1 Start Objective

Our start objective covers what we want to complete in the first part of the semester for the SportHub project. SportHub will be a multi–sport scores and favorites web application. Users will be able to browse upcoming games for different sports and mark games as favorites so they can quickly find them later.

During this phase we focus on planning and setting up the base framework for both the front end and the back end. We decide our main technologies, create the GitHub repository and folders, and write the first versions of our documents (SDD, SRS, user manual, design spec, and this snapshot objectives document). Tasks are divided so some team members focus more on the frontend while others focus on backend and infrastructure.

1.1 Front-End

React: Main UI framework. React lets us build our pages from small, reusable components and efficiently update only the parts of the screen that change.

Vite (or similar build tool): Used for local development and bundling the React app for production. It gives us a fast dev server and builds our JavaScript into optimized files.

UI Library (Bootstrap / Material UI): Provides ready–made components like buttons, navbars, and cards so we can create a clean, responsive layout for game lists and the favorites page.

1.2 Back-End

Node.js with Express: Backend framework used to build a REST API. The API will return data for games and favorites and handle requests from the frontend.

PostgreSQL: Relational database to store games (teams, sport, date, time) and which games each user has marked as a favorite.

Docker and docker-compose: Used to run the frontend, backend, and database as separate services. This will make it easier to start the whole project with a single command later in development.

2 Checkpoint 1

With the basic plan and structure in place, Checkpoint 1 focuses on getting the core viewing experience of SportHub running. By this snapshot we want a simple but functional way for users to see upcoming games.

Goals

- Create a frontend page that lists upcoming games with sport, teams, date, and time.
- Add basic filters (for example by sport and/or date).

- Implement a backend endpoint that returns a list of games (using mock data or an initial database table).
- Connect the frontend to the backend API so the list of games comes from our server instead of hard–coded values.
- Update SDD, SRS, design spec, and user manual to describe the "view games" feature and how the data flows.

3 Checkpoint 2

Checkpoint 2 adds our second major feature: the ability for users to keep track of their favorite games. This makes SportHub more personal and useful.

Goals

- Allow the user to mark or unmark games as favorites from the main games list.
- Create a "My Favorites" page that shows only the games the user has favorited.
- Extend the backend and database to store favorites for each user (or user session).
- Ensure that updates to favorites on the frontend are sent to the backend and saved correctly.
- Update all documents (SDD, SRS, design spec, user manual) and add or refine tests to cover the favorites feature.

4 Final Objective (Due Date Checkpoint)

The final snapshot is about completing SportHub, polishing the experience, and documenting both what we finished and ideas for future work.

Goals

- Refine the UI for all pages (games list, favorites, and any admin pages) so the app is easy to navigate and visually consistent.
- Complete basic admin tools so project members can add, edit, or remove games without manually modifying the database.
- Fix important bugs discovered during testing and make sure the main user flows work reliably.
- Finalize the Docker configuration so the whole system can be started with one command and is ready to be graded or demoed.

- Bring the SDD, SRS, user manual, design spec, and workflow diagram up to date with the final implementation.
- Write a short reflection on what was completed, what was not finished, and possible future work (more sports, real live-score APIs, authentication, mobile layout improvements, etc.).