Mini Project Report on

PESU Integrated Event Management System

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

Bachelor of Technology

in

Computer Science & Engineering

UE22CS351A – DBMS Project

Submitted by:

Suprith S	**PES2UG22CS600**
Siri N Shetty	**PES2UG22CS556**

Under the huidance of
Prof. Nivedita Kasturi
Assistant Professor
PES University

AUG – DEC 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

PESU Integrated Event Management System

is a bonafide work carried out by

Suprith S	**PES2UG22CS600**
Siri N Shetty	**PES2UG22CS556**

In partial fulfilment for the completion of fifth semester DBMS Project (UE22CS351A) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2024 – DEC. 2024. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature

Prof. Nivedita Kasturi

Assistant Professor

DECLARATION

We hereby declare that the DBMS Project **PESU Integrated Event Management System** has been carried out by us under the guidance of **Prof. Nivedita Kasturi, Assistant Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester **AUG – DEC 2024**.

Siri N Shetty **PES2UG22CS556**

Suprith S **PES2UG22CS600**

ABSTRACT

In a dynamic university environment with over 100+ clubs across campuses, events, workshops, and recruitment drives are frequent and integral to student life. However, managing the immense volume of student information for every event becomes highly inefficient when relying on Google Forms and Excel sheets. Asking students to repeatedly fill in the same details for every event is tedious and frustrating, while the reliance on random WhatsApp groups for event circulation often leaves many students uninformed due to inactivity or exclusion from these groups.

To address these challenges, we propose a centralized platform designed to streamline event and recruitment management. This platform enables club heads, acting as administrators, to create and manage events and recruitment drives, ensuring accessibility to all students. By leveraging relational databases, the platform eradicates repetitive data entry by automatically retrieving participant information from the database during registration. Students gain access to a user-friendly interface to browse and enroll in events effortlessly, ensuring no opportunity is missed. This solution not only enhances efficiency but also fosters a more inclusive and organized campus experience for all stakeholders.

TABLE OF CONTENTS

Introduction

The proposed PESU Integrated Event Management System aims to eliminate the repetitive collection of student information through multiple forms and manual processes by centralizing all data into a unified platform. Instead of using separate tools like Google Forms for club memberships, event registrations, and elective selection, this system streamlines the process, ensuring students need to provide their details only once. By integrating relationships between students, clubs, and events, the system reduces redundancy, automates workflows, and enhances efficiency, making it a practical and impactful project for improving administrative operations and student engagement.

The PESU Integrated Event Management System is designed for students, faculty, and administrators. It provides a secure and user-friendly interface for managing student information and club events, replacing the current reliance on disparate Google Forms.

Problem Definition with User Requirement Specifications

The purpose of the *PESU Integrated Event Management System* is to create a centralized, user-friendly platform that streamlines key aspects of student life and administrative processes at PES University. This system aims to replace the current dependency of unalike Google Forms with a database-driven application that enhances efficiency, improves data management, and provides a better user experience for students, faculty, and administrators. It is a new, standalone system that will integrate with existing university databases and systems where necessary.

