

```
# Project: Football Club Management System
```

```
## Project Background
```

You are the managers of a new football club! Your mission is to build a management system that will help

```
## Topics Covered in This Project
```

- Functions (regular and Arrow Functions)
- Objects (complex objects)
- Array Methods (filter, map, find, sort, reduce)
- Modules (ES6 import/export)

```
---
```

```
## Project Structure
```

```
```\nfootball-club/\n  data/\n    players.js\n  utils/\n    playerManager.js\n    statistics.js\n  main.js\n```\n
```

```
---
```

```
## Stage 1: Starter Code
```

```
### File: `data/players.js`
```

```
```\njavascript\n// Basic player data - DO NOT CHANGE THIS FILE\nexport const players = [\n  {\n    id: 1,\n    name: "John Smith",\n    position: "Goalkeeper",\n    age: 28,\n    goals: 0,\n    assists: 2,\n    matches: 15,\n    nationality: "England"\n  },\n  {\n    id: 2,\n    name: "David Johnson",\n    position: "Defender",\n    age: 25,\n    goals: 2,\n    assists: 3,\n    matches: 14,\n    nationality: "USA"\n  },\n  {\n    id: 3,\n    name: "Michael Brown",\n    position: "Midfielder",\n    age: 23,\n    goals: 5,\n    assists: 8,\n    matches: 15,\n    nationality: "Brazil"\n  },\n  {\n    id: 4,\n    name: "Alex Martinez",\n    position: "Forward",\n    age: 26,\n    goals: 12,\n  }\n]\n
```

```

        assists: 4,
        matches: 13,
        nationality: "Spain"
    },
    {
        id: 5,
        name: "Robert Wilson",
        position: "Forward",
        age: 24,
        goals: 8,
        assists: 5,
        matches: 15,
        nationality: "Argentina"
    },
    {
        id: 6,
        name: "James Taylor",
        position: "Midfielder",
        age: 27,
        goals: 3,
        assists: 6,
        matches: 12,
        nationality: "Germany"
    },
    {
        id: 7,
        name: "Chris Anderson",
        position: "Defender",
        age: 29,
        goals: 1,
        assists: 1,
        matches: 15,
        nationality: "France"
    }
];

// Additional export - club information
export const clubInfo = {
    name: "Champions FC",
    founded: 2020,
    stadium: "Victory Arena",
    city: "London"
};

### File: `utils/playerManager.js`

```javascript
// Import player data
import { players } from '../data/players.js';

// EXAMPLE FUNCTION #1 - Study this example!
// Function to get all players
export function getAllPlayers() {
    return players;
}

// EXAMPLE FUNCTION #2 - Study this example!
// Function to find a player by ID using arrow function
export const findPlayerById = (id) => {
    return players.find(player => player.id === id);
};

// =====
// YOUR FUNCTIONS START HERE - WRITE THEM!
// =====

// TODO 1: Write a function to add a new player
// The function should:
// - Accept playerData object as parameter
// - Create new player with id = players.length + 1
// - Set default values: matches: 0, goals: 0, assists: 0

```

```

// - Add the new player to the players array
// - Return the new player
// export function addPlayer(playerData) {
//     // Your code here...
// }

// TODO 2: Write a function to remove a player by id
// The function should:
// - Accept id as parameter
// - Find the player index in the array
// - Remove the player from the array
// - Return the removed player (or null if not found)
// export function removePlayer(id) {
//     // Your code here...
// }

// TODO 3: Write a function to update player data
// The function should:
// - Accept id and newData as parameters
// - Find the player by id
// - Update the player's properties with newData
// - Return the updated player
// export function updatePlayer(id, newData) {
//     // Your code here...
// }

// TODO 4: Write a function to find player by name (partial match)
// The function should:
// - Accept name as parameter (string)
// - Return player whose name includes the search term
// - Make the search case-insensitive
// Use: find() and includes() methods
// export const findPlayerByName = (name) => {
//     // Your code here...
// };
...

