Cal Poly Pomona

NBA Statistics Analyzer

May 13, 2025

Agenda

- **Presentation**
- **Demo**

Presenters

- ❑ Aryan Mitharwal
- ❑ Jerold Manansala
- ❑ Addison Sigsbury

Introduction

- Shared interest in basketball and the NBA
- Application built in JavaFX - NBA stat analyzer

  Search, sort, and compare players

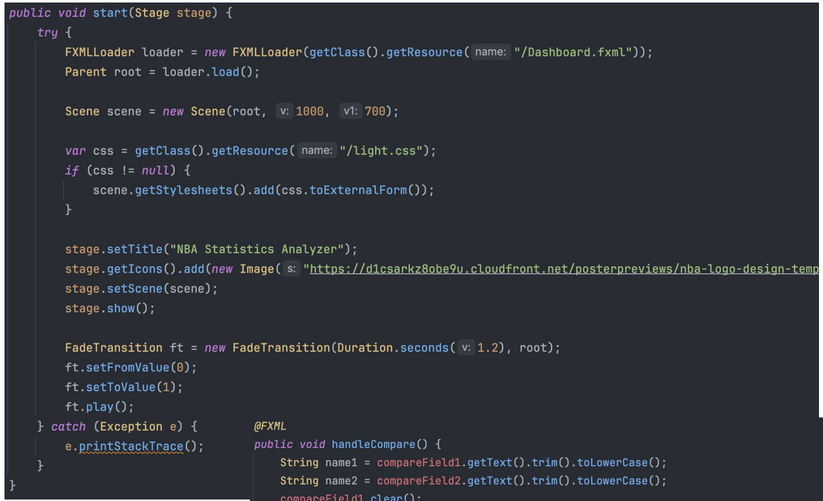
- Dataset from Kaggle

  Stats from seasons 1996 to 2023

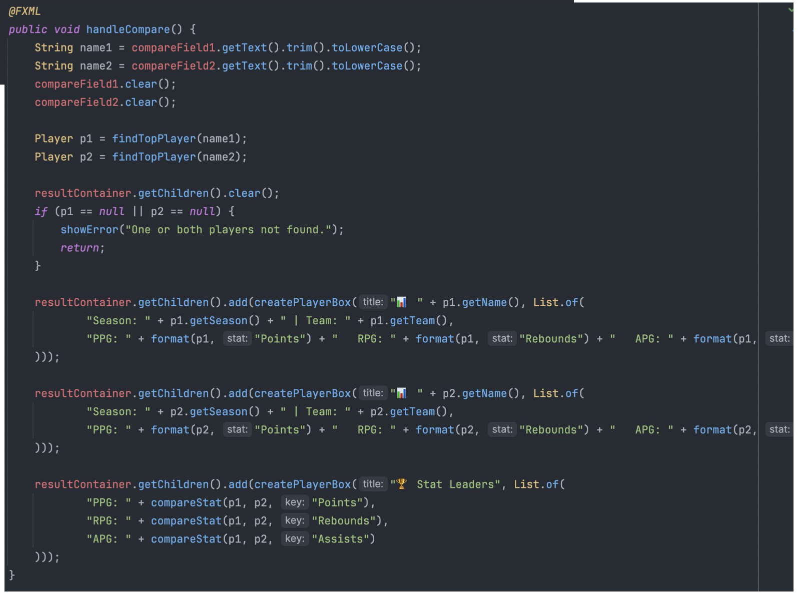
Main Methods

- Start() is what initializes the javafx application and loads the dashboard.
- It sets the title and adds necessary icons.

- HandleCompare() is the method that is used to actually compare two players.
- It takes the names of two different players and displays their necessary information like points, rebounds, and assists to the result container.



