EthnoVerse: A Digital Framework for Cultural Archiving

Kaavish Report
presented to the academic faculty
by

Maria Adnan	ma08439
Sara Baloch	sb08586
Afifah Uzair	au07879

In partial fulfillment of the requirements for
Bachelor of Science
Computer Science

Dhanani School of Science and Engineering

Habib University
Spring 2022

EthnoVerse: A Digital Framework for Cultural Archiving

This Kaavish project was supervised by:

____

Dr. Syeda Saleha Raza
Faculty of Computer Science
Habib University

Approved by the Faculty of Computer Science on ________________________.

Dedication

For ammi, abbu, and pappu.

Acknowledgements

We want to thank the CS faculty and ...

Abstract

In an age where technology enables the preservation and spread of knowledge, many cultural practices—especially those of marginalized communities—remain undocumented or inaccessible. Our idea for this project grew out of a Tehqiq 2025 ethnographic study on tattoo traditions among Sindh's Indigenous Hindu Tribes, where we found that living practices were largely absent from public and scholarly archives. This absence reflects a broader gap: there is no centralized, accessible system to document Pakistan's Indigenous communities as part of its living cultural landscape. EthnoVerse: A Digital Framework for Cultural Archiving addresses this gap by creating a web-based repository for multimedia documentation—texts, images, oral histories, and videos—organized through searchable metadata. Beyond static records, the project introduces an experimental 3D module that reconstructs community environments using Neural Radiance Fields (NeRF), allowing users to experience immersive virtual tours. Technically, EthnoVerse integrates database systems, web development, and computer vision into a scalable, ethically grounded platform. Socially, it reimagines archiving as a dynamic and participatory process—one that situates Indigenous communities not in the past, but within Pakistan's evolving cultural present.

Contents

1 Introduction **9**

1.1 Problem Statement 9

1.2 Proposed Solution 9

1.3 Intended Users 10

1.3.1 User Characteristics 10

1.4 Project Gantt chart and deliverables 10

1.5 Key Challenges and Mitigations 12

1.5.1 Technical complexity of 3D reconstruction 12

1.5.2 Data quality & quantity 12

1.5.3 Compute resources 12

1.5.4 Scope creep & time management 12

2 Literature Review **13**

3 Software Requirement Specification (SRS) **14**

3.1 Functional Requirements 14

3.1.1 EthnoVerse Web Platform 14

3.1.2 3D Reconstruction and Virtual Tour 16

3.2 Non-Functional Requirements 18

3.3 External Interfaces 19

3.3.1 User Interfaces 19

3.3.2 Hardware and Communication Interfaces 24

3.4 Use Cases 24

3.4.1 Use Case Diagram 24

3.5 System Diagram 25

4 Software Design Specification (SDS) **27**

4.1 Software Design 27

4.2 Data Design 27
4.3 Technical Details 27

5 Methodology, Experiments and Results OR Testing and Analysis 28

6 Conclusion and Future Work 29

7 Reflection 30

Appendix A More Math 31

Appendix B Data 32

Appendix C Code 33

References 34

List of Figures

1.1 Project Gantt Chart for EthnoVerse 11

List of Tables

3.1 Functional requirements for EthnoVerse web platform. 15
3.2 Functional requirements for the 3D reconstruction module. 17

1. Introduction

1.1 Problem Statement

Indigenous tribal communities in Pakistan are routinely under-represented in public archives and scholarly resources. While valuable, existing archival bodies in Pakistan primarily hold government, colonial, or language-specific records and do not provide accessible, multimedia documentation of contemporary tribal community life. As a result, cultural practices, oral histories, and built environments of these communities remain fragmented, inaccessible, or framed only as historical relics rather than as part of a continuing cultural present. This project addresses the absence of a centralized, searchable, multimedia archive that documents tribal communities in ways that are accessible to researchers, educators, and the public. Our motivation is both technical (to build a robust, extensible system demonstrating MERN-stack, and 3D reconstruction techniques) and social (to make community knowledge discoverable and ethically represented).

1.2 Proposed Solution

We propose **a web-based, extensible Digital Archival System (DAS)** to create a centralized platform to document and make accessible the cultural practices, oral histories, and artistic traditions of tribal communities in Pakistan. The system will allow authenticated administrators to upload multimedia content—text, images, audio, and video—along with metadata for categorization. Public users can browse and search this content through an intuitive, user-friendly interface. The platform aims to bridge the information gap by providing an open, structured, and searchable repository for cultural documentation that is typically fragmented across private archives. For proof of concept, the platform will include a dataset from ethnographic work with the Kolhi community in Tharparkar.