### File: `utils/statistics.js`

```javascript
import { players } from '../data/players.js';

// EXAMPLE FUNCTION #1 - Study this example!
// Function to calculate total team goals using reduce
export function getTotalGoals() {
    return players.reduce((total, player) => total + player.goals, 0);
}

// EXAMPLE FUNCTION #2 - Study this example!
// Function to get players by position using filter
export function getPlayersByPosition(position) {
    return players.filter(player => player.position === position);
}

// =====
// YOUR FUNCTIONS START HERE - WRITE THEM!
// =====

// TODO 5: Write a function to calculate total assists
// Use reduce() method
// export function getTotalAssists() {
//     // Your code here...
// }

// TODO 6: Write a function to find the top scorer
// The function should:
// - Find the player with the most goals
// - Return that player object
// Use: reduce() method
// export const getTopScorer = () => {
//     // Your code here...
// };

```

```

// TODO 7: Write a function to calculate average age
// The function should:
// - Calculate the sum of all player ages
// - Divide by number of players
// - Return the result rounded to 1 decimal place
// Use: reduce() and toFixed() methods
// export function getAverageAge() {
//     // Your code here...
// }

// TODO 8: Write a function to get players by nationality
// Use filter() method
// export function getPlayersByNationality(nationality) {
//     // Your code here...
// }

// TODO 9: Write a function to sort players by goals (highest to lowest)
// The function should:
// - Create a copy of the players array (use spread operator)
// - Sort by goals in descending order
// - Return the sorted array
// Use: sort() method
// export function sortPlayersByGoals() {
//     // Your code here...
// }

// TODO 10: Write a function to get young players (under certain age)
// The function should:
// - Accept maxAge as parameter
// - Return all players younger than maxAge
// Use: filter() method
// export function getYoungPlayers(maxAge) {
//     // Your code here...
// }

// TODO 11: Write a function to get players with minimum goals
// The function should:
// - Accept minGoals as parameter
// - Return all players with at least minGoals goals
// Use: filter() method
// export function getPlayersWithMinGoals(minGoals) {
//     // Your code here...
// }

// TODO 12: Write a function to calculate player score
// Score formula: (goals * 3) + (assists * 2)
// The function should:
// - Accept player object as parameter
// - Return the calculated score
// export function calculatePlayerScore(player) {
//     // Your code here...
// }

// TODO 13: Write a function to get top 3 scorers
// The function should:
// - Sort players by goals
// - Return only the first 3 players
// Use: sort() and slice() methods
// export function getTop3Scorers() {
//     // Your code here...
// }

// TODO 14: Write a function to get all player names
// The function should:
// - Return an array of just the player names (strings)
// Use: map() method
// export function getAllPlayerNames() {
//     // Your code here...
// }

// TODO 15: Write a function to calculate goals per match for a player
// The function should:

```

```

// - Accept player object as parameter
// - Calculate goals / matches
// - Return result rounded to 2 decimal places
// Handle division by zero (return 0 if matches = 0)
// export function getGoalsPerMatch(player) {
//     // Your code here...
// }
```

### File: `main.js`

```javascript
import { players, clubInfo } from '../data/players.js';
// Import your functions here as you create them
// Example:
// import { getAllPlayers, findPlayerById, addPlayer } from '../utils/playerManager.js';
// import { getTotalGoals, getPlayersByPosition } from '../utils/statistics.js';

console.log('='.repeat(50));
console.log( FOOTBALL CLUB MANAGEMENT SYSTEM');
console.log('='.repeat(50));

// Display club information
console.log('\nCLUB INFORMATION');
console.log('■'.repeat(50));
console.log(`Name: ${clubInfo.name}`);
console.log(`Founded: ${clubInfo.founded}`);
console.log(`Stadium: ${clubInfo.stadium}`);
console.log(`City: ${clubInfo.city}`);

// TODO: After you write getAllPlayers(), use it here
// console.log('\nPLAYER ROSTER');
// console.log('■'.repeat(50));
// getAllPlayers().forEach(player => {
//     console.log(`#${player.id} ${player.name} - ${player.position} - Age ${player.age}`);
// });

// TODO: After you write getTotalGoals(), use it here
// console.log('\nTEAM STATISTICS');
// console.log('■'.repeat(50));
// console.log(`Total Goals: ${getTotalGoals()}`);

// TODO: Test your functions here!
// Example tests:

// Test findPlayerById
// const player = findPlayerById(3);
// console.log('\nFound Player:', player);

// Test getPlayersByPosition
// const forwards = getPlayersByPosition('Forward');
// console.log('\nForwards:', forwards);

// Test addPlayer
// const newPlayer = addPlayer({
//     name: "Carlos Rodriguez",
//     position: "Midfielder",
//     age: 22,
//     nationality: "Mexico"
// });
// console.log('\n■ Added Player:', newPlayer);

console.log('\n' + '='.repeat(50));
console.log('Write your functions and test them here!');
console.log('='.repeat(50));

---

## Your Mission

### Required Functions (Must Complete All!)
#### In `utils/playerManager.js`