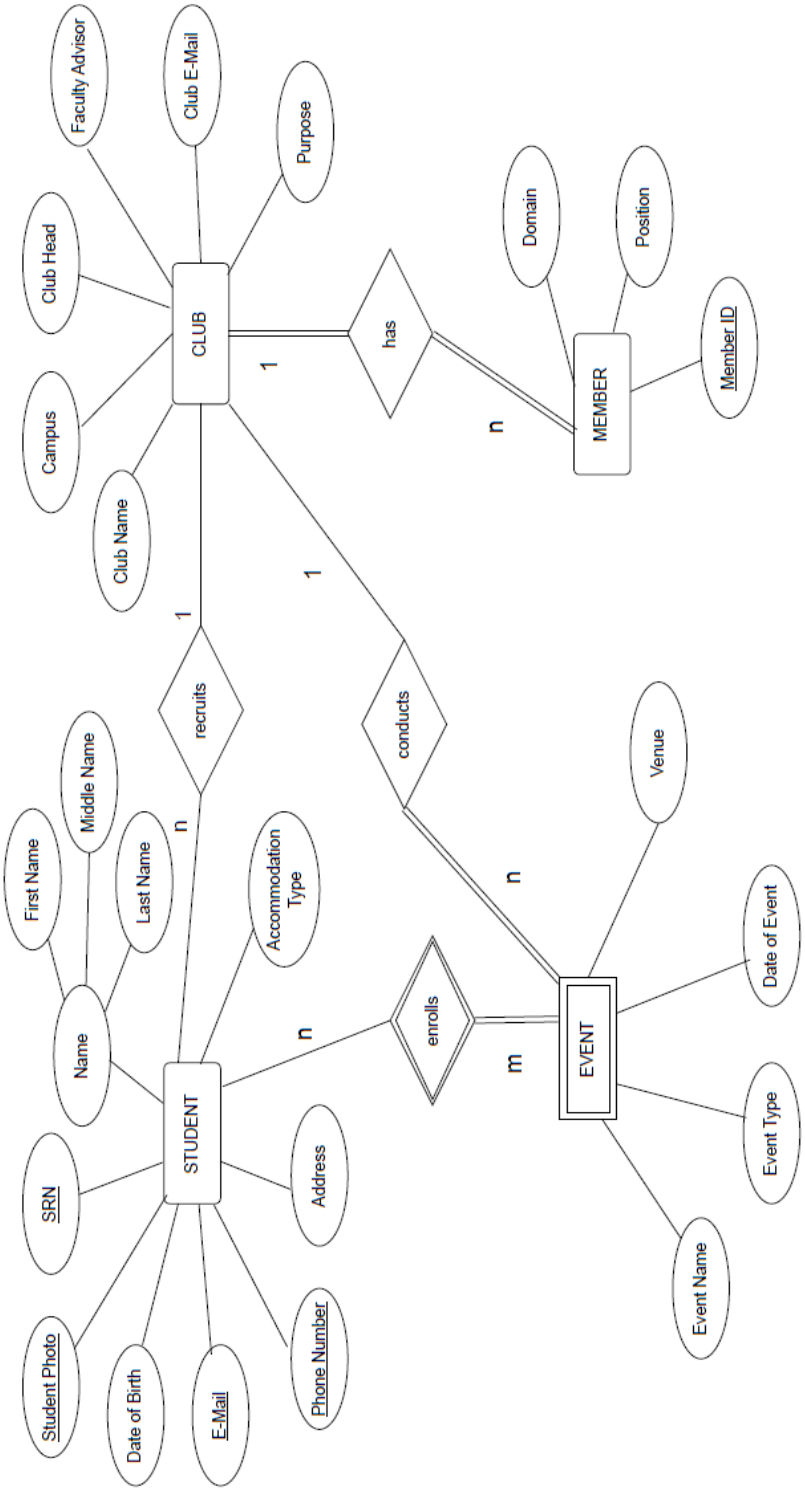
The proposed system aims to streamline club management by addressing the specific needs of different users, including club heads, members, and participants. Club heads require a secure platform to manage events, recruitments, and club-related information while having the flexibility to add custom questions for participant registration. Participants, on the other hand, require a simple interface to register for events and provide required details seamlessly.

The system eliminates repetitive data collection processes, ensuring a user-friendly, centralized, and efficient solution for managing clubs and events effectively.

