



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IITI Student Community Application

CS 416/616: SERVICE-ORIENTED SYSTEMS

Name	Roll Number
KRISH AGRAWAL	210001034
RAHUL	210001056
VIKAS MAURYA	210001076
AGRIMA BUNDELA	210002009
ADARSH MISHRA	2402101004
AKSHAT JAIN	2430201002

Course Instructor: PROF. ABHISHEK SRIVASTAVA

User Manual (Admin)

APRIL 25, 2025

1 Introduction

This manual is designed for authorized Club Representatives and System Administrators who will be using the web-based Admin Dashboard for the IITI Student Community Application. The dashboard serves as the primary interface for managing the content displayed within the student mobile application, including events, merchandise, and club information.

2 Accessing the Dashboard

1. Navigate to the designated Admin Dashboard URL using a web browser (The specific URL should be provided separately).
2. Log in using your authorized IIT Indore credentials.
3. Access is secured using Firebase Authentication, ensuring only authorized personnel can manage the application's content.

3 Dashboard Overview

The Admin Dashboard, built using React.js, provides distinct sections for managing the core components of the student application:

- **Events Management:** Create, update, delete, and manage registrations for campus events. See Figure 1.
- **Merchandise Management:** Add, edit, remove, and manage inventory for products sold by clubs. See Figure 2, 3.
- **Clubs Management:** Register new clubs, update existing club information, and manage club visibility. See Figure 4.

The dashboard utilizes real-time data synchronization with Firebase Firestore, meaning changes made should reflect relatively quickly both on the dashboard and in the student app.

4 Features Guide

4.1 Events Management

See Figure 1.

4.1.1 Creating Events

1. Navigate to the "Events" section within the dashboard.
2. Click the "Create Event" or a similarly labelled button.
3. A form will appear. Fill in all required event details meticulously:
 - Event Name (Clear and concise)
 - Date and Time (Specify start and end times)
 - Location using map
 - Description (Detailed information about the event)

- Organizing Club/Body
4. Upload relevant event posters or images. The interface may show upload progress. Images are stored securely in Firebase Storage. Ensure images are optimized for web/mobile viewing if possible.
 5. Review all entered information for accuracy.
 6. Save the event. The data will be saved to the Firebase Firestore database and become visible to students in the mobile app shortly after.

4.1.2 Editing Events

1. Locate the event you wish to modify in the list or table within the "Events" section.
2. Click the "Edit" button or icon associated with that event.
3. The event creation form will appear, pre-filled with the existing details.
4. Modify any necessary fields (name, date, time, location, description, images).
5. Save the changes. The updates will be applied to the Firestore record using the 'updateDoc()' function.

4.1.3 Deleting Events

1. Find the event you need to remove in the "Events" list.
2. Click the "Delete" button or icon for that specific event.
3. A confirmation prompt will likely appear to prevent accidental deletion. Confirm that you wish to proceed.
4. The event record will be permanently removed from the Firestore database using the 'deleteDoc()' function. This action cannot be undone.

4.2 Merchandise Management

See Figures 2, 3.

4.2.1 Adding Products

1. Navigate to the "Merchandise" section of the dashboard.
2. Click the "Add Product" or similar button.
3. Fill in the product details accurately in the provided form:
 - Product Name
 - Detailed Description
 - Price
 - Club Offering the Product
 - Available Sizes (e.g., S, M, L, XL)
 - Available Colors (e.g., White, Black, Blue)

- Purchase Deadline (if the item is available for a limited time)
 - Availability Status
4. Upload high-quality product images. The system supports uploading multiple images per product, potentially handled using ‘Promise.all()’ for efficiency. Images are stored in Firebase Storage.
 5. Review all details.
 6. Save the product. The information is stored in Firestore and the item will become visible in the Merchandise section of the student app.

4.2.2 Editing Products

1. Find the product you want to update in the merchandise list.
2. Click the ”Edit” button associated with it.
3. Modify product details, pricing, inventory levels, available sizes/colors, descriptions, or upload new/replacement images as needed.
4. Save the changes. Firestore’s ‘updateDoc()’ function will be used to update the product record.

4.2.3 Deleting Products

1. Locate the product to be removed.
2. Click the ”Delete” button for that product.
3. Confirm the deletion when prompted.
4. The product will be permanently removed from Firestore using ‘deleteDoc()’. Ensure associated images in Storage are also handled if necessary (manual deletion might be required depending on implementation).