1.3 Intended Users

EthnoVerse will serve multiple types of users, each interacting with the system in different ways:

- **General Public/Researchers, Students, and Scholars:** Individuals seeking to learn about Pakistan's tribal communities in an accessible and engaging manner. They can browse collections, view images and videos, listen to oral histories, and experience immersive 3D tours that bring cultural practices to life. Moreover, academics, educators, and students interested in ethnography, anthropology, history, and cultural studies. They can search through tagged data, access detailed metadata for archival material, and utilize the system's structured categorization for academic research or coursework.
- **Administrators:** Authorized personnel responsible for managing the archive's content and system operations. They handle user authentication, upload new media, edit or remove existing entries, and ensure the ethical and technical maintenance of the archive. Administrators are also responsible for ensuring data integrity, backups, and system updates.

1.3.1 User Characteristics

- **General Public:** Basic computer literacy; likely to access the system via desktop or mobile browsers. Motivated by curiosity or cultural interest.
- **Researchers/Students:** Moderate to advanced digital literacy; familiar with academic research tools and metadata use. Will require precision search and possibly citation functionality.
- **Administrators:** Advanced technical skills; capable of using web-based admin panels, managing databases, and ensuring data consistency.

1.4 Project Gantt chart and deliverables

Kaavish I Deliverables

- Software Requirement Specification (SRS)
- Software Design Specification (SDS)

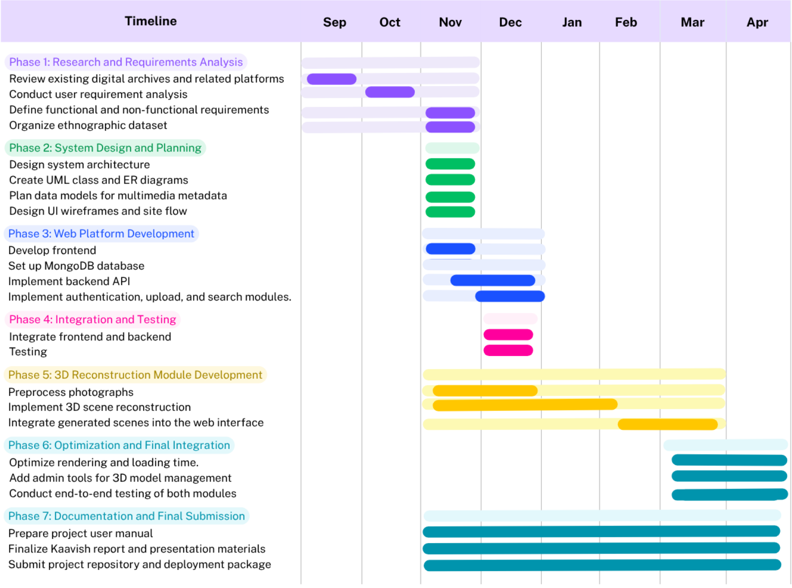


Figure 1.1: Project Gantt Chart for EthnoVerse

- Functional Web Platform Prototype (MERN-based)
- Preliminary Dataset Integration (Kolhi community)
- Testing and Evaluation Report

Kaavish II Deliverables

- 3D Reconstruction Module (NeRF / Gaussian Splatting)
- Integrated Virtual Tour Interface (Three.js-based)
- Optimization and Final Integration of all modules
- Final Report, Presentation, and Deployed System

1.5 Key Challenges and Mitigations

1.5.1 Technical complexity of 3D reconstruction

- **Risk:** Implementing NeRF/Gaussian Splatting from scratch is technically demanding and GPU-intensive.
- **Mitigation:** Treat the neural reconstruction as an experimental PoC with a documented fallback (photo-based panoramic/stitched virtual tour). Use preexisting open-source implementations where licensing allows; limit model scope to a single site for the final deliverable if resources/time are constrained.

1.5.2 Data quality & quantity

- **Risk:** Insufficient or uneven photographic coverage can prevent accurate 3D reconstruction or reduce media utility.
- **Mitigation:** Specify photo capture guidelines, and request additional images from field collaborators.

1.5.3 Compute resources

- **Risk:** Training or running reconstruction models may require more GPU capacity than available on local machines.
- **Mitigation:** Use lightweight model variants, perform experimental runs on cloud credits, or restrict training to smaller scene subsets. Document exact hardware used so results are reproducible.