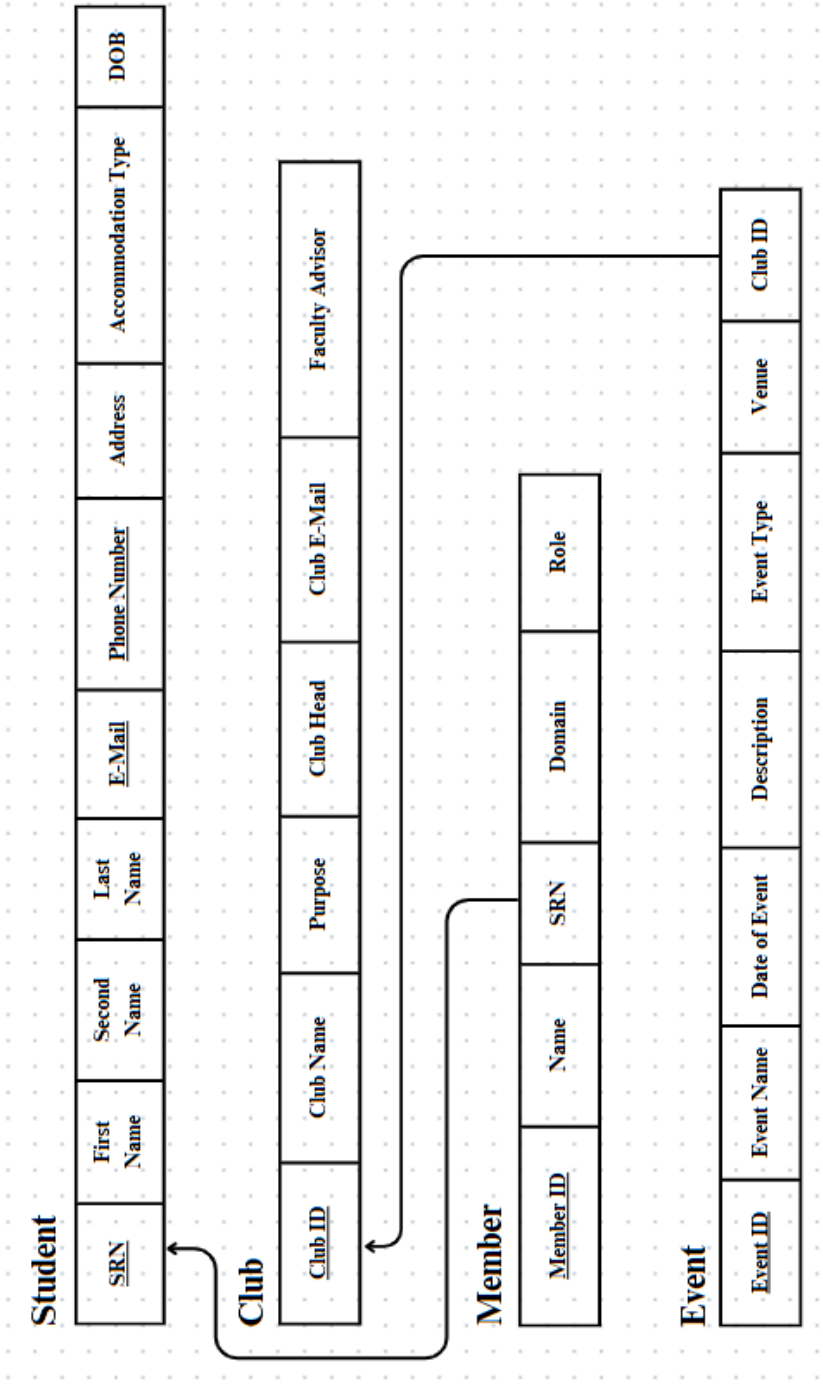
List of Softwares/Tools/Programming Languages Used

- **React JS- Frontend**
- **Tailwind CSS**
- **Lucide-React- For web designs**
- **Framer-Motion- For transition designs**
- **Flask- Backend**
- **Sqlite3- MySqlite library**
- **CSV library- to save sql table into csv format**

ER Model


Faculty Advisor
Club E-Mail
Purpose
Club Head
CLUB
has
Domain
Position
Campus
Club Name
1
n
MEMBER
Member ID
1
1
recruits
conducts
n
n
First Name
Middle Name
Last Name
Name
Accommodation Type
Venue
Date of Event
enrolls
n
m
EVENT
SRN
STUDENT
Address
Event Type
Event Name
Student Photo
Date of Birth
E-Mail
Phone Number

ER To Relational Mapping



DDL Statements

```python
@app.route('/api/recruitment/create', methods=['POST'])
def create_recruitment_form():
    data = request.json
    club_id = data.get('clubId')
    club_name = data.get('clubName')
    fields = data.get('fields')

    try:
        conn = get_db_connection()
        cursor = conn.cursor()

        # Remove existing recruitment form if it exists
        table_name = f"{club_name.lower().replace(' ', '_')}_recruitments"
        cursor.execute(f"DROP TABLE IF EXISTS {table_name}")

        # Create dynamic table based on fields
        create_table_query = f"CREATE TABLE {table_name} (id INTEGER PRIMARY KEY AUTOINCREMENT"
        for field in fields:
            field_type = 'TEXT'
            if field['type'] == 'number':
                field_type = 'REAL'
            elif field['type'] == 'email':
                field_type = 'TEXT'

            create_table_query += f", {field['label'].lower().replace(' ', '_')} {field_type} {'NOT NULL' if field['required'] else ''}"

        create_table_query += ")"

        cursor.execute(create_table_query)
        conn.commit()
        conn.close()
```

```python
@app.route('/api/events/close_registrations/<int:event_id>', methods=['POST'])
def close_event_registrations(event_id):
    try:
        conn = get_db_connection()
        cursor = conn.cursor()

        # Get event name for table name generation
        cursor.execute('SELECT event_name FROM events WHERE event_id = ?', (event_id,))
        event = cursor.fetchone()

        if not event:
            return jsonify({'success': False, 'message': 'Event not found'}), 404

        # Generate table name based on event name
        table_name = f"{event['event_name'].lower().replace(' ', '_')}_applications"

        # Drop the applications table
        cursor.execute(f"DROP TABLE IF EXISTS {table_name}")
        conn.commit()
        conn.close()

        return jsonify({
            'success': True,
            'message': 'Event registrations closed successfully'
        })
    except Exception as e:
        return jsonify({'success': False, 'message': str(e)}), 500
```