```

```

1. `addPlayer(playerData)` - Add new player to array
2. `removePlayer(id)` - Remove player by id
3. `updatePlayer(id, newData)` - Update player information
4. `findPlayerByName(name)` - Find player by name (partial match)

#### In `utils/statistics.js`

5. `getTotalAssists()` - Calculate total team assists
6. `getTopScorer()` - Find player with most goals
7. `getAverageAge()` - Calculate average age of players
8. `getPlayersByNationality(nationality)` - Filter by nationality
9. `sortPlayersByGoals()` - Sort players by goals (high to low)
10. `getYoungPlayers(maxAge)` - Get players under certain age
11. `getPlayersWithMinGoals(minGoals)` - Get players with minimum goals
12. `calculatePlayerScore(player)` - Calculate score: (goals × 3) + (assists × 2)
13. `getTop3Scorers()` - Get the top 3 goal scorers
14. `getAllPlayerNames()` - Get array of all player names
15. `getGoalsPerMatch(player)` - Calculate goals per match average

```

---

## ## Testing Your Functions

After writing each function:

```

1. Import it in `main.js`
2. Call the function with test data
3. Use `console.log()` to see the results
4. Make sure it works correctly before moving to the next one

```

## ### Example Test

```

```javascript
// After writing getTopScorer()
import { getTopScorer } from './utils/statistics.js';

const topScorer = getTopScorer();
console.log('Top Scorer:', topScorer.name, '-', topScorer.goals, 'goals');
```

```

---

## ## Array Methods You Must Use

Make sure to use these array methods in your functions:

```

- `filter()` - Filter players by criteria
- `map()` - Transform player data
- `find()` - Find specific player
- `reduce()` - Calculate totals/find max
- `sort()` - Order players
- `slice()` - Get portion of array
- `forEach()` - Iterate through players

```

---

## ## Bonus Challenges

After completing all required functions, try these:

### ### Bonus 1: Team Formation Builder

Create file: `utils/teamBuilder.js`

```

```javascript
// Build a 4-4-2 formation
export function buildFormation() {
  // Get 1 Goalkeeper, 4 Defenders, 4 Midfielders, 2 Forwards
  // Return object with formation
}

// Display formation nicely in console

```



Name: Champions FC  
Founded: 2020  
Stadium: Victory Arena  
City: London

#### PLAYER ROSTER

#1 John Smith - Goalkeeper - Age 28  
#2 David Johnson - Defender - Age 25  
...

#### TEAM STATISTICS

Total Goals: 31  
Total Assists: 29  
Average Age: 26.0  
Top Scorer: Alex Martinez (12 goals)

#### TOP 3 SCORERS

1. Alex Martinez - 12 goals  
2. Robert Wilson - 8 goals  
3. Michael Brown - 5 goals  
...

---

#### ## Common Mistakes to Avoid

1. **\*\*Don't modify the players.js file\*\*** - Only read from it
2. **\*\*Don't forget to export\*\*** your functions
3. **\*\*Don't forget to import\*\*** functions before using them
4. **\*\*Don't use var\*\*** - Use const or let
5. **\*\*Don't mutate original array\*\*** when sorting (use spread operator)
6. **\*\*Handle edge cases\*\*** - What if array is empty? Player not found?

---

#### ## Setup Instructions

1. Create the folder structure
2. Create all 5 files with the provided code
3. Open the project with Live Server
4. Open browser console (F12)
5. Start writing functions!
6. Test each function as you write it
7. Have fun!

---

**\*\*Good luck! Write clean code and test everything!\*\***