1.5.4 Scope creep & time management

- **Risk:** Ambitious goals (multi-community coverage, advanced search, full 3D) could exceed Kaavish timelines.
- **Mitigation:** Use the Kolhi dataset as a bounded PoC; prioritise core archival functionality (ingestion, metadata, search, admin) during Kaavish I and reserve advanced features for Kaavish II. Maintain weekly sprints and milestone checks.

2. Literature Review

This chapter presents the current state of the art in the domain and talks about other similar work that has been done in this area. It also establishes the novelty of our work by highlighting the differences between the existing work and our work.

We will keep updating this chapter (especially if our project is research-intensive) as our research proceeds and we come across more work related to our problem.

Of course, we take inspiration from [1] but wish the work was typeset in LaTeX[3], e.g. by taking help from [2].

3. Software Requirement Specification (SRS)

This Software Requirements Specification (SRS) outlines the software and system requirements for EthnoVerse. It details functional and non-functional requirements for system capabilities. The document also includes external interfaces to clarify the system's design and interactions.

3.1 Functional Requirements

This section specifies the functional requirements of the system. The requirements are divided into two primary categories: those pertaining to the **web platform** and those related to the **3D virtual tour module** developed using Neural Radiance Fields (NeRF). Each subsection includes a tabular summary and a module-wise functional outline for clarity.

3.1.1 EthnoVerse Web Platform

The web platform is responsible for storing, displaying, and managing multimedia data related to community archives. Only administrators have access to content management features, while public users can freely view and search content without authentication.

ID	Function Name	Description	Users
FR-W01	Admin Authentication	Admins log in securely using authentication to access upload, edit, and delete functionalities.	Admin
FR-W02	Media Upload	Admin uploads multimedia content (text, image, audio, video) with metadata (title, tags, community, description, consent).	Admin
FR-W03	Edit/Delete Content	Admin edits metadata or removes outdated/incorrect items. All edits are logged.	Admin
FR-W04	Metadata Management	Validates and stores metadata; supports updates and versioning.	Admin
FR-W05	Search Content	Users search items by title, tags, or description.	Admin, Public
FR-W06	Filter Results	Users filter items by community, media type, or upload date.	Admin, Public
FR-W07	View Media	Public users view multimedia content through responsive layouts.	Public
FR-W08	Dashboard Overview	Admin dashboard displays statistics such as total uploads and media distribution.	Admin
FR-W09	Backup and Data Integrity	Automated database backups and consistency checks ensure data reliability.	System
FR-W10	Logs and Audit Trail	Records admin actions (login, upload, edit, delete) for transparency.	System, Admin
FR-W11	Scalability and Generalization	System supports addition of new communities without structural changes or degradation. Database schema and API design ensure scalability and modular extensibility.	System, Admin

Table 3.1: Functional requirements for EthnoVerse web platform.

Module-wise Outline

- **Module 1: Admin Authentication**
  - Secure login and session management.
  - Only authenticated admins can perform CRUD operations on content.

- **Module 2: Archive Content Management**
  - Upload of multimedia items with validated metadata.
  - Edit or delete entries as required, with logs maintained for accountability.
  - Metadata management supporting version control and validation.
- **Module 3: Search and Discovery**
  - Keyword search across titles, tags, and descriptions.
  - Filtering by community, content type, or upload date.
- **Module 4: Content Presentation**
  - Responsive user interface (React.js) displaying media grid and detailed views.
  - Embedded media players for image, audio, and video files.
- **Module 5: System Administration**
  - Dashboard summarizing total uploads and user activity.
  - Automated data backups and integrity verification.
  - Logs for uploads, edits, deletions, and admin sessions.
- **Module 6: System Extensibility and Scalability**
  - Supports addition of new communities and datasets without structural changes.
  - Allows integration of new media types through modular schema design.
  - Ensures smooth performance as data volume and content grow.

3.1.2 3D Reconstruction and Virtual Tour

The 3D module demonstrates the technical application of computer graphics techniques to reconstruct community environments from photographs.

ID	Function Name	Description	Users
FR-3D01	Dataset Preparation	Collect and preprocess multiple-angle photographs of community spaces for NeRF input.	Admin
FR-3D02	Scene Reconstruction	Generate 3D radiance field representations using NeRF or Gaussian Splatting techniques.	System
FR-3D03	Rendering and Visualization	Render 3D scenes into navigable virtual tours integrated into the web interface (via Three.js).	Public, Admin
FR-3D04	Scene Management	Store, update, or remove 3D models and their metadata (scene name, location, source images).	Admin
FR-3D05	Performance Optimization	Optimize rendering quality and load times for web deployment.	System
FR-3D06	Fallback Handling	If NeRF reconstruction fails, fallback to photo-based 360° panoramic tour is generated.	System

Table 3.2: Functional requirements for the 3D reconstruction module.