4.3 Clubs Management

See Figure 4.

4.3.1 Registering New Clubs

1. Go to the ”Clubs” section of the dashboard.
2. Click the ”Add Club” or similar button.
3. Enter the required details for the new club:
 - Club Name
 - Club Description/Mandate
4. Upload the official club logo and any relevant banner images. These images will be stored in Firebase Storage.
5. Review the information.
6. Save the club. The club profile will be created in Firestore and listed in the student app.

4.3.2 Editing Clubs

1. Find the club you need to modify in the "Clubs" list.
2. Click the "Edit" button for that club.
3. Update the club's details (name, description, contact) or upload new logo/banner images as required.
4. Save the changes. The Firestore record will be updated via `updateDoc()`.

4.3.3 Deleting Clubs

1. Locate the club to be removed from the system.
2. Click the "Delete" button for that club.
3. Confirm the deletion when prompted.
4. The club record will be permanently deleted from Firestore using `deleteDoc()`. Associated events or merchandise might need separate handling depending on data relationships.

5 Technical Notes & Troubleshooting

- **Real-time Updates:** The dashboard leverages Firebase Firestore's `onSnapshot` listener for real-time data updates. Most changes should appear without needing a manual page refresh.
- **Error Handling:** Firebase operations (create, update, delete) are enclosed in `'try-catch'` blocks. If an operation fails, error messages or alerts should appear on the dashboard providing feedback. Note any specific error messages for troubleshooting.
- **CORS Configuration:** For browser-based image uploads directly to Firebase Storage to work correctly, the Firebase project's Cross-Origin Resource Sharing (CORS) policy must be configured to allow requests from the Admin Dashboard's domain. If image uploads consistently fail, incorrect CORS settings are a likely cause. This is typically configured using the `'gsutil'` command-line tool (e.g., `'gsutil cors set cors-config.json gs://your-bucket-name'`).
- **Security Rules:** Firebase Firestore and Storage security rules are implemented to restrict write access (creating, editing, deleting data) primarily to authenticated administrators. These rules are crucial for data integrity and security.
- **Performance:** For optimal performance, consider optimizing images before uploading. The backend might employ techniques like batched writes for bulk operations if implemented.

The screenshot shows a web browser at localhost:4173/#/events. The page is titled "College Admin Panel" with a "Logout" button. A sidebar on the left contains links for Dashboard, Events, Merchandise, and Clubs. The main content area is titled "Event Management" and features a "Create New Event" form. The form includes fields for Event Name, Location, Event Poster (with a "Choose File" button and "No file chosen" text), Latitude, Longitude, and a Description text area. A date picker is set to mm/dd/yyyy, --:-- --. A blue "Create Event" button is at the bottom of the form.

Figure 1: Admin Dashboard - Events Management Section.

The screenshot shows a web browser at localhost:4173/#/merchandise. The page is titled "College Admin Panel" with a "Logout" button. A sidebar on the left contains links for Dashboard, Events, Merchandise, and Clubs. The main content area is titled "Merchandise Management" and features an "Add New Product" form. The form includes fields for Product Name, Type (with a dropdown menu showing "T-shirt, Hoodie, etc."), Price (₹), Available Sizes (S, M, L, XL (comma separated)), Available Colors (Red, Blue, Green (comma separated)), Last Date To Purchase (mm/dd/yyyy), Club ID, Club Name, Customization Fields, and a Description text area. A "Product Images" section at the bottom has a "Choose Files" button and "No file chosen" text. A blue "Add Product" button is at the bottom of the form.

Figure 2: Admin Dashboard - Adding a New Merchandise Product.

