TABLE OF CONTENTS

1. TITLE**2**

2. INTRODUCTION & OBJECTIVES**3**

3. PROJECT CATEGORY**4**

4. ANALYSIS**4**

a. DATA FLOW DIAGRAM4

a. ENTITY RELATIONSHIP DIAGRAM7

5. COMPLETE PROJECT STRUCTURE**8**

a. MODULES WITH DESCRIPTION WITH EFFORT ESTIMATION 8

b. DATA STRUCTURE FOR ALL MODULES8

c. DATABASE DESIGN8

d. MODULES WITH PROCESS LOGIC14

e. TESTING DETAILS15

f. REPORT GENERATION15

6. TOOLS/ PLATFORM, HARDWARE & SOFTWARE SPECIFICATIONS**16**

7. PROJECT FOR ANY INDUSTRY/CLIENT: NO

8. FUTURE SCOPE AND FURTHER ENHANCEMENT **16**

# Kalari Arts Platform

# INTRODUCTION

The **Kalari Arts Platform** is a web-based application designed to promote and preserve the traditional martial art of Kalari from Kerala, India. It serves as a comprehensive hub for practitioners, instructors, and enthusiasts, offering features such as class enrollment, instructional videos, training schedules, and event participation. With modules for students, instructors, and admins, users can access personalized learning materials, participate in Kalari events, and manage payments, while instructors oversee classes and provide feedback. The platform fosters community engagement and supports the revitalization of Kalari in modern society.

**OBJECTIVES**

* Enhance Communication: Facilitate seamless communication between villagers and the Gram Panchayat to address public grievances and queries effectively.
* **Increase Transparency**: Implement a transparent system that allows users to track their applications and understand the status of various government schemes.
* **Simplify Service Access**: Provide a user-friendly interface for villagers to easily access information about services, apply for schemes, and submit complaints or feedback.
* **Streamline Administrative Processes**: Automate administrative tasks such as application approvals, payments, and service updates to improve efficiency within the Panchayat.
* **Empower Citizens**: Enable citizens to take an active role in local governance by providing them with the tools and resources needed to access essential services and participate in decision-making.
* **Promote Digital Literacy**: Encourage digital literacy among villagers by providing training and support for using the application, thus bridging the digital divide in rural areas.
* **Facilitate Real-time Monitoring**: Allow Panchayat officials to monitor service requests and applications in real-time, ensuring timely responses to public needs and concerns.
* **Improve Resource Management**: Optimize the allocation and management of resources by tracking service delivery and identifying areas for improvement in local governance.
* **Support Sustainable Development**: Align the application with government initiatives aimed at sustainable rural development, ensuring that services provided contribute to the overall growth and welfare of the community.

**PROJECT CATEGORY**

The **Kalari Arts Platform** is categorized as a **Full Stack Web Application** that utilizes **Object-Oriented Programming (OOP)** and a **Relational Database Management System (RDBMS)** for effective data management and functionality. By implementing OOP principles, the platform creates modular and reusable components, encapsulating key entities like **Users**, **Instructors**, **Training Modules**, and **Events** as objects. This design enhances maintainability and scalability, allowing developers to update specific features without affecting the entire system.

**SYSTEM ANALYSIS**

# System Analysis for the Kalari Arts Platform entails a thorough assessment of user requirements and existing practices to create a solution that effectively promotes and preserves the traditional martial art of Kalari. This phase begins with gathering input from stakeholders, including practitioners and instructors, to identify their needs, such as user registration, course management, event scheduling, and payment processing. A feasibility study evaluates the platform's technical, operational, and economic viability while addressing potential risks and challenges. The insights gained from this analysis will inform the platform's design and development, ensuring it meets user expectations, maintains usability, and contributes to the successful promotion of Kalari as a vital cultural heritage.

