INSTITUTE OF AERONAUTICAL ENGINEERING
DUNDIGAL, HYDERABAD-500 043, TELANGANA,
INDIA.

Front-end Web Development

COURSE CODE: ACSE04

TASK2

Title: TeamStats

By:

24951A6660 –

SOMEPALLI HASINI

From: CSE(AI/ML)

INSTITUTE OF AERONAUTICAL ENGINEERING
(AUTONOMOUS)
Dundigal - 500 043, Hyderabad, Telangana

Complex Problem Solving Self-Assessment Form

1	Name of the Student	Somepalli Hasini
2	Roll Number	24951A6660
3	Branch and Section	CSE(AI/ML)
4	Program	B. Tech
5	Course Name	Front end web development
6	Course Code	ACSE04
7	Please tick (✓) relevant Engineering Competency (ECs) Profiles	

	EC	Profiles	(✓)
	EC 1	Ensures that all aspects of an engineering activity are soundly based on fundamental principles - by diagnosing, and taking appropriate action with data, calculations, results, proposals, processes, practices, and documented information that may be ill-founded, illogical, erroneous, unreliable or unrealistic requirements applicable to the engineering discipline	(✓)
	EC 2	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	(✓)
	EC 3	Support sustainable development solutions by ensuring functional requirements, minimize environmental impact and optimize resource utilization throughout the life cycle, while balancing performance and cost effectiveness.	(✓)
	EC 4	Competently addresses complex engineering problems which involve uncertainty, ambiguity, imprecise information and wide-ranging or conflicting technical, engineering and other issues.	(✓)
	EC 5	Conceptualizes alternative engineering approaches and evaluates potential outcomes against appropriate criteria to justify an optimal solution choice.	(✓)
	EC 6	Identifies, quantifies, mitigates and manages technical, health, environmental, safety, economic and other contextual risks associated to seek achievable sustainable outcomes with engineering application in the designated engineering discipline.	(✓)
	EC 7	Involve the coordination of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies) in the timely delivery of outcomes	(✓)
	EC 8	Design and develop solution to complex engineering problem considering a very perspective and taking account of stakeholder views with widely varying needs.	(✓)
	EC 9	Meet all level, legal, regulatory, relevant standards and codes of practice, protect public health and safety in the course of all engineering activities.	(✓)
	EC 10	High level problems including many component parts or sub-problems, partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the top consideration.	(✓)

	EC	Profiles	(√)
	EC 11	Undertake CPD activities to maintain and extend competences and enhance the ability to adapt to emerging technologies and the ever-changing nature of work.	(√)
	EC 12	Recognize complexity and assess alternatives in light of competing requirements and incomplete knowledge. Require judgement in decision making in the course of all complex engineering activities.	(√)
8	Please tick (√) relevant Course Outcomes (COs) Covered		
	CO	**Course Outcomes**	(√)
	CO 1	Understand the fundamental concepts of sorting algorithms and their significance in data processing for large-scale applications.	(√)
	CO 2	Analyze the limitations of traditional sorting algorithms when applied to big data environments.	(√)
	CO 3	Design and implement hybrid sorting algorithms in Java that combine multiple techniques (e.g., merge-sort, quick-sort, and heap-sort) for improved performance.	(√)
	CO 4	Evaluate the computational efficiency, time complexity, and scalability of various hybrid sorting approaches for large datasets..	(√)
	CO 5	Apply Java-based parallel and distributed programming concepts (e.g., multithreading, Hadoop/MapReduce) to optimize hybrid sorting performance in big data contexts.	(√)
	CO 6	Compare hybrid sorting techniques using performance metrics such as execution time, memory utilization, and throughput on different data sizes.	(√)
	CO 7	Develop a mini-project or case study demonstrating the practical application of hybrid sorting in real-world big data scenarios such as data analytics, machine learning preprocessing, or cloud computing systems.	(√)

9	Course ELRV Video Lectures Viewed	**Number of Videos**	**Viewing time in Hours**
10	Justify your understanding of WK1	Foundations for analysis and optimizing operations	
11	Justify your understanding of WK2 – WK9	Core to advanced concepts, tools, design, and ethics.	
12	How many WKs from WK2 to WK9 were implanted?	All 8 WKs from WK2 – WK9 are implemented in this and analysis	
	Mention them	WK2 to WK9	

Hasini S.