Module-wise Outline

- **Module 1: Data Preparation**
  - Capture or collect photographs of community sites (courtyards, huts, shrines) from multiple angles.
  - Preprocess images (resize, align, normalize lighting) before NeRF input.
- **Module 2: Scene Reconstruction**
  - Implement NeRF or Gaussian Splatting for 3D radiance field estimation.
  - Generate scene models stored in compressed formats for web use.
- **Module 3: Rendering and Visualization**
  - Convert reconstructed 3D scenes into Three.js environments.
  - Embed virtual tours within the frontend interface.
- **Module 4: Scene Management and Maintenance**
  - Store reconstructed models and associated metadata (scene name, date, contributors).

- Allow admin to update or delete 3D scenes.
  - Implement caching and loading optimization for performance.

- **Module 5: Fallback and Error Handling**
  - If reconstruction fails or resources are insufficient, generate a static 360° panoramic viewer.
  - Log all reconstruction processes and errors for debugging.

3.2 Non-Functional Requirements

The following non-functional requirements ensure the performance, reliability, scalability, and ethical compliance of the EthnoVerse system.

- **Performance:**
  - The system must support at least 50 concurrent users with average response times under 2 seconds for typical queries.
  - Search results should render within 1–3 seconds for datasets containing up to 10,000 items.
  - Media files must be optimized for web display using compressed images and adaptive video streaming techniques.
- **Reliability:**
  - The database must maintain at least 99% uptime under normal conditions.
  - Failed upload sessions should automatically retry or roll back transactions to preserve data integrity.
  - The system should recover gracefully from unexpected shutdowns or network interruptions.
- **Security:**
  - All administrator passwords must be stored using salted hashing.
  - JWT-based authentication must be enforced for all protected endpoints.
  - HTTPS must be enabled on deployment; role-based authorization will restrict all CRUD operations to authenticated admins.

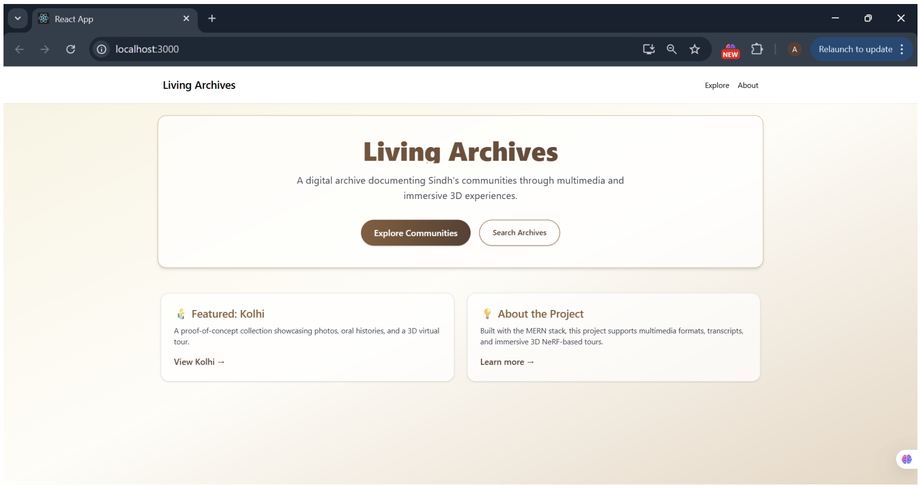
- **Usability:**
  - The user interface must follow WCAG 2.1 accessibility standards and maintain a responsive design across devices.
  - The upload interface should include clearly labeled fields and drag-and-drop functionality.
  - Search and filter options must remain visible and accessible on all main views.
- **Compliance and Ethics:**
  - Ethical representation of indigenous content will be ensured with support from an SDP professor and student volunteer specializing in ethnographic research.
  - Administrators must verify consent documentation before publishing any community media.
  - Cultural materials will be presented with appropriate attribution and context.

3.3 External Interfaces

3.3.1 User Interfaces

The Living Archives platform provides two main user-facing interfaces:

- **Public Interface (Web Portal):**
  - Accessible via web browser.
  - Features a home page with search bar, category filters, and featured archives.
  - Each media item has a detail page with description, metadata, and related items.
  - 3D tour viewer embedded within the page for immersive experiences.



