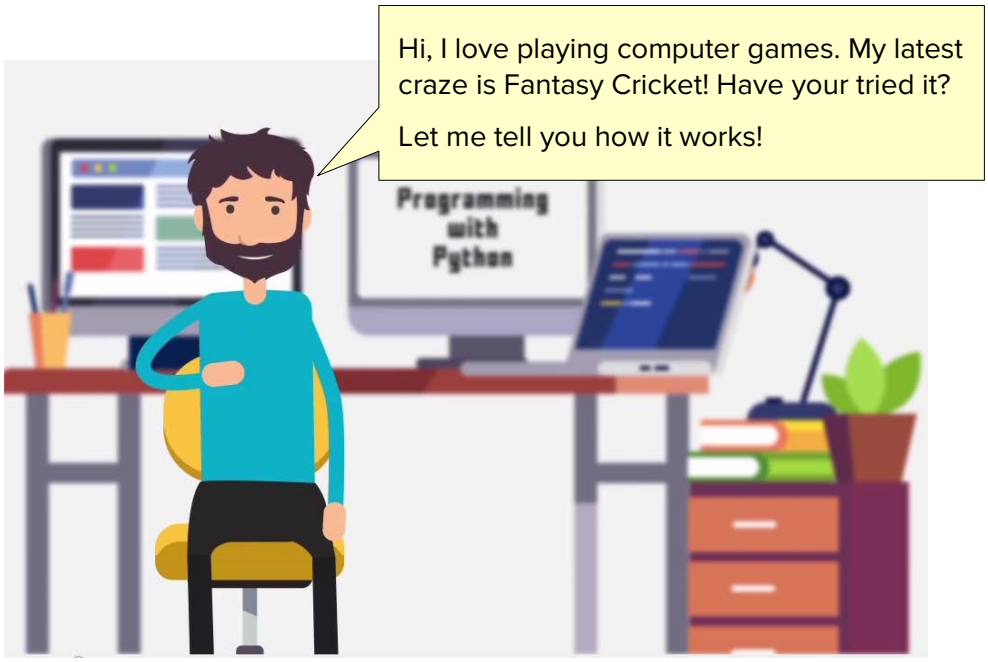
Introduction

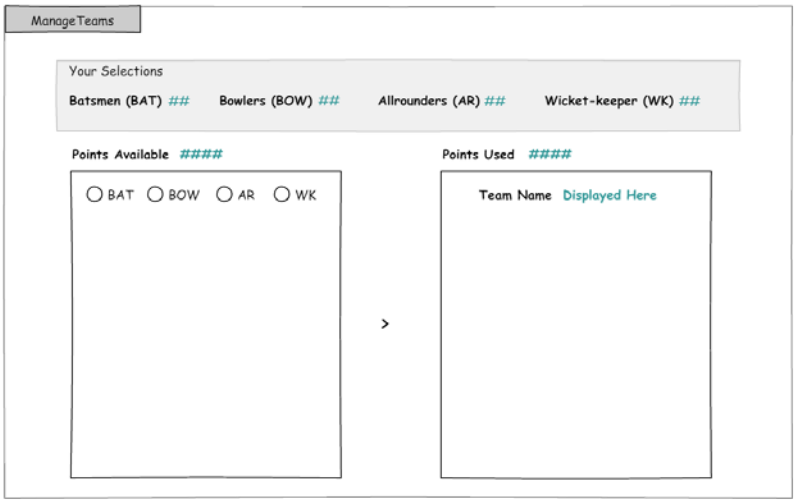
You have completed all the modules in this training. Now, it's time for you to apply your knowledge to create an application in Python. Read the scenario below and then respond to the problem statement described.