DML Statements

(CRUD Operation Screenshots)

```python
if not event:
    return jsonify({'success': False, 'message': 'Event not found'}), 404

# Generate table name based on event name
table_name = f"{event['event_name'].lower().replace(' ', '_')}_applications"

# Ensure trigger exists
create_registration_trigger(cursor, event['event_name'])

# Get application data from request
data = request.json

try:
    # Dynamically build insert query
    columns = list(data.keys())
    placeholders = ','.join(['?' for _ in columns])
    values = [data[col] for col in columns]

    query = f"INSERT INTO {table_name} ({','.join(columns)}) VALUES ({placeholders})"

    cursor.execute(query, values)
    conn.commit()

    return jsonify({
        'success': True,
        'message': 'Application submitted successfully'
    })

except sqlite3.IntegrityError as e:
```

```python
@app.route('/api/recruiting-clubs', methods=['GET'])
def get_recruiting_clubs():
    try:
        conn = get_db_connection()
        cursor = conn.cursor()

        # Find clubs with recruitment tables
        cursor.execute("SELECT name FROM sqlite_master WHERE type='table'")
        tables = cursor.fetchall()

        recruiting_clubs = []
        for table in tables:
            table_name = table[0]
            if table_name.endswith('_recruitments'):
                # Extract club name from table name
                club_name = ' '.join(table_name.split('_')[:-1]).title()

                # Fetch club details
                cursor.execute('''
                    SELECT club_id, club_name, club_logo_image
                    FROM clubs
                    WHERE club_name = ?
                ''', (club_name,))

                club = cursor.fetchone()
                if club:
                    recruiting_clubs.append(dict(zip(
                        ['club_id', 'club_name', 'club_logo_image'],
                        club
```

Queries

(JOIN QUERY, AGGREGATE FUNCTION QUERIES AND NESTED QUERY)

```python
@app.route('/api/events', methods=['GET'])
def get_all_events():
    try:
        conn = get_db_connection()
        cursor = conn.cursor()

        # Check if table exists first
        cursor.execute("SELECT name FROM sqlite_master WHERE type='table' AND name='events'")
        if not cursor.fetchone():
            return jsonify({
                'success': False,
                'message': 'Events table does not exist'
            }), 500

        # Fetch all events with club name
        cursor.execute('''
            SELECT
                events.event_id,
                events.event_name,
                events.event_image,
                events.event_date,
                events.event_time,
                events.event_venue,
                clubs.club_name
            FROM events
            JOIN clubs ON events.club_id = clubs.club_id
        ''')

        events = cursor.fetchall()
        conn.close()
```

```python
# Create new trigger
cursor.execute(f"""
    CREATE TRIGGER {trigger_name}
    BEFORE INSERT ON {table_name}
    BEGIN
        SELECT
            CASE
                WHEN (SELECT COUNT(*) FROM {table_name}) >= 3
                THEN RAISE(ABORT, 'Registration limit reached. Maximum 3 applications allowed.')
            END;
    END;
""")
```

Stored Procedure, Functions and Triggers

```python
def create_registration_trigger(cursor, event_name):
    """Create or replace trigger for registration limit"""
    table_name = f"{event_name.lower().replace(' ', '_')}_applications"
    trigger_name = f"check_registration_limit_{table_name}"

    # Drop existing trigger if it exists
    cursor.execute(f"""
        DROP TRIGGER IF EXISTS {trigger_name};
    """)

    # Create new trigger
    cursor.execute(f"""
        CREATE TRIGGER {trigger_name}
        BEFORE INSERT ON {table_name}
        BEGIN
            SELECT
                CASE
                    WHEN (SELECT COUNT(*) FROM {table_name}) >= 3
                    THEN RAISE(ABORT, 'Registration limit reached. Maximum 3 applications allowed.')
                END;
        END;
    """)
```

