BASAVARAJESWARI GROUP OF INSTITUTIONS

# Ballari Institute of Technology & Management

AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA,
BELAGAVI 590018

**INTERNSHIP**

**Report On**

## Locker room management system

Submitted in partial fulfilment of the requirements for the award of degree of

**Bachelor of Engineering**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**(DATA SCIENCE)**

**Submitted by**

**Keerthi**

**3BR22CD044**

**Internship Carried Out By**

**EZ TRAININGS & TECHNOLOGIES PVT.LTD**

**HYDERABAD**

| **Internal Guide** | **External Guide** |
|---|---|
| Mrs. V P Anushya Asst. Prof | Technical Trainer: |
| Mrs.Parvathi | Mr.Vishal kumar |

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)**

**"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road, Alipur,
Ballar1-583 104 (Karnataka) (India) Ph: 08392 – 237100 /
237190, Fax: 08392 – 237197**

**2024-2025**

**BASAVARAJESWARI GROUP OF INSTITUTIONS**

# BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

**Autonomous institute under VISVESVARAYA TECHNOLOGICAL UNIVERSITYJNANA SANGAMA, BELAGAVI 590018**

NACC Accredited Institution*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)**

**"JnanaGangotri"Campus,No.873/2,Ballari-HospetRoad,Allipur,** Ballari
**583 104 (Karnataka)(India)**
**Ph: 08392 – 237100 / 237190, Fax: 08392 –237197**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-DATA SCIENCE

# CERTIFICATE

This is to certify that the Internship entitled **" EVENT MANAGEMENT PLATFORM "** has been successfully completed by KEERTHI bearing USN **3BR22CD044** a bonafide student of Ballari Institute of Technology and Management, Ballari.

For the partial fulfilment of the requirements for the

**Bachelor's Degree in Computer Science and Engineering-Data science**

of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi during the academic year 2024-2025.

**Signature of Internship Co-ordinator**

**Signature of HOD**

# DECLARATION

I, KEERTHI second year student of Computer science –Data science and Engineering, Ballari Institute of Technology, Ballari, declare that Internship entitled **EVENT MANAGEMENT PLATFORM** is a part of Internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT.LTD ,Hyderabad** at "**BITM,BALLARI".** This report is submitted in partial fulfilment of the requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering-Data science of the Visvesvaraya Technological University, Belagavi.

**Date : 28.09.2024** **Signature of the Student**

**Place : Ballari**

# ACKNOWLEDGEMENT

The satisfactions that a company the successful completion of my internship on " Event Management Platform " would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to our respective coordinator **"V.PAnushya(Asst.prof,CSE-DS) , parvathi(Asst.prof,CSE-DS)"** for his noble gesture, support co-ordination and valuable suggestions given to me in the completion of Internship.

I also thank Dr.Aradhana**,** H.O.D. Department of **Computer science and engineering-Data science**

for extending all his valuable support and encouragement.

# Table of Contents

**Introduction**:

A Locker Room Management System is designed to streamline the management of lockers in facilities such as gyms, sports complexes, schools, and recreational centers. This system typically includes features like:

1. **Locker Assignment**: Automates the allocation of lockers to users, ensuring efficient use of space.
2. **User Registration**: Allows users to create profiles and manage their locker usage
3. **Access Control**: Implements security measures, such as key cards or digital codes, to restrict locker access.
4. **Inventory Tracking**: Monitors the status of each locker, including occupancy and maintenance needs.
5. **Notifications**: Sends alerts for locker expirations, renewals, or maintenance updates.

## Problem statement:

CRUD: Locker assignments.

• manage_locker_assignments(team_id): Manage locker assignments for sports teams.

• monitor_locker_room_access(access_data): Monitor access to ensure security and privacy.
Code: To create a Python OOP-based locker room management system that handles locker assignments and monitors locker room access, we'll define classes and methods for CRUD operations on locker assignments, and classes for monitoring locker room access. We'll apply data structures like dictionaries for efficient data manipulation.

## Real-Life Example: Locker Room Management System for a Sports Team

Imagine a sports team facility where multiple teams share locker rooms for practice and games. The **Locker Room Management System** is essential for efficiently managing locker assignments, ensuring security, and monitoring access to the locker room area.

### 1. CRUD: Locker Assignments

The system allows facility managers to create, read, update, and delete locker assignments for various sports teams.

**Example:**

- **Create:** A facility manager assigns lockers to a new basketball team that just joined the league. They create locker assignments for each player, ensuring that each locker is uniquely identified.
- **Read:** A coach wants to check the current locker assignments for the football team before a game. They retrieve the list of assignments to ensure each player has their designated locker ready for use.
- **Update:** After a player is transferred to another team, the facility manager updates the locker assignment to reflect the change, ensuring the locker can be reassigned to a new player joining the team.
- **Delete:** Once the baseball season ends, the manager deletes the locker assignments for the baseball team to prepare for the upcoming soccer season.