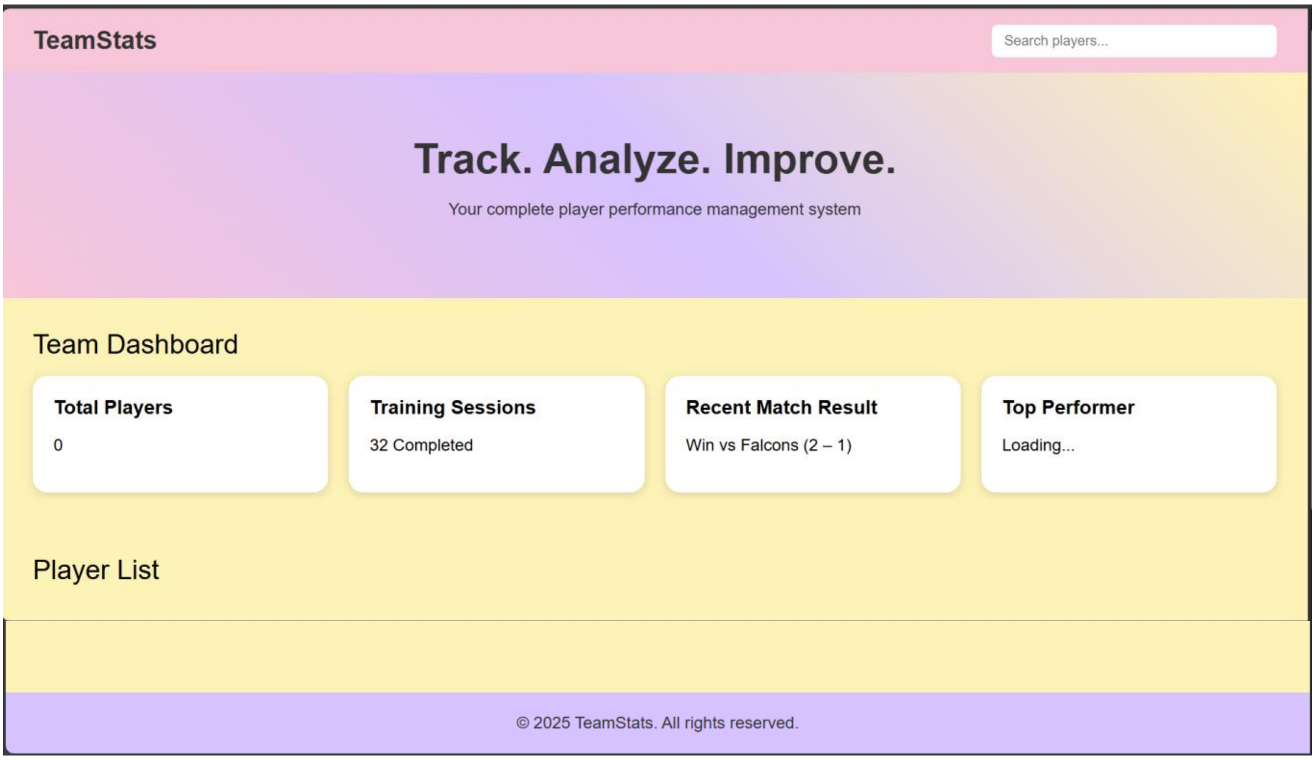
Date:12/12/25

Signature of the Student

1. Title and Domain

Title: *TeamStats – Player Performance Monitoring and Dashboard System*
Domain: *Web Development / Sports Analytics / Frontend Engineering*

TeamStats is a web-based frontend platform designed to assist coaches, team managers, and athletes in maintaining performance insights. The system focuses heavily on **player statistics**, **training progress tracking**, **dashboard visualization**, and an interface optimized for clarity and usability. The project sits within the domain of **sports analytics tools**, implemented using foundational web technologies such as **HTML**, **CSS**, and **JavaScript**.



TeamStats

Track. Analyze. Improve.

Your complete player performance management system

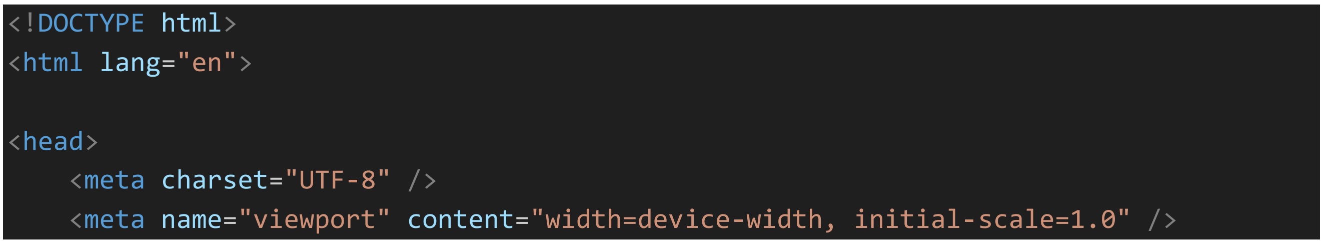
Team Dashboard

Total Players	Training Sessions	Recent Match Result	Top Performer
0	32 Completed	Win vs Falcons (2 – 1)	Loading...