Living Archives

A digital archive documenting Sindh's communities through multimedia and immersive 3D experiences.

Featured: Kolhi

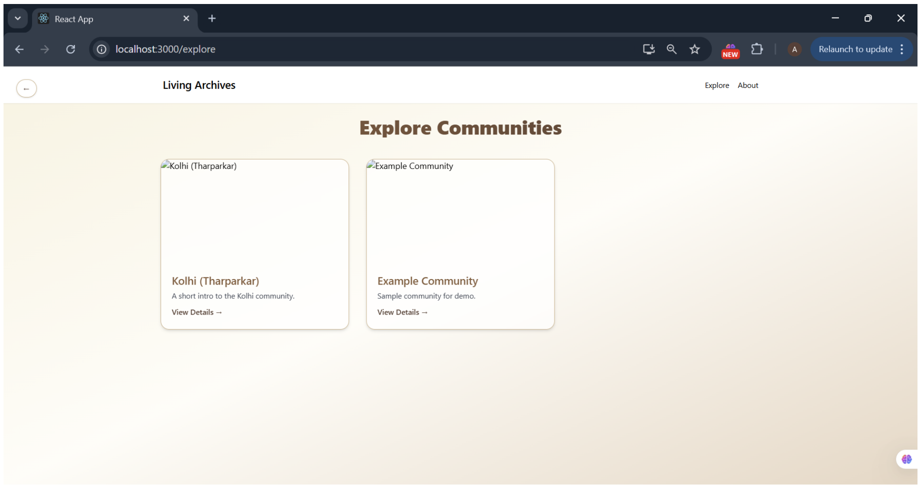
A proof-of-concept collection showcasing photos, oral histories, and a 3D virtual tour.

View Kolhi →

About the Project

Built with the MERN stack, this project supports multimedia formats, transcripts, and immersive 3D NeRF-based tours.

Learn more →



Explore Communities

Kolhi (Tharparkar)

Kolhi (Tharparkar)

A short intro to the Kolhi community.

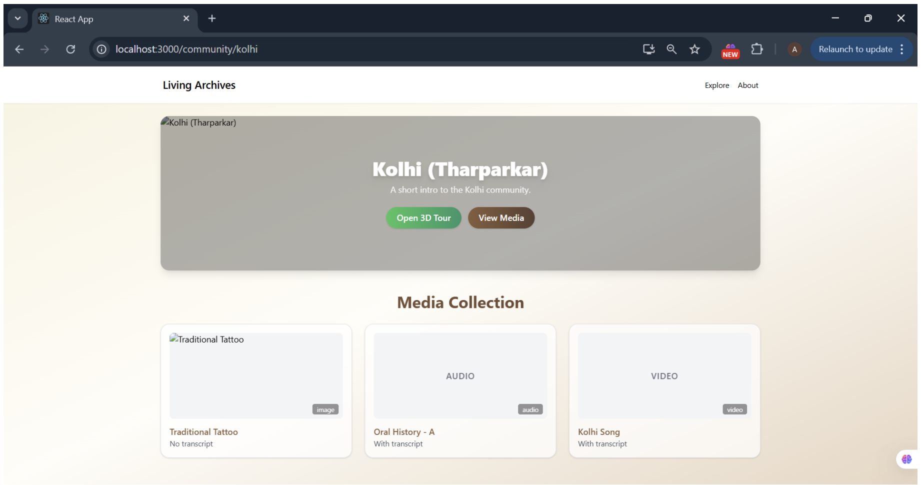
View Details →

Example Community

Example Community

Sample community for demo.

View Details →



Kolhi (Tharparkar)

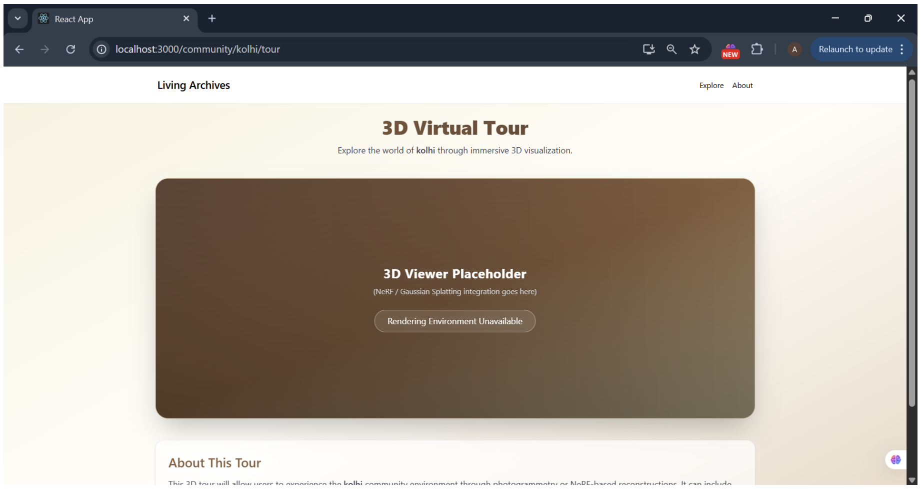
A short intro to the Kolhi community.