### 2. manage_locker_assignments(team_id)

The system enables users to manage locker assignments based on team IDs, allowing for quick modifications and updates.

**Example:**

- The manager receives a request to change locker assignments for the hockey team due to a team member's injury. They use the team ID to access the current locker assignments and reassign lockers as necessary, ensuring the injured player has easy access to their gear without disrupting the rest of the team.

### 3. monitor_locker_room_access(access_data)

The system tracks who accesses the locker room and when, helping to maintain security and privacy.

**Example:**

- The facility is equipped with a key card access system. Whenever a player or staff member enters or exits the locker room, their access data is logged. The manager reviews access logs to ensure that only authorized personnel enter the locker room, helping to prevent unauthorized access or potential security breach

**Methodology**:

```
class Locker:
    def __init__(self, locker_number, status="Available"):
        self.locker_number = locker_number
        self.status = status
        self.user = None

class LockerRoom:
    def __init__(self):
        self.lockers = {}

    def add_locker(self, locker_number):
        self.lockers[locker_number] = Locker(locker_number)

    def remove_locker(self, locker_number):
        if locker_number in self.lockers:
            del self.lockers[locker_number]
        else:
            print("Locker not found.")

    def rent_locker(self, locker_number, user):
        if locker_number in self.lockers and self.lockers[locker_number].status == "Available":
            self.lockers[locker_number].status = "Occupied"
            self.lockers[locker_number].user = user
            print(f"Locker {locker_number} rented to {user}.")
        else:
            print("Locker not available.")

    def return_locker(self, locker_number):
        if locker_number in self.lockers and self.lockers[locker_number].status == "Occupied":
            self.lockers[locker_number].status = "Available"
            self.lockers[locker_number].user = None
            print(f"Locker {locker_number} returned.")
        else:
            print("Locker not occupied.")

    def display_lockers(self):
        print("Locker Room Status:")
        for locker_number, locker in self.lockers.items():
            print(f"Locker {locker_number}: {locker.status} ({locker.user})")


def main():
    locker_room = LockerRoom()

    while True:
        print("\nLocker Room Management System")
        print("1. Add Locker")
        print("2. Remove Locker")
        print("3. Rent Locker")
        print("4. Return Locker")
        print("5. Display Lockers")
        print("6. Exit")

        choice = input("Choose an option: ")

        if choice == "1":
            locker_number = input("Enter locker number: ")
            locker_room.add_locker(locker_number)
        elif choice == "2":
            locker_number = input("Enter locker number: ")
            locker_room.remove_locker(locker_number)
        elif choice == "3":
            locker_number = input("Enter locker number: ")
            user = input("Enter user name: ")
            locker_room.rent_locker(locker_number, user)
        elif choice == "4":
            locker_number = input("Enter locker number: ")
            locker_room.return_locker(locker_number)
        elif choice == "5":
            locker_room.display_lockers()
        elif choice == "6":
            break
        else:
            print("Invalid option. Please choose again.")

if __name__ == "__main__":
    main()
```

Output:

```
Locker Room Management System
1. Add Locker
2. Remove Locker
3. Rent Locker
4. Return Locker
5. Display Lockers
6. Exit
Choose an option: 1
Enter locker number: 01

Locker Room Management System
1. Add Locker
2. Remove Locker
3. Rent Locker
4. Return Locker
5. Display Lockers
6. Exit
Choose an option: 1
Enter locker number: 02

Locker Room Management System
```

```
1. Add Locker
2. Remove Locker
3. Rent Locker
4. Return Locker
5. Display Lockers
6. Exit
Choose an option: 3
Enter locker number: 01
Enter user name: Harry
Locker 01 rented to Harry.

Locker Room Management System
1. Add Locker
2. Remove Locker
3. Rent Locker
4. Return Locker
5. Display Lockers
6. Exit
Choose an option: 3
Enter locker number: 02
Enter user name: Esha
Locker 02 rented to Esha.

Locker Room Management System
1. Add Locker
2. Remove Locker
3. Rent Locker
4. Return Locker
5. Display Lockers
6. Exit
Choose an option: 4
Enter locker number: 02
Locker 02 returned.

Locker Room Management System
1. Add Locker
2. Remove Locker
3. Rent Locker
4. Return Locker
5. Display Lockers
6. Exit
Choose an option: 5
Locker Room Status:
Locker 01: Occupied (Harry)
Locker 02: Available (None)
```

## Conclusion

The **Locker Room Management System** offers a streamlined approach for sports teams and facility managers by:

- Efficiently managing locker assignments to ensure each player has a designated space for their belongings.
- Allowing for quick adjustments to assignments based on team changes or individual needs.
- Monitoring access to locker rooms to ensure security and maintain the privacy of team members.

This system enhances the overall management of locker facilities, helping sports teams maintain organization and security, ultimately contributing to a better team environment