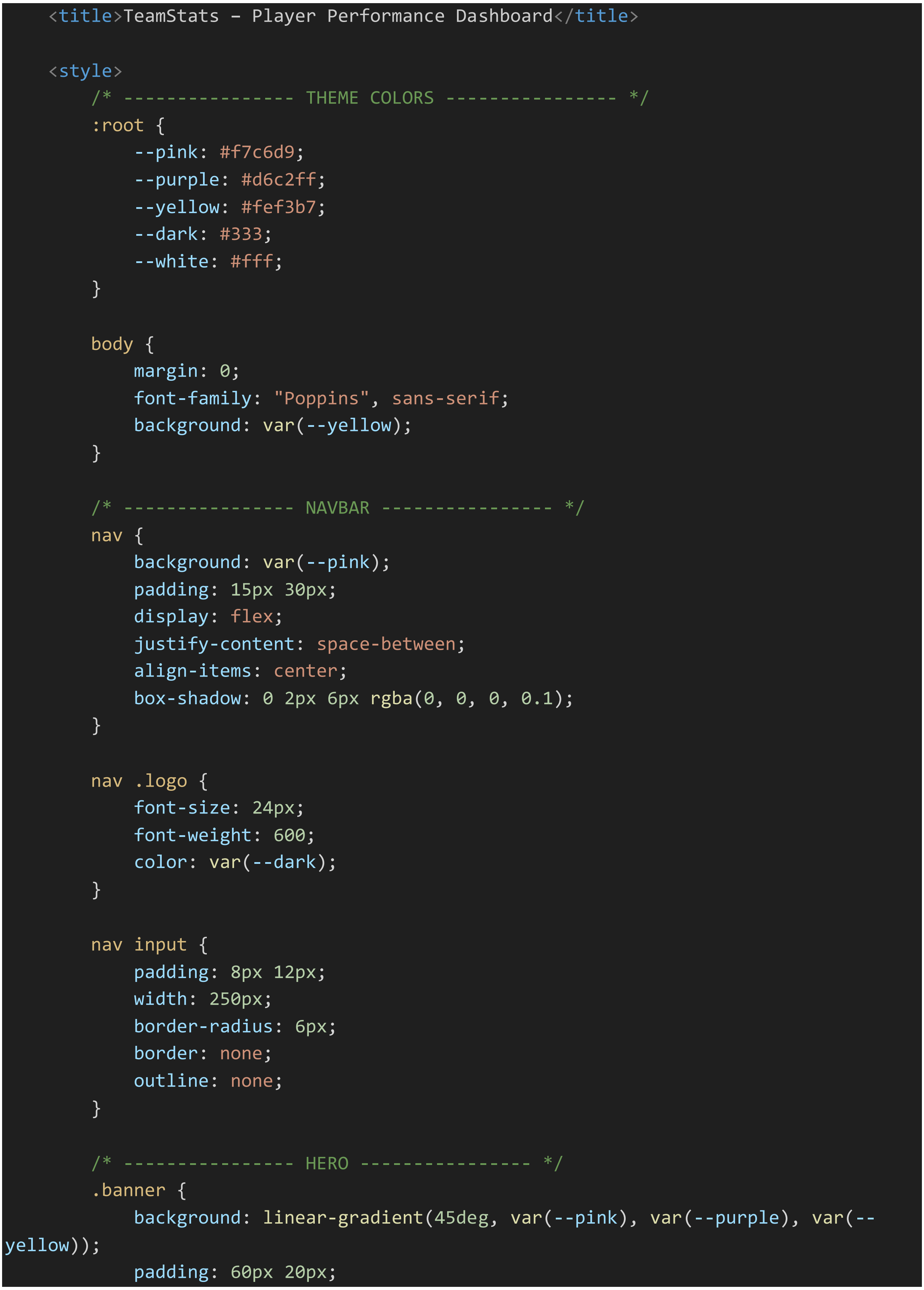
Player List

© 2025 TeamStats. All rights reserved.