Media Collection

Traditional Tattoo
No transcript

Oral History - A
With transcript

Kolhi Song
With transcript


React App
localhost:3000/community/kolhi/tour
Relaunch to update
Living Archives
Explore
About
Rendering Environment Unavailable

3D Virtual Tour

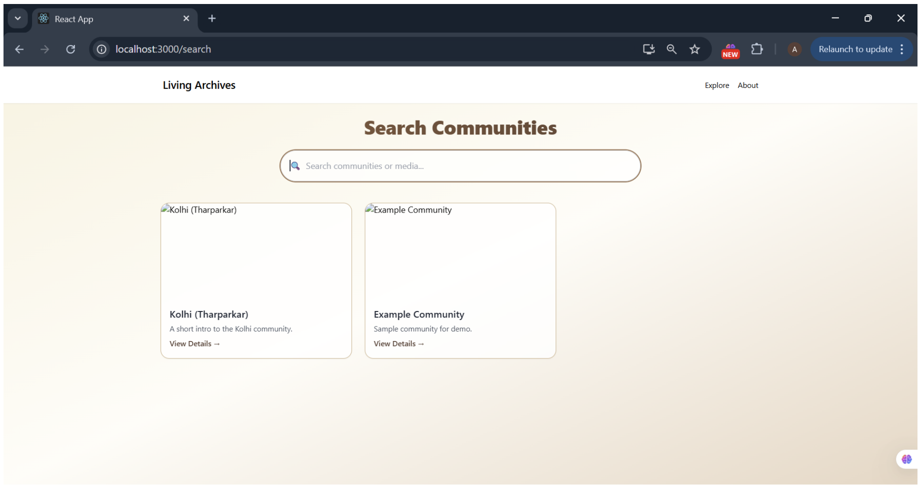
Explore the world of **kolhi** through immersive 3D visualization.

3D Viewer Placeholder

(NeRF / Gaussian Splatting integration goes here)

About This Tour

This 3D tour will allow users to experience the **kolhi** community environment through photogrammetry or NeRF-based reconstructions. It can include



Search Communities

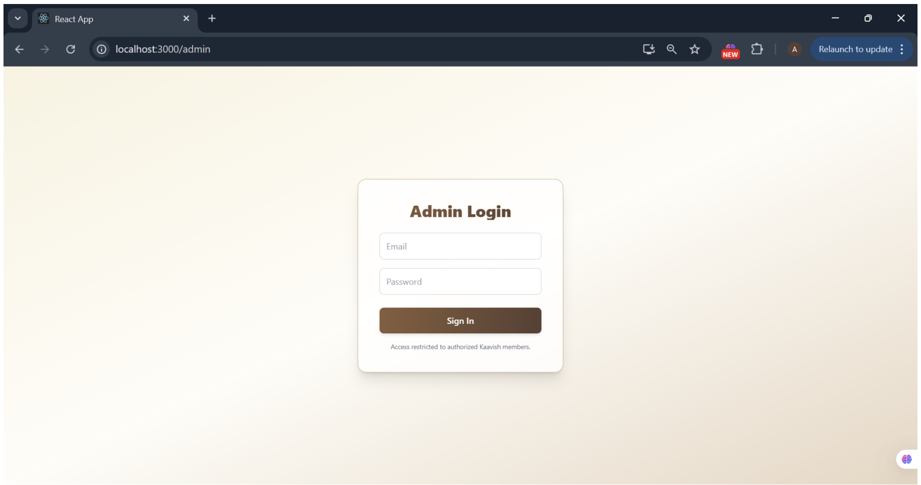
Kolhi (Tharparkar)

A short intro to the Kolhi community.