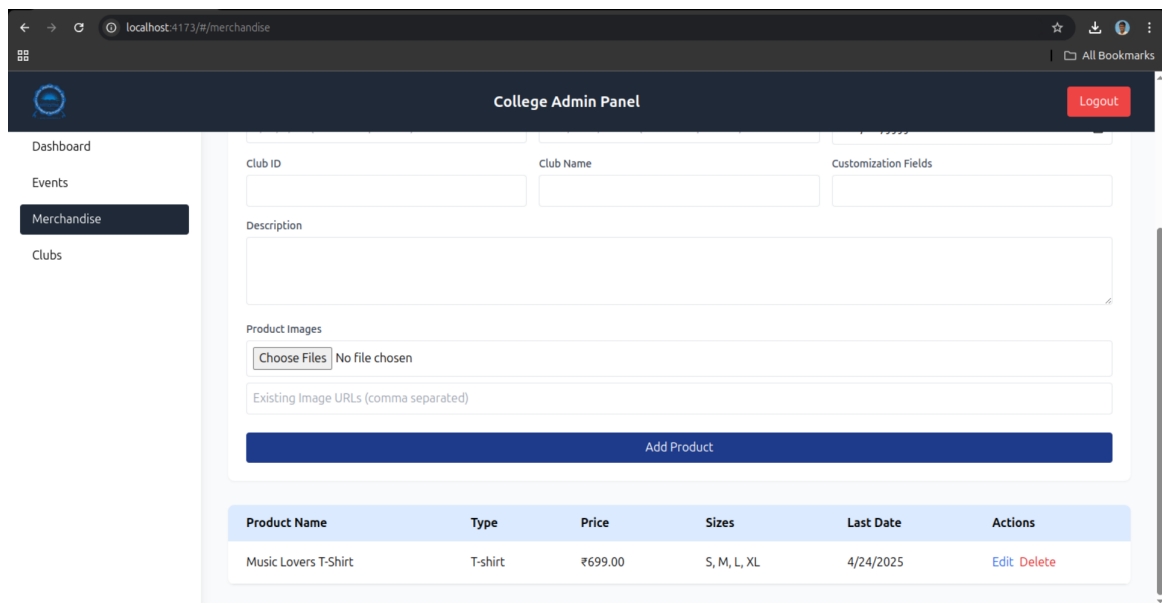


Figure 3: Admin Dashboard - Viewing Merchandise List.

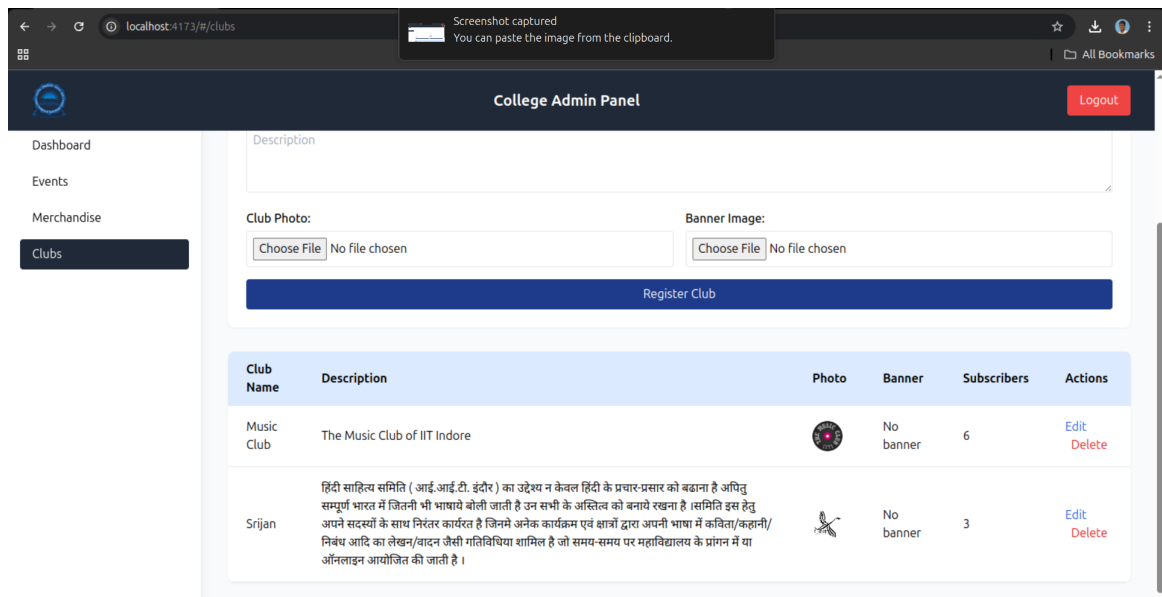


Figure 4: Admin Dashboard - Clubs Management Section.