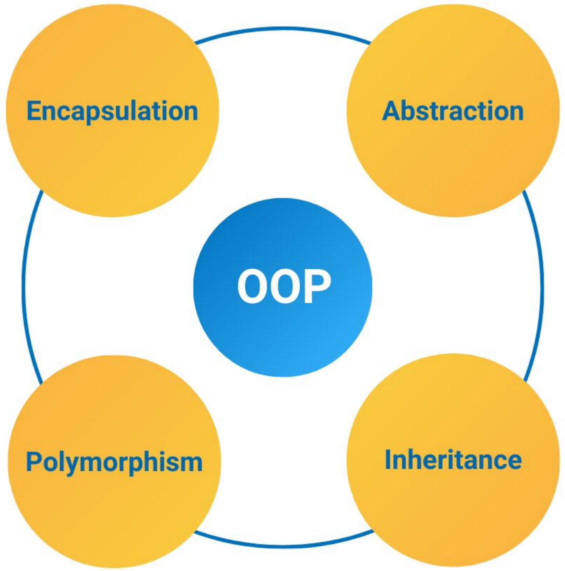
```java
public void start(Stage stage) {
    try {
        FXMLLoader loader = new FXMLLoader(getClass().getResource( name: "/Dashboard.fxml"));
        Parent root = loader.load();

        Scene scene = new Scene(root,  v: 1000,  v1: 700);

        var css = getClass().getResource( name: "/light.css");
        if (css != null) {
            scene.getStylesheets().add(css.toExternalForm());
        }

        stage.setTitle("NBA Statistics Analyzer");
        stage.getIcons().add(new Image( s: "https://d1csarkz8obe9u.cloudfront.net/posterpreviews/nba-logo-design-temp
        stage.setScene(scene);
        stage.show();

        FadeTransition ft = new FadeTransition(Duration.seconds( v: 1.2), root);
        ft.setFromValue(0);
        ft.setToValue(1);
        ft.play();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



```java
@FXML
public void handleCompare() {
    String name1 = compareField1.getText().trim().toLowerCase();
    String name2 = compareField2.getText().trim().toLowerCase();
    compareField1.clear();
    compareField2.clear();

    Player p1 = findTopPlayer(name1);
    Player p2 = findTopPlayer(name2);

    resultContainer.getChildren().clear();
    if (p1 == null || p2 == null) {
        showError("One or both players not found.");
        return;
    }

    resultContainer.getChildren().add(createPlayerBox( title: "📊  " + p1.getName(), List.of(
            "Season: " + p1.getSeason() + " | Team: " + p1.getTeam(),
            "PPG: " + format(p1,  stat: "Points") + "   RPG: " + format(p1,  stat: "Rebounds") + "   APG: " + format(p1,  stat:
    )));

    resultContainer.getChildren().add(createPlayerBox( title: "📊  " + p2.getName(), List.of(
            "Season: " + p2.getSeason() + " | Team: " + p2.getTeam(),
            "PPG: " + format(p2,  stat: "Points") + "   RPG: " + format(p2,  stat: "Rebounds") + "   APG: " + format(p2,  stat:
    )));