Scenario

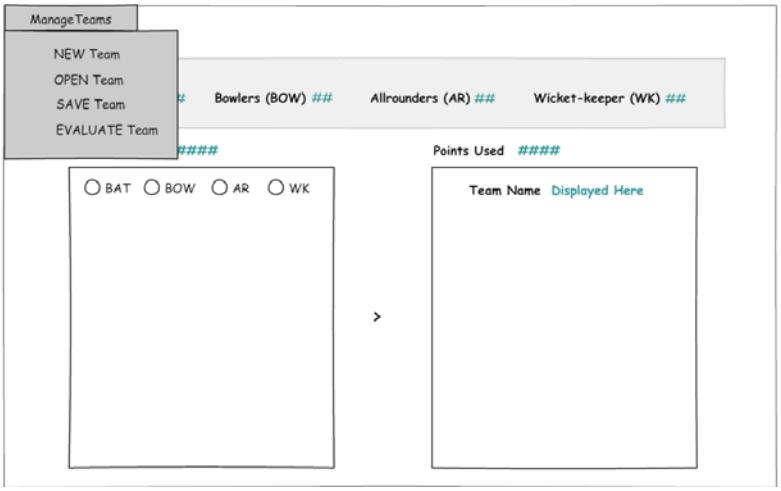


Fantasy Cricket

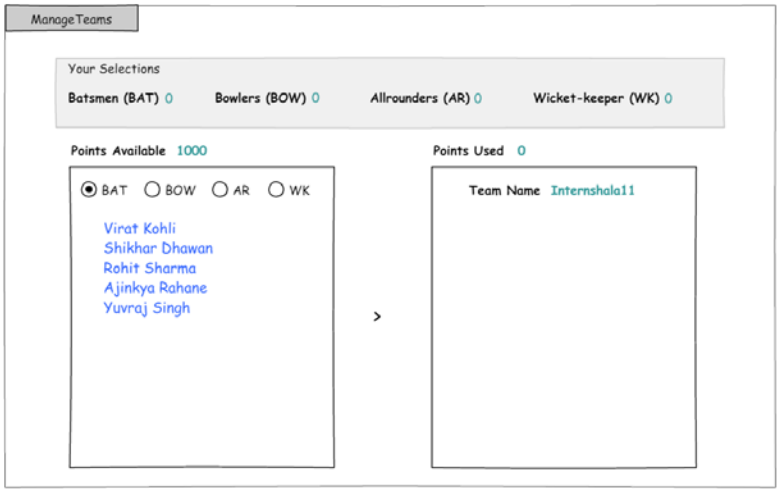
It is an online game where you create a virtual team of real cricket players and score points depending on how your chosen players perform in real life matches. To win a tournament, you must try and get the maximum points and the No. 1 rank amongst other participants. Here's how a Fantasy Cricket game may look like.



1 - Opening screen of the application. You can see the players of each category by selecting the category. To begin with, the selection is disabled until a new team is created from the Manage Teams menu. A pop up asking the name of the team appears.

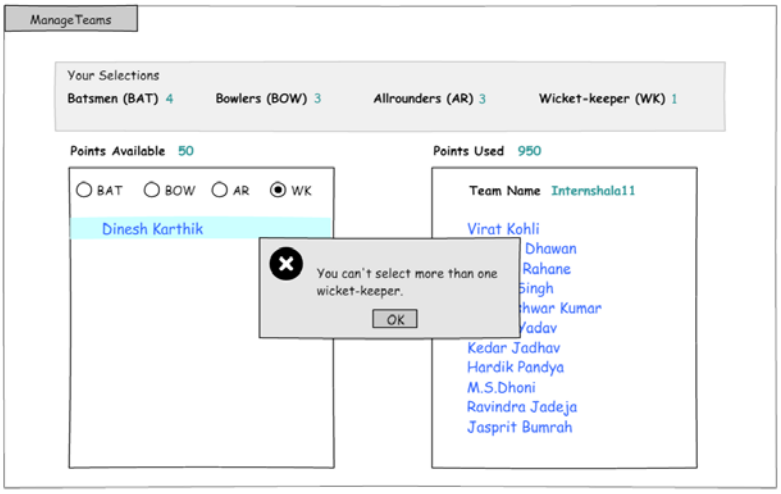


2 - The toolbar menu options which allow you to create a new team, open an existing team, save your team and finally evaluate the score of a saved team.

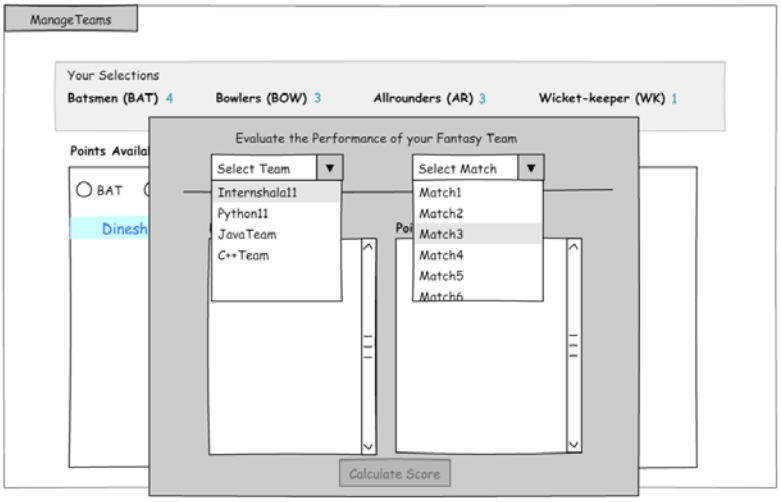


3 - After clicking New Team, the left box is populated with player names. As you select a different category, the corresponding list of players is displayed.