# DATA MODELS

1. **DATA FLOW DIAGRAM: LEVEL 0**

**LEVEL 1 DFD ADMIN**

**(b)ENTITY RELATIONSHIP DIARAM**

# A COMPLETE PROJECT STRUCTURE

### (a)MODULES AND DESCRIPTION WITH EFFORT ESTIMATION

**1. User Module**

This module allows students and practitioners to register, log in, manage their profiles, and view available classes. Users can enroll in Kalari training sessions, access instructional materials, and provide feedback on their learning experience. The module also supports personalized recommendations and progress tracking for individual users.  
**Effort Estimation:** 45 hours

**2. Instructor Module**

Instructors use this module to manage their classes, update training schedules, and interact with students. They can upload instructional materials, track student attendance, assess performance, and respond to feedback. The module facilitates efficient communication between instructors and students to enhance the learning experience.  
**Effort Estimation:** 40 hours

**3. Admin Module**

Admins oversee the platform by managing user and instructor accounts, class offerings, and financial transactions. This module includes the ability to generate detailed reports on user engagement, class attendance, course enrollment, and payment history. Admins can address issues, configure platform settings, and ensure smooth operation of the system.  
**Effort Estimation:** 40 hours

**4. Payment Module**

The Payment Module allows students to make secure payments for Kalari classes and events. Admins can manage these transactions, track payment histories, and generate financial reports. This module ensures a seamless and secure payment process for users and enables admins to handle financial operations efficiently.  
**Effort Estimation:** 25 hours

**5. Event Management Module**

This module allows admins to create and manage Kalari events, such as workshops or tournaments, while students can register and participate. The system tracks event details, participant lists, and schedules, providing updates to users. Instructors can also share event-related materials to help students prepare.  
**Effort Estimation:** 30 hours

**6. Training Materials Module**

This module facilitates the upload and distribution of Kalari training resources, including videos, articles, and manuals. Instructors can easily share materials with students, and users can access and download these resources to enhance their learning. The module ensures efficient resource management and delivery.  
**Effort Estimation:** 25 hours

**Total Effort Estimation: Total:** 205 hours

### (b)DATA STRUCTURES FOR ALL THE MODULES.

### The Kalari Arts Platform utilizes various data structures across its modules: the User Module employs fields such as User ID, Name, Email, Password, Profile Picture, Enrolled Classes, and Feedback, while the Instructor Module contains Instructor ID, Name, Email, Password, Classes Taught, and Availability for efficient management of user and instructor information.

**(c)DATABASE DESIGN**

**1. Users Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **user\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for each user** |
| **username** | **VARCHAR(50)** | **NOT NULL, UNIQUE** | **Username for user login** |
| **password** | **VARCHAR(255)** | **NOT NULL** | **User's password (hashed)** |
| **email** | **VARCHAR(100)** | **NOT NULL, UNIQUE** | **User's email address** |
| **role** | **ENUM('student', 'admin', 'instructor')** | **NOT NULL** | **Role of the user in the system** |
| **created\_at** | **DATETIME** | **NOT NULL** | **Timestamp for when the user was created** |
| **updated\_at** | **DATETIME** |  | **Timestamp for the last profile update** |

**2. Instructors Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **instructor\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for each instructor** |
| **user\_id** | **INT** | **NOT NULL, FOREIGN KEY REFERENCES Users(user\_id)** | **Link to the user account** |
| **bio** | **TEXT** |  | **Brief biography of the instructor** |
| **experience\_years** | **INT** |  | **Years of experience in Kalari** |
| **created\_at** | **DATETIME** | **NOT NULL** | **Timestamp for when the instructor was added** |
| **updated\_at** | **DATETIME** |  | **Timestamp for the last update** |

**3. Classes Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **class\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for each class** |
| **instructor\_id** | **INT** | **NOT NULL, FOREIGN KEY REFERENCES Instructors(instructor\_id)** | **Instructor teaching the class** |
| **class\_name** | **VARCHAR(100)** | **NOT NULL** | **Name of the class** |
| **schedule** | **DATETIME** | **NOT NULL** | **Date and time of the class** |
| **duration** | **INT** | **NOT NULL** | **Duration of the class in minutes** |
| **created\_at** | **DATETIME** | **NOT NULL** | **Timestamp for when the class was created** |
| **updated\_at** | **DATETIME** |  | **Timestamp for the last update** |