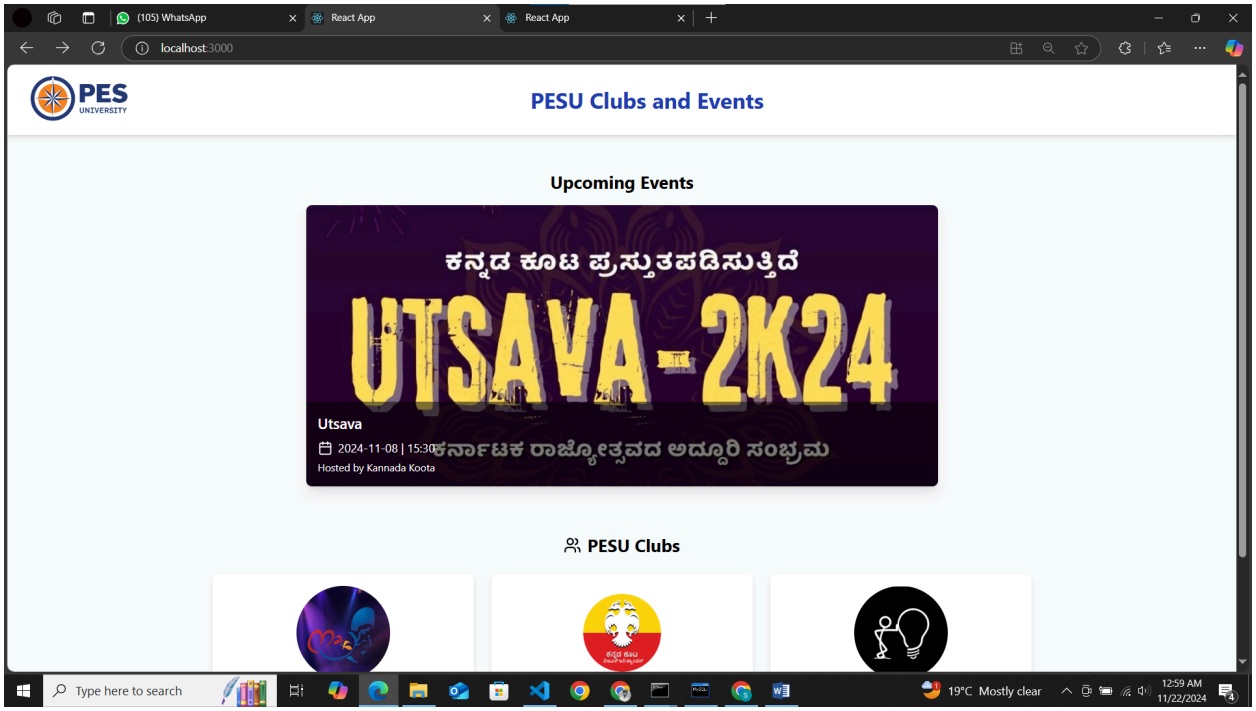
```
PROCEDURE:

DELIMITER //



CREATE PROCEDURE create_event_with_limit(

    IN p_club_id INT,

    IN p_event_name VARCHAR(255),

    IN p_event_description TEXT,

    IN p_event_date DATE,

    IN p_event_image VARCHAR(255),

    IN p_event_time TIME,

    IN p_event_venue VARCHAR(255),

    OUT p_success BOOLEAN,

    OUT p_message VARCHAR(255),
```