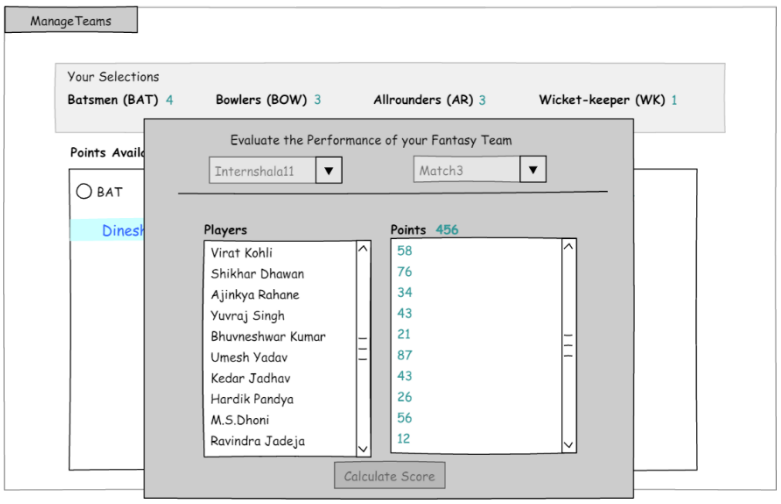
4 - On double-clicking each player name, the right box gets populated. Points available and used are displayed accordingly.



5 - Message if the game logic is not followed



6 - Pop-up on clicking Evaluate Score. You can select your team here and the match for which the players' performance is compared.



7 - The final score for your fantasy team based on the match selected.

Problem Statement

Create a Fantasy Cricket game in Python. The game should have all the features displayed in the mock-up screens in the scenario. To calculate the points for each player, you can use rules similar to the sample rules displayed below.

Sample of Rules

Batting

- 1 point for 2 runs scored
- Additional 5 points for half century
- Additional 10 points for century
- 2 points for strike rate (runs/balls faced) of 80-100
- Additional 4 points for strike rate>100
- 1 point for hitting a boundary (four) and 2 points for over boundary (six)

Bowling

- 10 points for each wicket
- Additional 5 points for three wickets per innings
- Additional 10 points for 5 wickets or more in innings
- 4 points for economy rate (runs given per over) between 3.5 and 4.5
- 7 points for economy rate between 2 and 3.5
- 10 points for economy rate less than 2

Fielding

- 10 points each for catch/stumping/run out

Database Design

For the database, you are required to use three tables – match, stats and teams.

match

Player	Scored	Faced	Fours	Sixes	Bowled	Maiden	Given	Wkts	Catches	Stumping	RO*

*Run Out

teams

name	players	value

stats

player	matches	runs	100s	50s	value	ctg

Note: The teams table will be populated after score calculation. The data to enter in the remaining 2 tables is given below:

Hint: If you are wondering where to start and how to plan your work, here are some suggestions.

i. First, create the database of players. Plan the required tables and add data to your database.
ii. Next, create the GUI. Generate the required Python code for the UI.
iii. Finally, populate the Python code generated in step ii with more attributes and method definitions (action listeners).

Assignment Submission

Your submission should have fully functional code with the required modules, packages and database files. These should be submitted as an archive file. The key elements that should be present in your submission are:

i. The main application code
ii. A Database of cricket players and their relevant stats
iii. The application .ui file

Project Evaluation Criteria

The following criteria will be used to evaluate your project submission.

Elements	Categories	Criteria
Application		
	Flow	Populate left list widget from database as per category radio button clicked
		Add players from players list to selected players list

		Remove players from selected players list to players list
		Show error if selection criterion is violated
		Calculate team score
	UI	Widgets should be placed symmetrically
Code		
	Naming Convention	Meaningful variable names
		Meaningful function names
	Modularity	A single function should be used for a single task/No code repetition (make generalized functions)
	Error Handling	Implement error handling wherever required
Database Design		
	Tables	Number of tables
		Column names
		Data types
		Data