Example Community

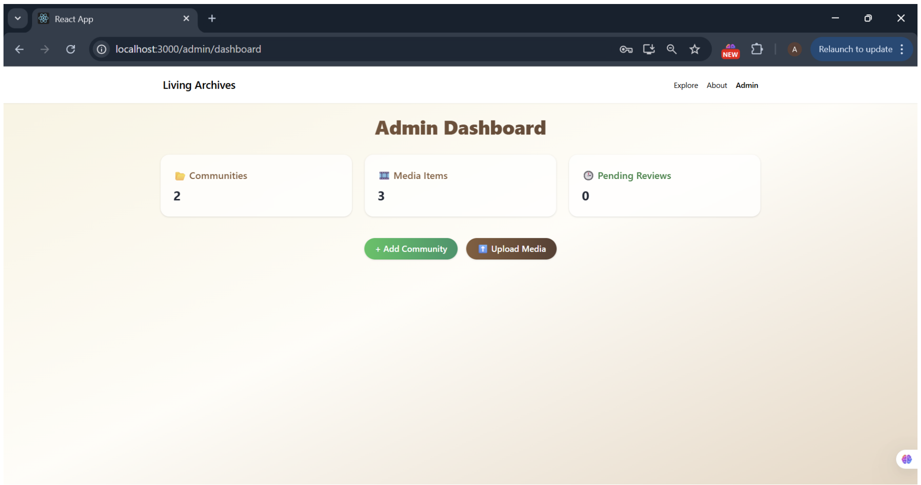
Sample community for demo.

- **Admin Interface:**
  - Accessible only to authorized users.
  - Includes upload form, metadata editor, user management, and backup controls.



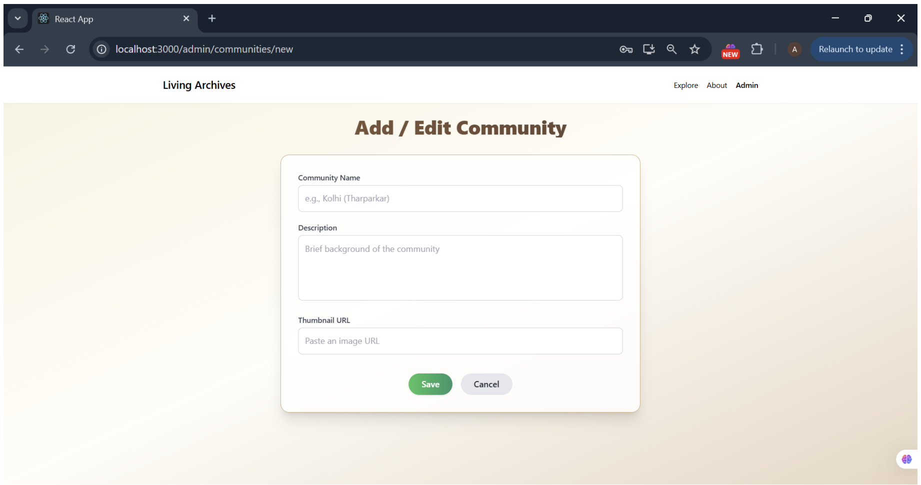
Admin Login

Access restricted to authorized Kaavish members.



Admin Dashboard

Communities	Media Items	Pending Reviews
2	3	0

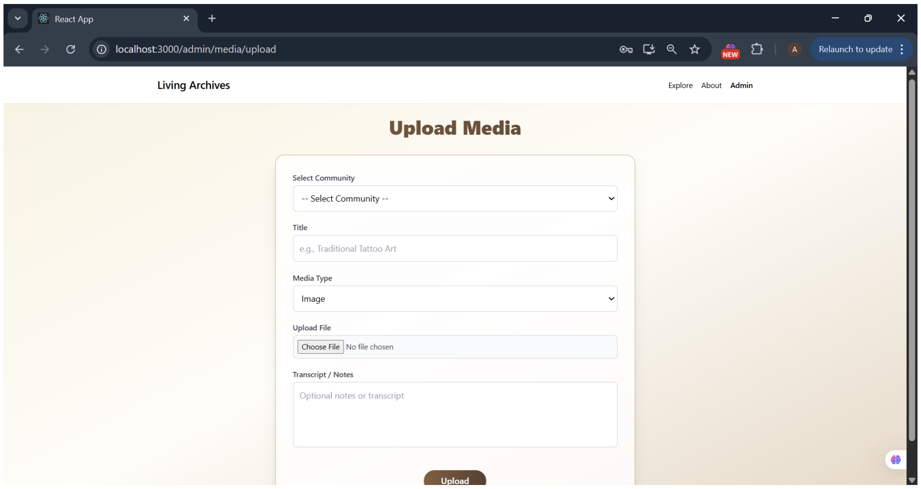


Add / Edit Community

Community Name: e.g., Kolhi (Tharparkar)

