# Sports Buddy

## Project Overview

Sports Buddy is a web application designed to manage and promote sports events. It provides functionalities for user registration, login, and an admin dashboard to manage sports events, categories, cities, and areas.

## Technologies Used

- HTML

- CSS

- JavaScript

- Firebase (Authentication, Firestore, Hosting)

## Project Setup

### Prerequisites

- Node.js and npm installed

- Firebase CLI installed (`npm install -g firebase-tools`)

- A Firebase project set up in the Firebase Console

### Project Initialization

1. Clone the repository:

```bash

git clone https://github.com/your-username/sports-buddy.git

cd sports-buddy

```

2. Install project dependencies (if any):

```bash

npm install

3. Initialize Firebase in your project directory:

```bash

firebase init

- Select Hosting and Firestore (or Realtime Database).

- Set up the Firebase configuration in your project (follow the prompts).

4. Replace Firebase configuration details in `scripts.js`:

```javascript

const firebaseConfig = {

apiKey: "YOUR\_API\_KEY",

authDomain: "YOUR\_AUTH\_DOMAIN",

databaseURL: "YOUR\_DATABASE\_URL",

projectId: "YOUR\_PROJECT\_ID",

storageBucket: "YOUR\_STORAGE\_BUCKET",

messagingSenderId: "YOUR\_MESSAGING\_SENDER\_ID",

appId: "YOUR\_APP\_ID"

};

## Project Structure

sports-buddy/

├── .firebaserc

├── firebase.json

├── public/

│ ├── index.html

│ ├── styles.css

│ └── scripts.js

├── README.md

└── .gitignore

```

## Features

- \*\*User Registration\*\*: Allows users to register with their email and password.

- \*\*User Login\*\*: Allows users to log in with their registered email and password.

- \*\*Admin Dashboard\*\*: Provides admin functionalities to manage sports events, categories, cities, and areas.

## Usage

### Running the Project

1. Deploy the project to Firebase Hosting:

```bash

firebase deploy

```

2. Open the deployed URL in your browser to access the application.

### Admin Functionalities

- \*\*Login as Admin\*\*: Admin users can log in to access the dashboard.

- \*\*Manage Sports Events\*\*: Admins can add, update, or delete sports events.

- \*\*Manage Categories\*\*: Admins can add, update, or delete sports categories.

- \*\*Manage Cities and Areas\*\*: Admins can add, update, or delete cities and areas.

## Code Structure

### HTML (`index.html`)

Contains the structure of the web application including registration and login forms.

### CSS (`styles.css`)

Contains the styling for the web application.

### JavaScript (`scripts.js`)

Contains the client-side logic for handling user registration, login, and admin functionalities using Firebase.

## Firebase Integration

### Authentication

- Used for user registration and login.

### Firestore (or Realtime Database)

- Used for storing and managing data related to sports events, categories, cities, and areas.

### Hosting

- Used to host the web application.

## Logging

- Use `console.log()` and `console.error()` for client-side logging.

- For server-side logging, use Firebase Cloud Functions if needed.

## Coding Standards

- Follow ES6+ syntax and best practices.

- Use meaningful variable and function names.

- Write modular and reusable code.

- Ensure code is safe, testable, maintainable, and portable.