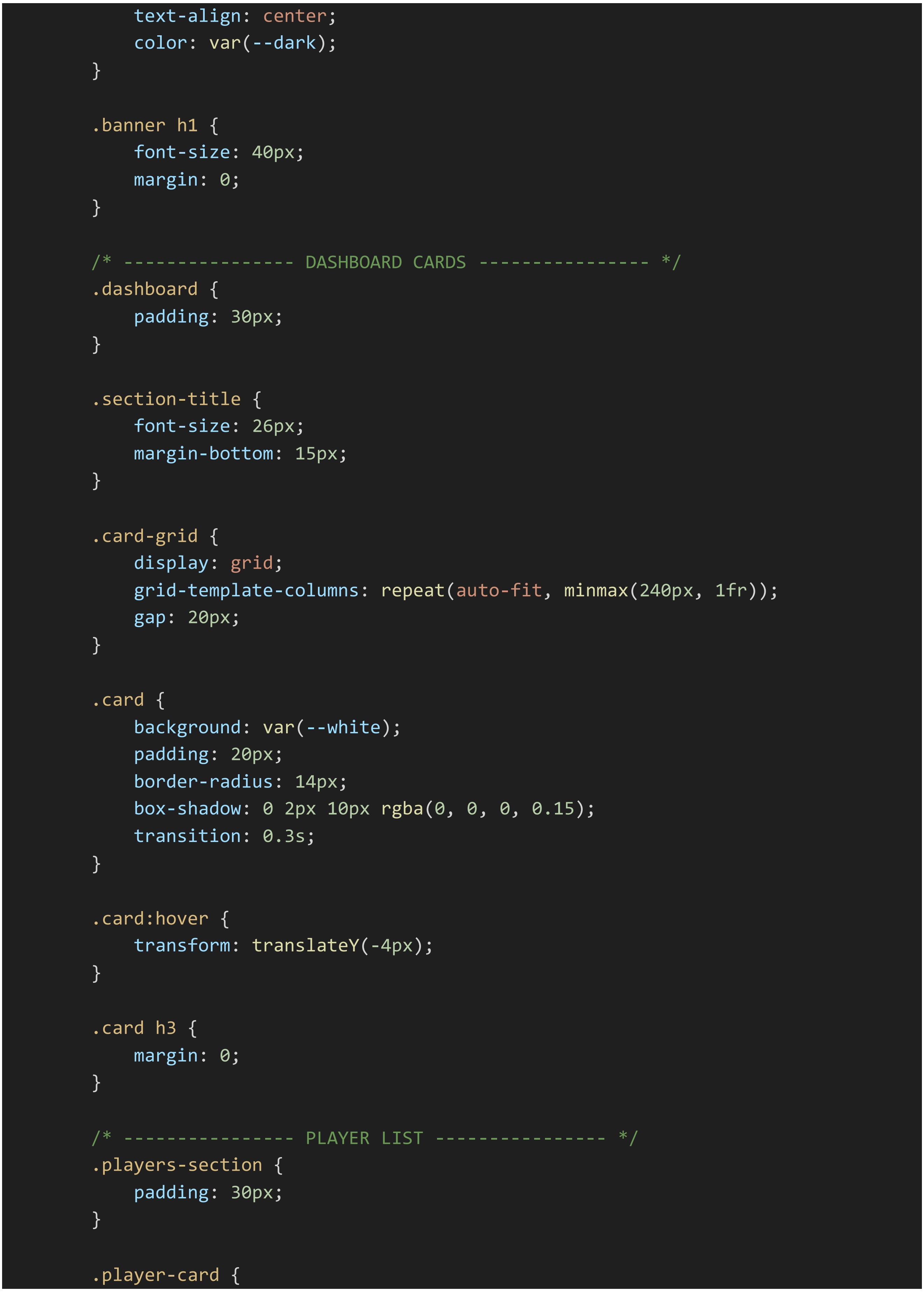
CODE:



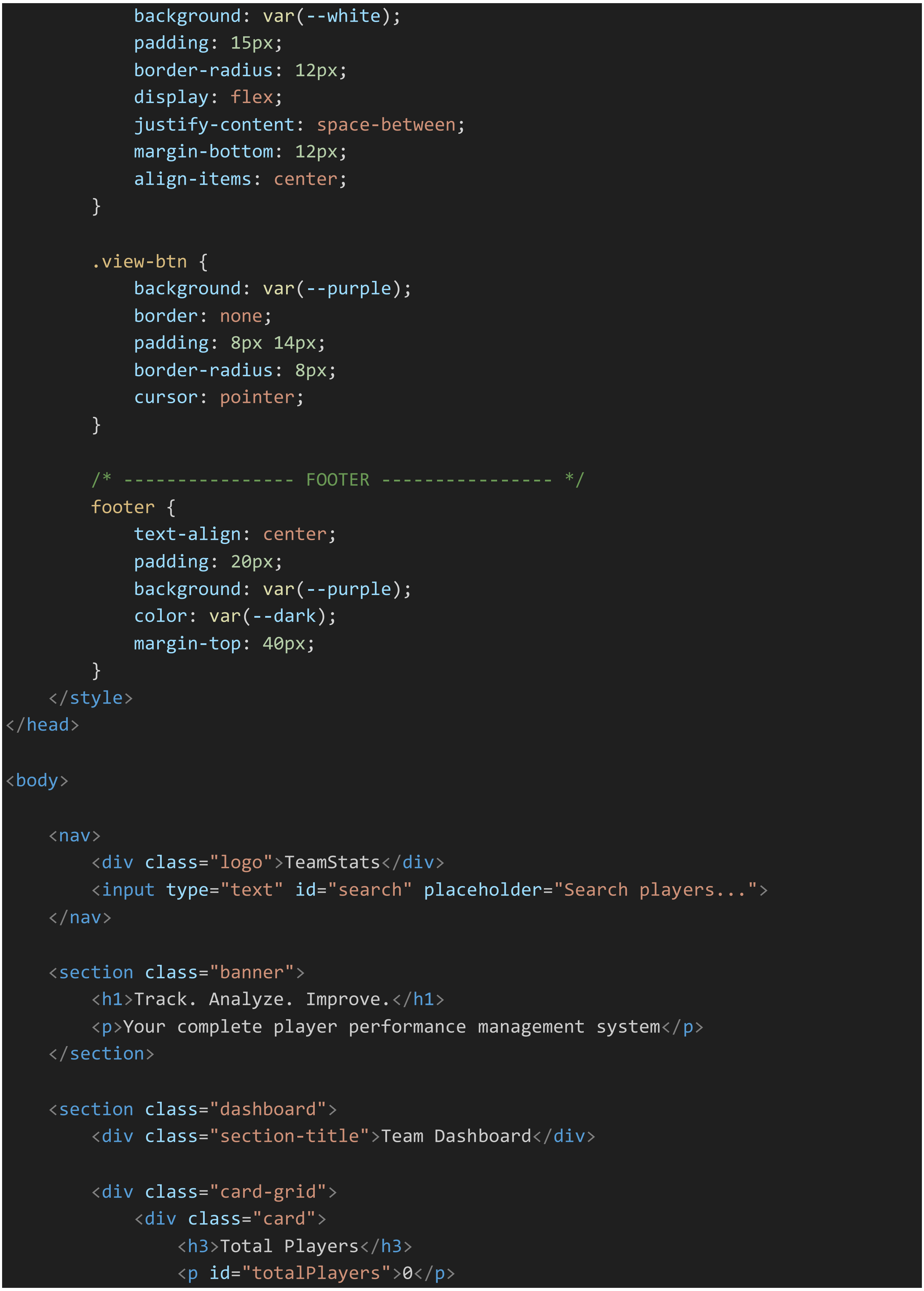
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```



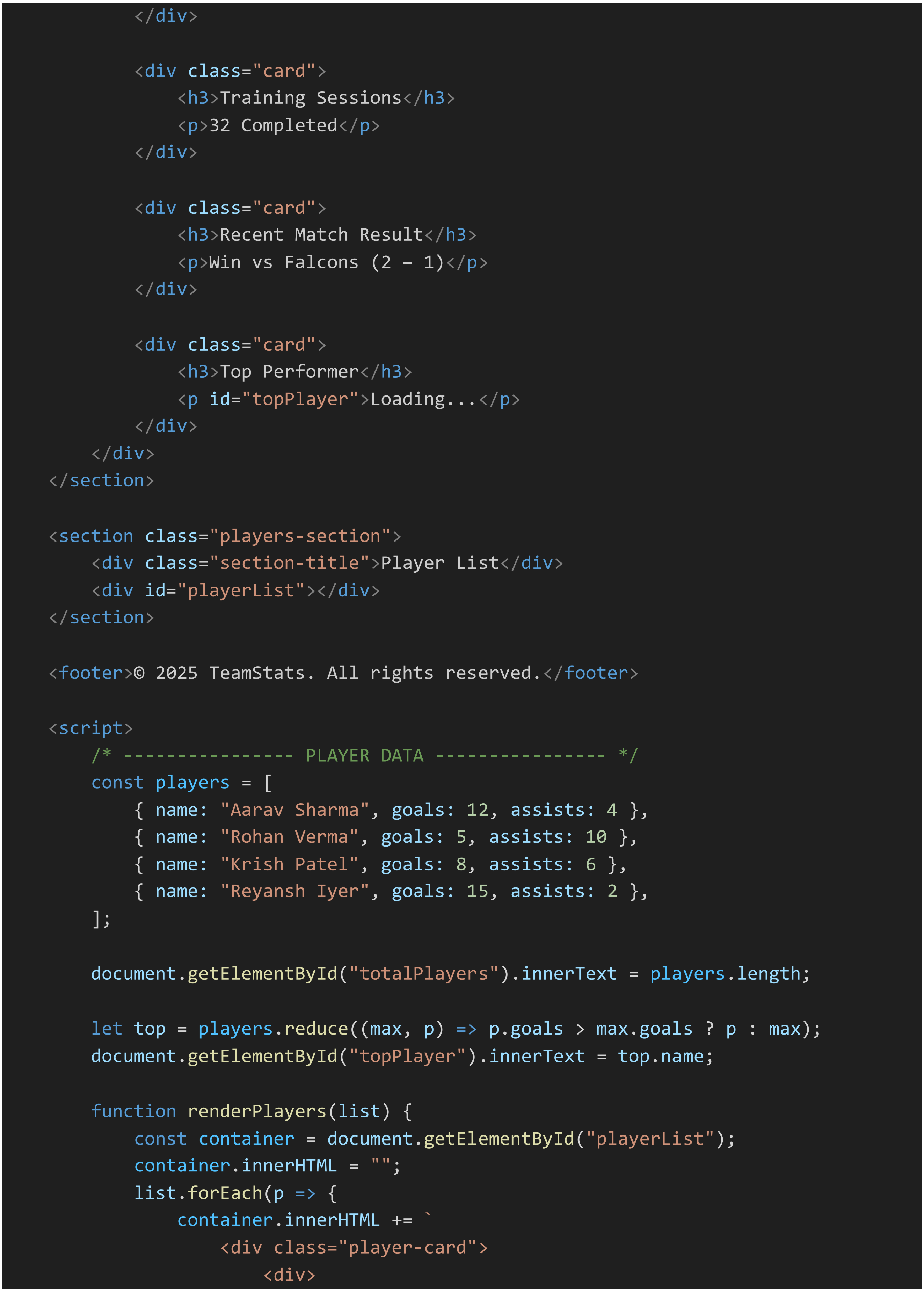
```
    <title>TeamStats – Player Performance Dashboard</title>

    <style>
        /* ---------------- THEME COLORS ---------------- */
        :root {
            --pink: #f7c6d9;
            --purple: #d6c2ff;
            --yellow: #fef3b7;
            --dark: #333;
            --white: #fff;
        }

        body {
            margin: 0;
            font-family: "Poppins", sans-serif;
            background: var(--yellow);
        }

        /* ---------------- NAVBAR ---------------- */
        nav {
            background: var(--pink);
            padding: 15px 30px;
            display: flex;
            justify-content: space-between;
            align-items: center;
            box-shadow: 0 2px 6px rgba(0, 0, 0, 0.1);
        }

        nav .logo {
            font-size: 24px;
            font-weight: 600;
            color: var(--dark);
        }

        nav input {
            padding: 8px 12px;
            width: 250px;
            border-radius: 6px;
            border: none;
            outline: none;
        }

        /* ---------------- HERO ---------------- */
        .banner {
            background: linear-gradient(45deg, var(--pink), var(--purple), var(--
yellow));
            padding: 60px 20px;
```