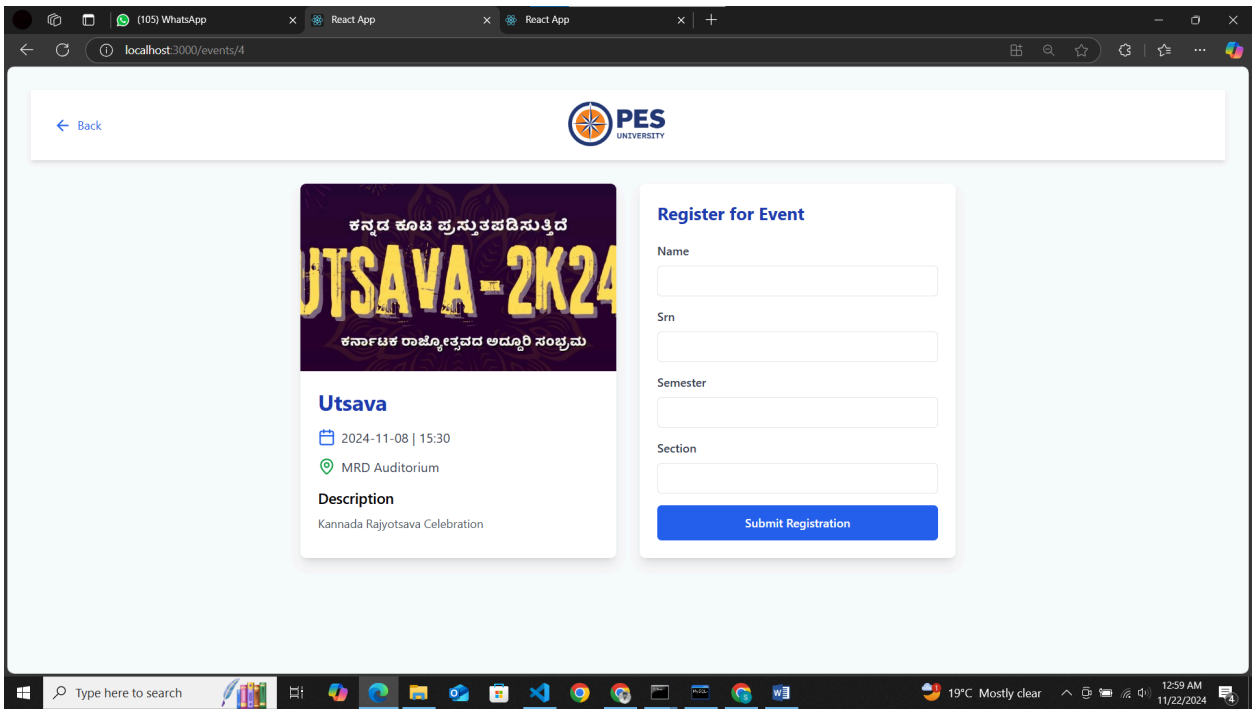
```
    OUT p_event_id INT
)
BEGIN
    DECLARE event_count INT;

    -- Get current event count for the club
    SELECT COUNT(*) INTO event_count
    FROM events
    WHERE club_id = p_club_id;

    -- Check if limit is reached
    IF event_count >= 3 THEN
        SET p_success = FALSE;
        SET p_message = 'Club has reached the maximum limit of 3 events';
        SET p_event_id = -1;
    ELSE
        -- Insert new event
        INSERT INTO events (
            club_id,
            event_name,
            event_description,
            event_date,
            event_image,
            event_time,
            event_venue
```

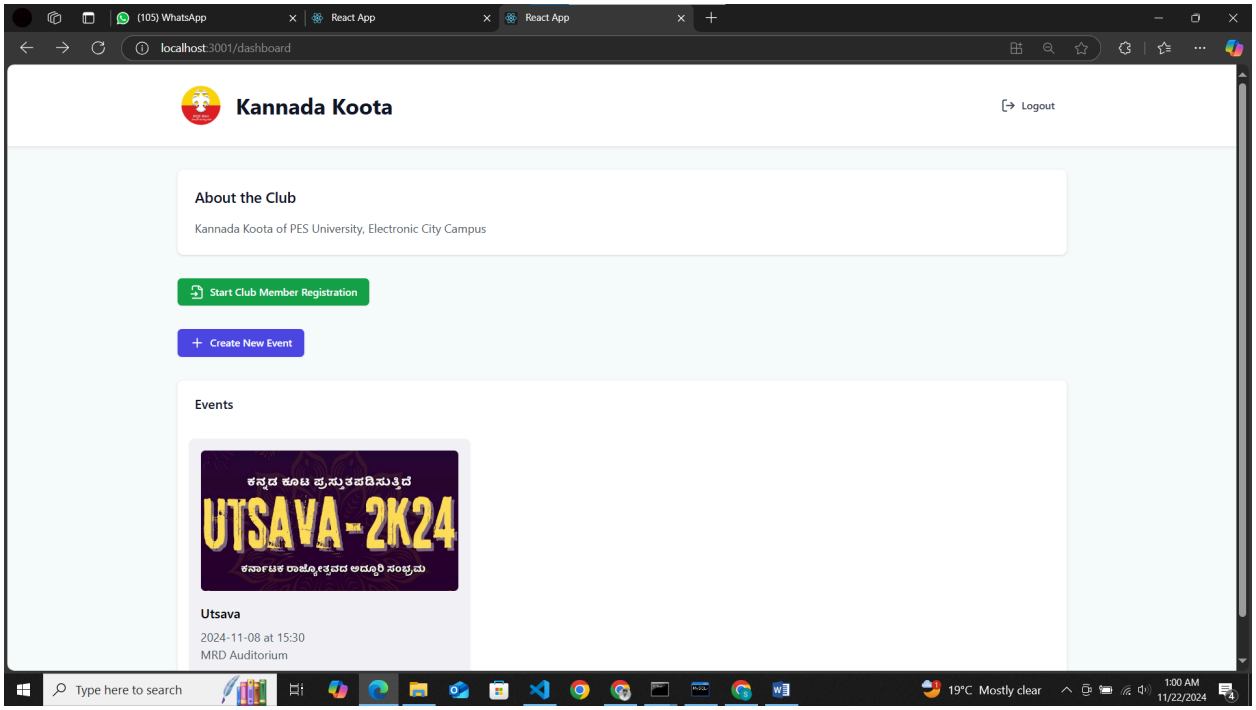
```
        ) VALUES (
            p_club_id,
            p_event_name,
            p_event_description,
            p_event_date,
            p_event_image,
            p_event_time,
            p_event_venue
        );

        -- Get the new event ID
        SET p_event_id = LAST_INSERT_ID();
        SET p_success = TRUE;
        SET p_message = 'Event created successfully';
    END IF;
END //

DELIMITER ;
```