**4. Enrollments Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **enrollment\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for each enrollment** |
| **user\_id** | **INT** | **NOT NULL, FOREIGN KEY REFERENCES Users(user\_id)** | **Student enrolling in the class** |
| **class\_id** | **INT** | **NOT NULL, FOREIGN KEY REFERENCES Classes(class\_id)** | **Class the student is enrolled in** |
| **enrollment\_date** | **DATETIME** | **NOT NULL** | **Date when the student enrolled** |

**5. Events Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **event\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for each event** |
| **event\_name** | **VARCHAR(100)** | **NOT NULL** | **Name of the event** |
| **event\_date** | **DATETIME** | **NOT NULL** | **Date and time of the event** |
| **location** | **VARCHAR(255)** |  | **Location of the event** |
| **instructor\_id** | **INT** | **FOREIGN KEY REFERENCES Instructors(instructor\_id)** | **Instructor leading the event** |
| **created\_at** | **DATETIME** | **NOT NULL** | **Timestamp for when the event was created** |
| **updated\_at** | **DATETIME** |  | **Timestamp for the last update** |

**6. Payments Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **payment\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for each payment** |
| **user\_id** | **INT** | **NOT NULL, FOREIGN KEY REFERENCES Users(user\_id)** | **User making the payment** |
| **amount** | **DECIMAL(10, 2)** | **NOT NULL** | **Amount paid** |
| **payment\_date** | **DATETIME** | **NOT NULL** | **Date when the payment was made** |
| **class\_id** | **INT** | **FOREIGN KEY REFERENCES Classes(class\_id)** | **Class associated with the payment** |
| **created\_at** | **DATETIME** | **NOT NULL** | **Timestamp for when the payment was created** |

**7. Training Materials Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **material\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for each training material** |
| **instructor\_id** | **INT** | **NOT NULL, FOREIGN KEY REFERENCES Instructors(instructor\_id)** | **Instructor uploading the material** |
| **title** | **VARCHAR(100)** | **NOT NULL** | **Title of the training material** |
| **description** | **TEXT** |  | **Description of the training material** |
| **file\_url** | **VARCHAR(255)** | **NOT NULL** | **URL to access the material** |
| **created\_at** | **DATETIME** | **NOT NULL** | **Timestamp for when the material was uploaded** |

**8. Admins Table**

| **Column Name** | **Data Type** | **Constraints** | **Description** |
| --- | --- | --- | --- |
| **admin\_id** | **INT** | **PRIMARY KEY, AUTO\_INCREMENT** | **Unique identifier for each admin** |
| **user\_id** | **INT** | **NOT NULL, FOREIGN KEY REFERENCES Users(user\_id)** | **Link to the admin user account** |
| **created\_at** | **DATETIME** | **NOT NULL** | **Timestamp for when the admin was added** |
| **updated\_at** | **DATETIME** |  | **Timestamp for the last update** |

### (d)MODULES WITH PROCESS LOGIC

* **Student Module**

The Student Module allows students to register by providing personal details and creating a profile, enabling them to log in to access the platform. After logging in, they can browse available classes and schedules, enrolling in classes by selecting them and completing the payment process. Once enrolled, students can access training materials and resources associated with their classes. They can also view upcoming Kalari events, register for them, and participate actively. After attending classes or events, students have the opportunity to provide feedback on their experiences.

* **Instructor Module**

The Instructor Module enables instructors to register and create profiles, allowing them to log in to manage their classes. Instructors can create new classes, update schedules, and upload training materials to support their students. They can view enrolled students, track attendance, and monitor student performance. Additionally, instructors can respond to feedback provided by students to enhance the learning experience.

* **Admin Module**

In the Admin Module, admins can log in to manage user and instructor accounts, ensuring proper access and permissions for all platform users. They oversee class offerings, with the ability to edit or remove classes as needed. The module also allows admins to generate reports on user engagement, class attendance, and financial transactions. Moreover, admins are responsible for addressing user concerns and troubleshooting any platform-related issues reported by users or instructors.