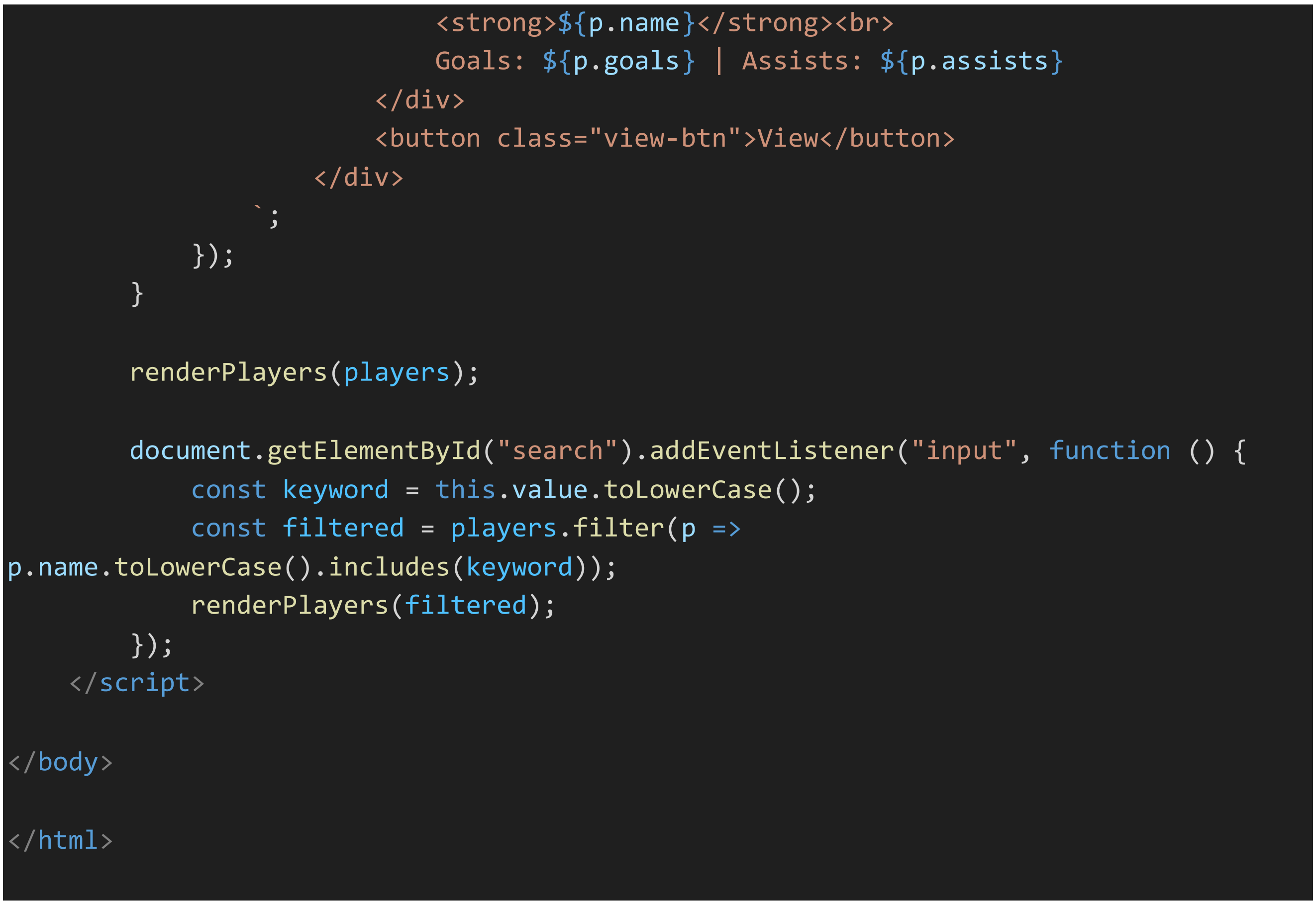
```
            text-align: center;
            color: var(--dark);
        }

        .banner h1 {
            font-size: 40px;
            margin: 0;
        }

        /* ---------------- DASHBOARD CARDS ---------------- */
        .dashboard {
            padding: 30px;
        }

        .section-title {
            font-size: 26px;
            margin-bottom: 15px;
        }

        .card-grid {
            display: grid;
            grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));
            gap: 20px;
        }

        .card {
            background: var(--white);
            padding: 20px;
            border-radius: 14px;
            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.15);
            transition: 0.3s;
        }

        .card:hover {
            transform: translateY(-4px);
        }

        .card h3 {
            margin: 0;
        }

        /* ---------------- PLAYER LIST ---------------- */
        .players-section {
            padding: 30px;
        }

        .player-card {
```



```
            background: var(--white);
            padding: 15px;
            border-radius: 12px;
            display: flex;
            justify-content: space-between;
            margin-bottom: 12px;
            align-items: center;
        }

        .view-btn {
            background: var(--purple);
            border: none;
            padding: 8px 14px;
            border-radius: 8px;
            cursor: pointer;
        }

        /* ---------------- FOOTER ---------------- */
        footer {
            text-align: center;
            padding: 20px;
            background: var(--purple);
            color: var(--dark);
            margin-top: 40px;
        }
    </style>
</head>

<body>

    <nav>
        <div class="logo">TeamStats</div>
        <input type="text" id="search" placeholder="Search players...">
    </nav>

    <section class="banner">
        <h1>Track. Analyze. Improve.</h1>
        <p>Your complete player performance management system</p>
    </section>

    <section class="dashboard">
        <div class="section-title">Team Dashboard</div>

        <div class="card-grid">
            <div class="card">
                <h3>Total Players</h3>
                <p id="totalPlayers">0</p>
```



```
            </div>

            <div class="card">
                <h3>Training Sessions</h3>
                <p>32 Completed</p>
            </div>

            <div class="card">
                <h3>Recent Match Result</h3>
                <p>Win vs Falcons (2 – 1)</p>
            </div>

            <div class="card">
                <h3>Top Performer</h3>
                <p id="topPlayer">Loading...</p>
            </div>
        </div>
    </section>

    <section class="players-section">
        <div class="section-title">Player List</div>
        <div id="playerList"></div>
    </section>

    <footer>© 2025 TeamStats. All rights reserved.</footer>

    <script>
        /* ---------------- PLAYER DATA ---------------- */
        const players = [
            { name: "Aarav Sharma", goals: 12, assists: 4 },
            { name: "Rohan Verma", goals: 5, assists: 10 },
            { name: "Krish Patel", goals: 8, assists: 6 },
            { name: "Reyansh Iyer", goals: 15, assists: 2 },
        ];

        document.getElementById("totalPlayers").innerText = players.length;

        let top = players.reduce((max, p) => p.goals > max.goals ? p : max);
        document.getElementById("topPlayer").innerText = top.name;

        function renderPlayers(list) {
            const container = document.getElementById("playerList");
            container.innerHTML = "";
            list.forEach(p => {
                container.innerHTML += `
                    <div class="player-card">
                        <div>
```



```
                            <strong>${p.name}</strong><br>
                            Goals: ${p.goals} | Assists: ${p.assists}
                        </div>
                        <button class="view-btn">View</button>
                    </div>
                `;
            });
        }

        renderPlayers(players);

        document.getElementById("search").addEventListener("input", function () {
            const keyword = this.value.toLowerCase();
            const filtered = players.filter(p =>
p.name.toLowerCase().includes(keyword));
            renderPlayers(filtered);
        });
    </script>

</body>

</html>
```