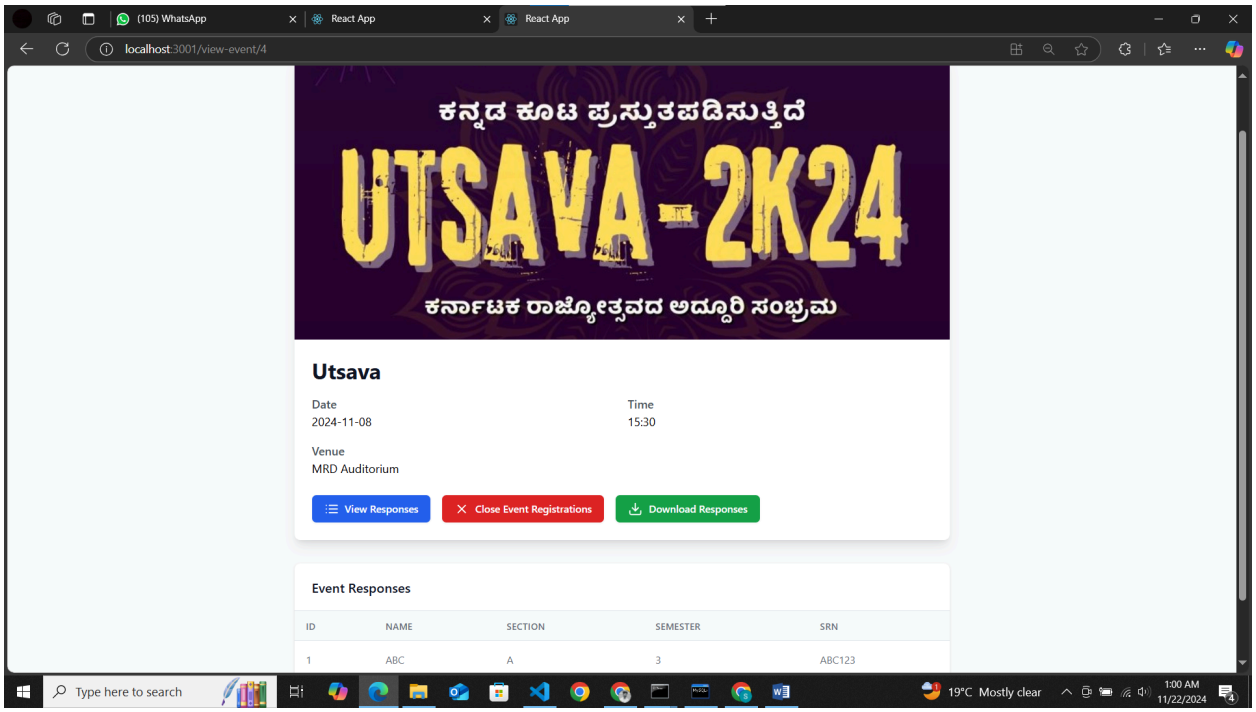
Front End Development

(Functionalities/Features Of The Application)