* **Payment Module**

The Payment Module facilitates the payment process for students after enrolling in a class. Students navigate to the payment section to view payment details and methods, completing the transaction. Once a payment is made, the system confirms the transaction and updates the student’s enrollment status. Payment records are maintained in the system for accounting and reporting purposes.

* **Events Module**

In the Events Module, students can browse upcoming Kalari events and view essential details like date, time, and location. After registering for an event, the system updates the student’s profile to reflect their participation. Instructors can also add new events and manage registrations, ensuring that all interested students can participate.

* **Training Materials Module**

The Training Materials Module allows instructors to upload training materials, categorizing them by class or topic for easier access. Students can access these materials based on their enrolled classes or personal interests. The system tracks which materials have been accessed by students, aiding instructors in tailoring their teachings and ensuring that all necessary resources are utilized.

### (e)TESTING DETAILS

1.**Unit Testing**

Unit testing involves verifying individual components or functions of the application to ensure that each part operates correctly. In the context of the Kalari Arts Platform, this would involve testing functionalities such as user registration, class enrollment, and feedback submission in isolation. The goal is to catch bugs and issues at an early stage, making it easier to ensure that each unit performs its specified task accurately.

**2. Integration Testing**

Integration testing focuses on assessing how different modules or components of the application work together. For the Kalari Arts Platform, this would involve testing the interactions between the User, Instructor, and Admin modules. It ensures that data flows correctly between these modules, such as verifying that class enrollment information is properly recorded and accessible to both users and instructors. This type of testing helps identify any issues arising from the integration of different parts of the application.

**3. System Testing**

System testing is a comprehensive testing phase that evaluates the entire application as a complete system. It includes functional and non-functional testing to verify that the application meets specified requirements and performs as expected in a real-world environment. For the Kalari Arts Platform, system testing would involve testing all features, such as searching for classes, viewing schedules, and submitting payments, to ensure they work seamlessly together. This testing phase also checks for performance, security, and usability to ensure a positive user experience.

(f) REPORT GENERATION

* User Reports
* Class Reports
* Instructor Reports
* Payment Reports
* Feedback Reports
* Administrative Reports

**HARDWARE & SOFTWARE SPECIFICATIONS**

**Hardware Specifications**

* Processor: Quad-core processor.
* RAM: Minimum of 16 GB.
* Storage: SSD with at least 256 GB.

**Software Specifications**

* Operating System: Linux (Ubuntu/CentOS) or Windows.
* Database: MySQL.
* Programming Languages:
  + Front-end: HTML, CSS, JavaScript.
  + Back-end: Python (Django framework).
* Development Tools: Visual Studio Code, Git for version control.

**FUTURE SCOPE AND FURTHER ENHANCEMENT OF THE PROJECT**

The **Kalari Arts Platform** has significant potential for future enhancements and expansions. Some possible areas for development include:

1. **Mobile Application Development**: Creating a dedicated mobile app for both Android and iOS platforms to increase accessibility and engagement among users.
2. **Enhanced User Experience**: Incorporating advanced features such as personalized user dashboards, AI-driven recommendations for classes based on user preferences, and interactive learning modules to improve user interaction.
3. **Community Building**: Establishing forums and social networking features to allow users to connect, share experiences, and collaborate on projects related to Kalari arts, fostering a sense of community.
4. **Integration with Other Platforms**: Collaborating with educational institutions or cultural organizations to provide additional resources and promotional opportunities, including webinars, workshops, and collaborative events.
5. **Performance Analytics**: Implementing analytics tools for instructors to track student progress and outcomes, allowing for tailored teaching approaches and improved class offerings.
6. **International Outreach**: Expanding the platform's reach to a global audience by offering classes in multiple languages and introducing online courses that cater to diverse cultural contexts.
7. **Virtual Reality (VR) Experiences**: Exploring the use of VR technology to create immersive training environments for users, enhancing the traditional Kalari experience with modern technology.

----------------------------------------------