2. Concepts Applied

2.1 HTML Structure and Information Architecture

The website uses semantic HTML to maintain organized content flow:

- `<nav>` houses the application title and search bar
- `<section>` elements define dashboard, banner, and player listing
- `<footer>` closes the application layout

The navigation bar code:

```
<nav>
    <div class="logo">TeamStats</div>
    <input type="text" id="search" placeholder="Search
players...">
```

```
</nav>
```

This structure ensures clear hierarchy and optimal usability for coaches.

2.2 CSS Styling and Pastel Theme System

A soothing pastel color palette is implemented using CSS variables:

```
:root {
    --pink: #f7c6d9;
    --purple: #d6c2ff;
    --yellow: #fef3b7;
}
```

These variables:

- Maintain color consistency
- Support theme scalability
- Improve readability and visual identity

The dashboard cards use:

```
.card {
    background: var(--white);
    padding: 20px;
    border-radius: 14px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.15);
}
```

This creates a modern, soft shadow interface appropriate for analytics dashboards.

2.3 Responsive Grid Layout

The card layout utilizes **CSS Grid** to support automatic resizing:

```
grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));
```

This provides:

- Seamless mobile responsiveness
- Automatic column adjustment
- Cleaner code without media queries

Players and team metrics remain readable, regardless of device width.

2.4 JavaScript Logic and Dynamic Rendering

Player Data Structure

```
const players = [
    { name: "Aarav Sharma", goals: 12, assists: 4 },
    ...
];
```

JavaScript arrays store player stats, simplifying updates and filtering.

Dashboard Logic

```
let top = players.reduce((max, p) => p.goals > max.goals ? p : max);
document.getElementById("topPlayer").innerText = top.name;
```

Concepts used:

- `.reduce()` for performance evaluation
- DOM updating for dashboard metrics

Player List Rendering

```
function renderPlayers(list) {
    container.innerHTML = "";
    list.forEach(p => {...});
}
```

This shows:

- DOM manipulation
- Template literal rendering

Search Functionality

```
const filtered = players.filter(p =>
    p.name.toLowerCase().includes(keyword)
);
```

This enables:

- Real-time player search
- Partial and case-insensitive matching

2.5 UI/UX Principles

- **Soft pastel palette** to reduce visual strain
- **Clear visual hierarchy** with titles, metrics, and sections
- **Consistent spacing, rounded corners, and shadows** for modern aesthetics

- **Minimalistic data layout** for quick coaching decisions

3. Features Implemented

3.1 Player Statistics Display

Each player card displays:

- Name
- Goals
- Assists
- Quick “View” button (placeholder for future details tab)

This helps coaches get instant performance snapshots.

3.2 Interactive Search System

Coaches can search players instantly through the search bar. The JS filter function updates the UI in real time.

3.3 Dashboard Overview

Includes:

- Total players
- Training sessions completed
- Recent match results
- Top performer

This mirrors real sports dashboards used by training academies.

3.4 Fully Responsive Layout

Automatic resizing ensures coaches can access TeamStats on:

- Laptops
- Tablets
- Smartphones

CSS Grid simplifies responsiveness.

3.5 Pastel Theme UI

The unique visual theme aligns with calm analytics visual design, balancing clarity with aesthetics.

4. Challenges Faced & Solutions

4.1 Challenge: Creating Clean Dashboard Layouts

Issue: Data visualization requires clear grouping.
Solution:
Used grid cards with spacing, shadows, and rounded corners for clarity.

4.2 Challenge: Handling Multiple Player Data Frontend-Only

Issue: No backend meant storing all data in the browser.
Solution:
Used structured JS arrays and functions to simulate real database-like behavior.

4.3 Challenge: Making Search Function Fast and Efficient

Issue: Filtering can become slow with many players.
Solution:
Used efficient `.filter()` and cleaned input using `.toLowerCase()`.

4.4 Challenge: Maintaining Color Harmony

Pastel colors can clash if not balanced.
Solution:
Used soft whites and neutral text to create visual breathing room.

4.5 Challenge: Ensuring Mobile Responsiveness

Grid breaks on narrow screens.
Solution:
`auto-fit` + `minmax()` allows resizing without custom queries.

Conclusion

TeamStats successfully integrates modern frontend technologies to create an intuitive sports analytics platform. Through responsive design, clean UI, and interactive JavaScript features, the platform delivers practical utility for coaches and players seeking performance insights. The project demonstrates essential skills in web development, data visualization logic, and interface usability.