Sports Equipment Management System Project Report

1. What Our Project Is About:

Our project is a **Sports Equipment Management System** made in C.
It runs completely in the terminal and helps a college store information about:

- Sports equipment
- Students who borrow equipment
- Login system for admin and students

We used **file handling** to save all data permanently so nothing is lost when the program closes.

---

2. What Features Our Project Has:

Equipment Management (Admin)

- Add new sports items
- View all equipment
- Update equipment details
- Delete equipment
- Search equipment by ID

All equipment is saved in **equipment.txt**.

Borrower Management (Students + Admin)

- Borrow equipment
- Return equipment
- View list of borrowers
- Search borrower by ID

Stored in **borrowers.txt**.
Quantity updates automatically when items are borrowed or returned.

Login System

- Student can sign up and sign in
- Admin has a passkey + credential check
- Ensures only authorized people access admin functions

Main Menu (Complete Navigation)

We combined all modules into one menu where:

- Admin can manage everything
- Students can search, borrow, and return equipment

3. How We Built This :

Step 1: Creating the Structures

We made all the main data types in data_structures.h:

- Equipment → stores id, name, price, manufacturer, quantity
- Borrower → stores borrower name, equipment they took, dates, return status

These structs are shared across all modules.

Step 2: Making the Equipment Module

We created functions that read/write equipment details into a file using fopen, fprintf, and fscanf.
We used a temporary file method for updating and deleting records safely.

Step 3: Creating the Borrower Module

This part connects borrowing with equipment quantity.
When a student borrows:

- Quantity of equipment decreases
  When they return:
- Quantity increases back

Borrower details are saved in borrowers.txt.

Step 4: Adding Login Features

We created signup and login for students.
Admin login asks for a passkey "passkey" and then checks credentials from a file.

Step 5: Designing the Main Menu

Once all modules were ready, we connected everything inside main.

---

4. Team Contribution:

Sanyam(BA2025043):

- Built **borrower module**
- Helped in creating data_structures.h
- Worked on file handling and equipment availability logic

Bharat(BE2025007):

- Built the **equipment module**
- Helped in making data_structures.h
- Wrote all functions for add/view/update/delete/search equipment

---

Aditya(BE2025007):

- Built the **main.c** (main menu + integration)
- Worked on the **login system**
- Debugged and connected all parts together

---

5. Explanation of All Functions:

Equipment Functions

- **addEquipment()** – Takes equipment details from admin and saves them into equipment.txt.
- **viewEquipment()** – Reads and prints every equipment item.
- **searchEquipment()** – Looks up equipment by ID.
- **updateEquipment()** – Lets admin modify equipment details.
- **deleteEquipment()** – Removes an equipment item from the file.

Borrower Functions

- **addBorrower()** – Saves a new borrow record and decreases equipment quantity.
- **viewBorrowers()** – Lists all borrowers and their borrowed items.
- **searchBorrower()** – Searches borrower details by ID.
- **returnBorrowedItem()** – Marks an item as returned and increases equipment quantity.
- **isEquipmentAvailable()** – Checks if the equipment has quantity > 0.
- **updateEquipmentQuantity()** – Updates stock when items are borrowed/returned.

Login Functions

- **sign_up()** – Creates a new student account.
- **sign_in()** – Lets student log in by checking username and password.
- **admin()** – Checks admin's passkey and credentials.

---

main.c Logic

- Handles role selection (Admin/Student)
- Shows menus and calls the correct functions
- Connects all modules together