    resultContainer.getChildren().add(createPlayerBox( title: "🏆  Stat Leaders", List.of(
            "PPG: " + compareStat(p1, p2,  key: "Points"),
            "RPG: " + compareStat(p1, p2,  key: "Rebounds"),
            "APG: " + compareStat(p1, p2,  key: "Assists")
    )));
}
```

OOP Fundamentals Unveiled: Structuring Code

- Inheritance is used in our Main class by extending the JavaFX application class to construct our JavaFX program.
- We demonstrate polymorphism by instantiating the Player class and creating player objects to hold data like name, season, team etc.
- We applied abstraction by implementing classes that only expose necessary logic through public methods while keeping implementation logic within private methods and fields.
- Lastly, encapsulation was demonstrated through the use of private fields and restricted access to them.



Timeline

January

February

March

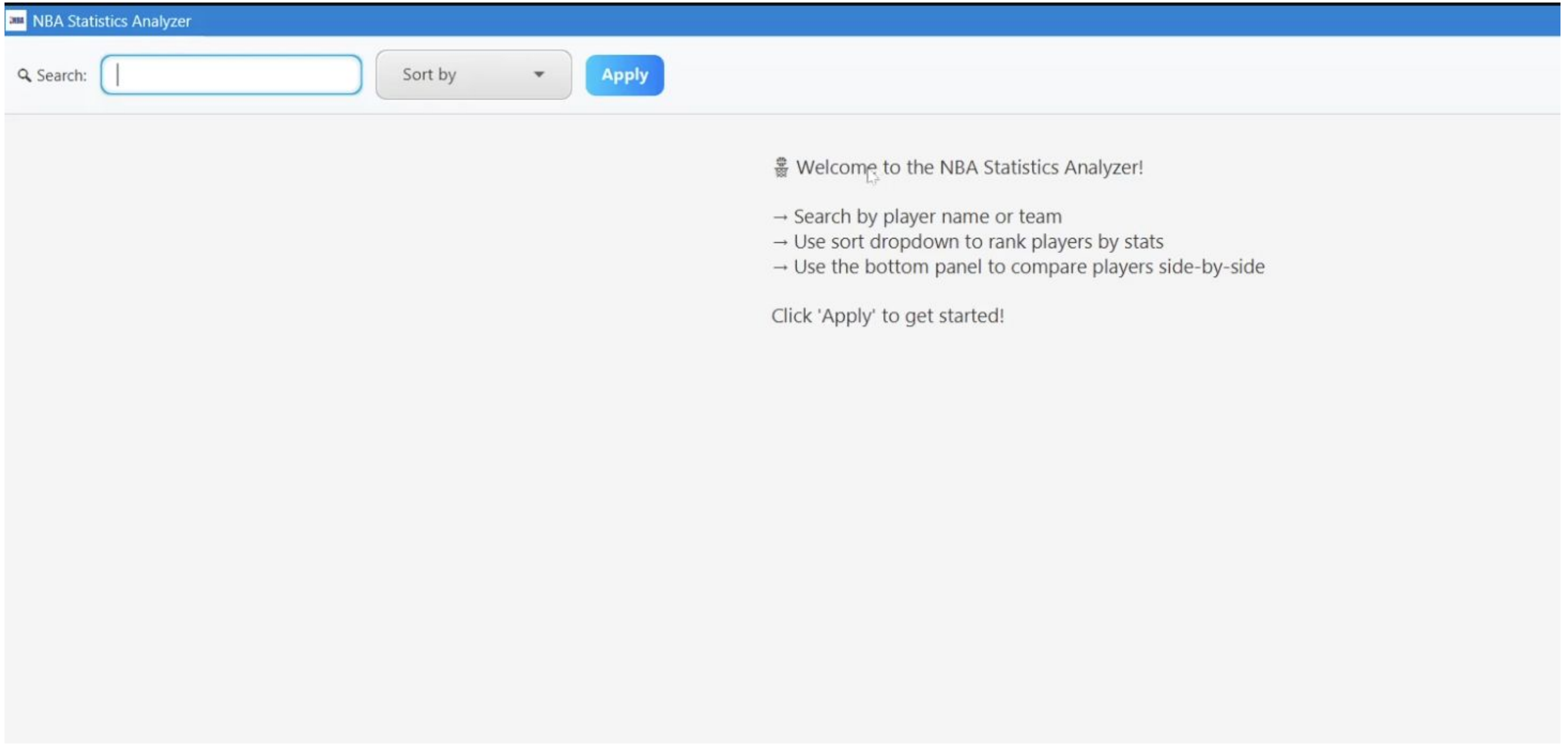
April

- February/March:
  Conceptualize the idea for our project and decide on what technologies to use and other implementation details.

- March/April:
  Develop a prototype of the project and work out most of the program minus a few final details.

- April/May:
  Finalize the project; fix any existing bugs, implement any extra features, add the finishing touches.

Screenshots of the Prototype

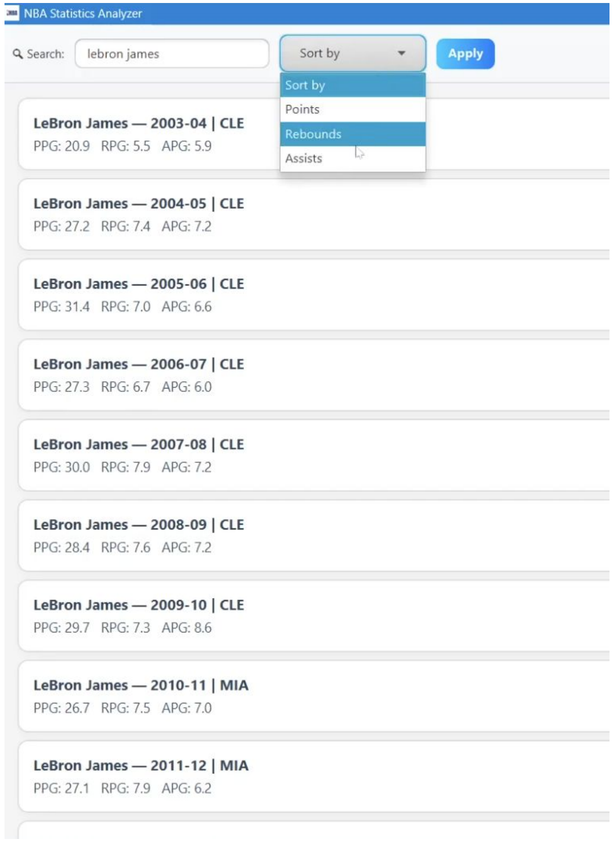


Welcome to the NBA Statistics Analyzer!

→ Search by player name or team
→ Use sort dropdown to rank players by stats
→ Use the bottom panel to compare players side-by-side

Click 'Apply' to get started!

Screenshots of the Prototype


NBA Statistics Analyzer

Search: lebron james | Sort by | Apply

LeBron James — 2003-04 | CLE
PPG: 20.9 RPG: 5.5 APG: 5.9

LeBron James — 2004-05 | CLE
PPG: 27.2 RPG: 7.4 APG: 7.2

LeBron James — 2005-06 | CLE
PPG: 31.4 RPG: 7.0 APG: 6.6

LeBron James — 2006-07 | CLE
PPG: 27.3 RPG: 6.7 APG: 6.0

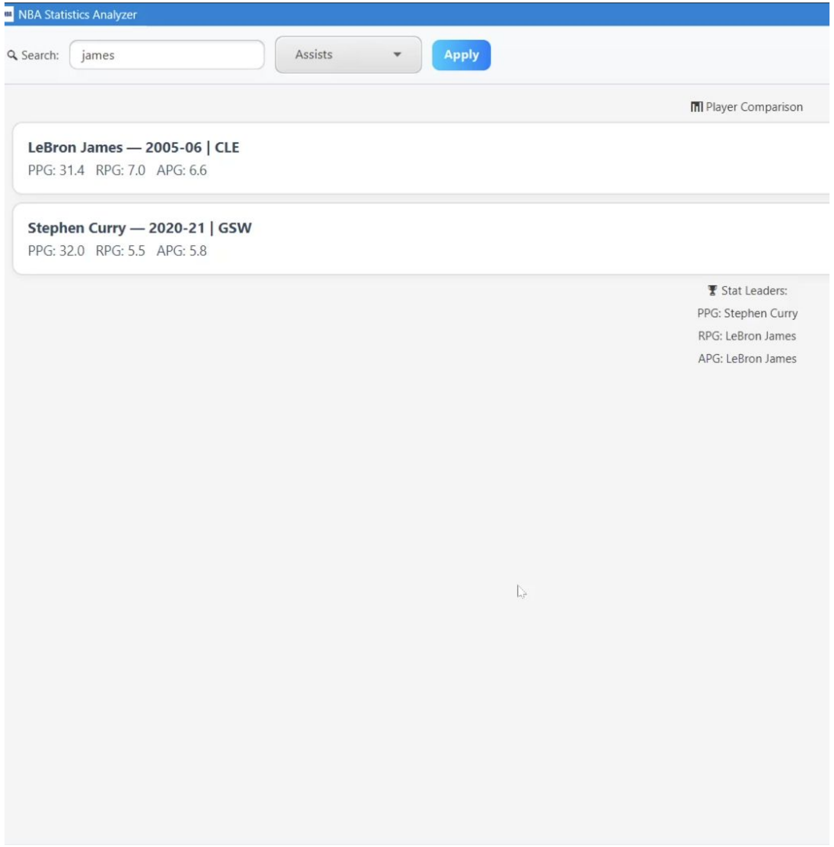
LeBron James — 2007-08 | CLE
PPG: 30.0 RPG: 7.9 APG: 7.2

LeBron James — 2008-09 | CLE
PPG: 28.4 RPG: 7.6 APG: 7.2

LeBron James — 2009-10 | CLE
PPG: 29.7 RPG: 7.3 APG: 8.6

LeBron James — 2010-11 | MIA
PPG: 26.7 RPG: 7.5 APG: 7.0

LeBron James — 2011-12 | MIA
PPG: 27.1 RPG: 7.9 APG: 6.2


NBA Statistics Analyzer

Search: james | Assists | Apply

Player Comparison

LeBron James — 2005-06 | CLE
PPG: 31.4 RPG: 7.0 APG: 6.6

Stephen Curry — 2020-21 | GSW
PPG: 32.0 RPG: 5.5 APG: 5.8

Stat Leaders:
PPG: Stephen Curry
RPG: LeBron James
APG: LeBron James