Description: Brief background of the community

Thumbnail URL: Paste an image URL



Upload Media

Select Community: -- Select Community --

Title: e.g., Traditional Tattoo Art

Media Type: Image

Upload File: Choose File No file chosen

Transcript / Notes: Optional notes or transcript

3.3.2 Hardware and Communication Interfaces

- The system operates on standard hardware; a mid-range GPU (e.g., NVIDIA RTX 3060 or equivalent) is recommended for 3D reconstruction tasks.
- Communication between frontend and backend via RESTful APIs over HTTPS.
- Cloud or on-premise deployment supported using standard web server configurations.

3.4 Use Cases

3.4.1 Use Case Diagram

Actors:

- **Public User** — browses, searches, and views content.
- **Researcher** — advanced search and data export (future extension).
- **Contributor** — uploads new content with metadata and consent information.
- **Administrator** — manages users, reviews uploads, edits or removes content, monitors system logs.

Primary Use Cases:

1. User logs in or registers an account.
2. User browses or searches the archive.
3. Contributor uploads new content with metadata and files.
4. System processes and stores uploaded files.
5. Admin reviews and publishes approved uploads.
6. User views detailed media pages or explores 3D scenes.
7. System performs automatic backup at scheduled intervals.

3.5 System Diagram

Components:

- **Frontend (React.js):** Provides interactive user interface for browsing, searching, and viewing content.
- **Backend (Node.js + Express):** Handles API requests, authentication, and routing between frontend and database.
- **Database (MongoDB):** Stores metadata, user records, and references to media files.
- **Media Storage:** Stores uploaded files (audio, video, images, 3D models) in cloud or local storage.
- **3D Reconstruction Engine:** Uses NeRF or Gaussian Splatting algorithms for generating interactive 3D environments.
- **Admin Console:** Provides management tools for reviewing uploads, moderating content, and system maintenance.

Data flow between components is REST-based: user actions on the frontend trigger API calls to the backend, which fetches or updates data in MongoDB and serves media content from storage. The architecture supports modular scaling and clean separation between data, logic, and presentation layers.

We expect every project to have at least of the following subsections. This section must be aligned with your project deliverables. Please consult with your project supervisor regarding which of the following section(s) you should include in your report

4. Software Design Specification (SDS)

This chapter provides important artifacts related to design of our project.

4.1 Software Design

This section presents the UML class diagram and gives a brief description of each class in our system. Attributes and methods of each class and relationship among classes are clearly presented.

4.2 Data Design

This section presents the structure of our database that caters to persistent data storage in our project. The structure is shown as a normalized data model for relational databases. It clearly shows entities, attributes, relationships with their cardinalities, and primary and foreign keys. We have used DB designer (or any other similar data modeling tool) to build our data model.

4.3 Technical Details

Our project does not have persistent data so we have no ERD. Instead we exaplin here the technical details of the algortihsm we use. These include the inputs and the outputs, how and where these algorothms fit in our tool chain, the techniques used in these algorithms, etc.

5. Methodology, Experiments and Results OR Testing and Analysis

We did many experiments and got the best results.

6. Conclusion and Future Work

Our work is awesome. We would write more but we need to catch the flight to collect our Turing Award.

7. Reflection

Learning Reflection: Learning experience, skills gained, and challenges overcome. Team Reflection (if applicable): Collaboration experience, role distribution, communication, and teamwork challenges. Project Reflection: Also a reflection on what was proposed vs what was achieved? Were any proposed features removed/were any features added that were not proposed initially? What are the reasons for these choices? What are opportunities and challenges led to these design decisions? Process Reflection: What worked well? What could be improved in the approach?

Appendix A. More Math

Here, we describe the background math for the techniques used in the text.

Appendix B. Data

Here is a dump of our 2TB data set. Enjoy!

Appendix C. Code

Here is our code.

```
print('Hello World!')
print('Computing true random number.')
print('Capturing interstellar radiation.')
print('This will take time!')
import random
import time
time.sleep(3600*random.randint(1,10))
print(4)
```

Our code can be found at `https://github.com/habib-university/Kaavish-Template`.

References

[1] Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: `http://dx.doi.org/10.1002/andp.19053221004`.

[2] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

[3] Donald Knuth. *Knuth: Computers and Typesetting*. 1984. URL: `http://www-cs-faculty.stanford.edu/~uno/abcde.html`.