(105) WhatsApp
React App
React App
localhost:3001/dashboard
Kannada Koota
Logout
About the Club
Kannada Koota of PES University, Electronic City Campus
Start Club Member Registration
Create New Event
Events
ಕನ್ನಡ ಕೂಟ ಪ್ರಸ್ತುತಪಡಿಸುತ್ತಿದೆ
UTSAVA-2K24
ಕರ್ನಾಟಕ ರಾಜ್ಯೋತ್ಸವದ ಅದ್ಧೂರಿ ಸಂಭ್ರಮ
Utsava
2024-11-08 at 15:30
MRD Auditorium
Type here to search
19°C Mostly clear
1:00 AM
11/22/2024


(105) WhatsApp
React App
React App
localhost:3001/view-event/4
ಕನ್ನಡ ಕೂಟ ಪ್ರಸ್ತುತಪಡಿಸುತ್ತಿದೆ
UTSAVA-2K24
ಕರ್ನಾಟಕ ರಾಜ್ಯೋತ್ಸವದ ಅದ್ಧೂರಿ ಸಂಭ್ರಮ
Utsava
Date
2024-11-08
Time
15:30
Venue
MRD Auditorium
View Responses
Close Event Registrations
Download Responses
Event Responses
ID
NAME
SECTION
SEMESTER
SRN
1
ABC
A
3
ABC123
Type here to search
19°C Mostly clear
1:00 AM
11/22/2024

References/Bibliography

The development of this project was *inspired by the challenges we, as club heads at PES University*, frequently faced in collecting and managing student data for events, recruitment drives, and workshops. These experiences highlighted the inefficiencies of using repetitive Google Forms and Excel sheets, motivating us to design a centralized solution.

Our technical implementation is supported by the following resources:

- **React JS Documentation:** https://legacy.reactjs.org/docs/getting-started.html
- **Flask Documentation:** https://flask.palletsprojects.com/en/stable/
- **Sqlite3 Documentation:** https://docs.python.org/3/library/sqlite3.html
- **pesuacademy.com:** https://www.pesuacademy.com/
- **clubs.pes.edu:** https://clubs.pes.edu/
- **Database Management System Course Material:**
  Prescribed slides and materials for the course *Database Management System: UE22CS351A*, forming the foundation for relational database design and queries.

In addition, this project incorporates real-world insights gained from the event management ecosystem at PES University and informal feedback from peers and fellow club organizers. This blend of academic learning and practical experience enabled us to develop an efficient, scalable, and user-centric solution tailored to the unique needs of our vibrant university community.

Appendix A Definitions, Acronyms And Abbreviations

Definitions

- **Database Management System (DBMS)**: A software system used for managing databases, allowing users to interact with the data using queries and commands.
- **Relational Database**: A type of database that organizes data into tables (relations) with predefined relationships.
- **Event Management System**: A platform to manage events, workshops, recruitment drives, and related student participation effectively.
- **Role-Based Access Control (RBAC)**: A method of regulating access to the system based on user roles (e.g., Admin, Student).
- **Event**: Activities such as workshops, recruitment drives, or club activities organized for student participation.
- **Club Head**: A role within the platform responsible for managing club events, recruitments, and student interactions.
- **Enrollment**: The process of students registering for events or workshops.
- **Custom Questions**: Event-specific questions created by club heads to gather additional details from participants.

Acronyms

- **PESU**: People's Education Society University
- **DBMS**: Database Management System
- **ER**: Entity-Relationship
- **UI**: User Interface
- **API**: Application Programming Interface
- **URS**: User Requirement Specification
- **SRS**: Software Requirement Specification
- **RBAC**: Role-Based Access Control
- **PII**: Personally Identifiable Information

Abbreviations

- **ID**: Identifier
- **Admin**: Administrator
- **Info**: Information
- **Desc**: Description
- **Auth**: Authentication
- **Config**: Configuration
- **Stats**: Statistics
- **Sys**: System

Database Systems

- **PK**: Primary Key
- **FK**: Foreign Key
- **CRUD**: Create, Read, Update, Delete
- **DML**: Data Manipulation Language
- **DDL**: Data Definition Language

Technical Terms

- **SSL**: Secure Sockets Layer
- **HTTP**: Hypertext Transfer Protocol
- **HTTPS**: Hypertext Transfer Protocol Secure
- **JSON**: JavaScript Object Notation
- **SQL**: Structured Query Language