Ralph Makofane

Past Paper Portal

How it works documentation

Ralph Makofane

Table of Contents

Project Background: Past Paper Portal

The shift towards digital education has accelerated the need for efficient, accessible, and user-friendly online platforms that support academic activities. Traditional methods of accessing and managing academic resources, such as past exam papers, are often large and time-consuming. By creating a centralized, digital portal, institutions can streamline the experience of accessing and managing essential academic resources, improving educational outcomes and easing administrative tasks.

The project, the Past Paper Portal, is designed to meet the needs of students, lecturers, and administrators within an academic institution by providing an organized, interactive platform for accessing past exam papers, managing academic resources, and enhancing academic engagement.

Project Introduction

The Past Paper Portal aims to serve as a comprehensive solution for students and faculty members by digitizing the archive of past examination papers, simplifying access, and providing management features. Students can conveniently search, view, and download question papers, while lecturers have access to upload, delete, and manage resources. Administrators oversee user permissions and ensure proper access levels across the platform.

This web application incorporates authentication to maintain security and leverages real-time data handling for efficient interaction. The design prioritizes user experience with an intuitive interface and responsive design across devices.

Motivation

The motivation for this project stems from the challenges encountered by both students and faculty in locating and managing past exam resources in physical archives. Often, students have limited access to past papers, and faculty members spend valuable time managing hard copies. Additionally, an increasing number of educational institutions are investing in online resources, which enhance accessibility, reduce paper usage, and align with modern digital learning trends.

Key motivations for developing this application include:

1. Enhanced Accessibility: To enable students to access academic resources anytime and from any location, supporting diverse learning needs and schedules.
2. Centralized Management: To provide faculty members with tools to efficiently manage and update academic resources, reducing the administrative burden and allowing for more focus on teaching and mentoring.
3. Sustainability: By creating a digital archive, this project contributes to the reduction of paper usage, aligning with sustainability goals in education.
4. Data Security: With sensitive academic information, the portal ensures secure access through authentication measures, protecting resources and user information.

5. Educational Improvement: By making past exam resources more accessible, students can better prepare for exams, leading to improved learning outcomes and higher overall academic performance.

How the system works:

Access at https://paperportalcode.web.app

1. User Signup and Login

1. Signup:
    - New users must register through the signup page by providing their email address, a secure password, and optionally some basic details like name and institution ID.
    - The system sends a verification email through Firebase Authentication. The user must verify their email before gaining access.
2. Login:
    - Returning users log in by entering their registered email and password.
    - Role assignment (student or lecturer) is determined during signup or managed by the admin.

2. Role Selection and Permissions

Role: Student

- Permissions:
    - View past question papers for their selected course and year.
    - Download papers for offline use.
    - Access only the courses and years assigned to their profile.
- How It Works:

1. After login, students are directed to the Student Dashboard.
2. They select their course (e.g., DIP ICT or B ICT) using buttons on the dashboard.
3. Once the course is selected, a dropdown allows them to choose their year of study (e.g., First Year, Second Year).
4. Selecting a year dynamically displays the available modules for that year.
5. Clicking on a module presents the list of available question papers for that module.
    - Each paper has options to preview or download.

Role: Lecturer

- Permissions:
    - Upload question papers to specific modules for the courses and years they are assigned to.
    - Delete uploaded question papers.
    - Preview and download existing question papers.
- How It Works:

1. After login, lecturers are directed to the Lecturer Dashboard.
2. Similar to students, lecturers select their course and year of study to view modules.
3. Each module offers the following functionalities:
    - Upload Papers:
        - Lecturers click the *Upload* button, which opens a drag-and-drop interface or file selector.
        - Once a file is uploaded, it is stored in Firebase Storage and associated metadata (e.g., filename, module, year) is saved in Firestore.
    - Preview Papers:
        - Clicking *Preview* opens a PDF viewer to display the paper directly in the browser.
    - Delete Papers:
        - Lecturers can delete papers from the system. This removes the file from Firebase Storage and its metadata from Firestore.

3. System Workflow

Shared Features (Students and Lecturers):

- Course and Year Selection:
    - Both roles start by selecting their course (DIP ICT or B ICT).
    - The year dropdown is dynamically populated only after selecting the course.
- Module Display:
    - The modules associated with the selected course and year are displayed.

- This data is fetched dynamically from the Firestore database based on the selected filters (course and year).

Student Interaction Workflow:

1. Choose a Course:
    - E.g., DIP ICT.
2. Select a Year:
    - E.g., Second Year.
3. View Modules:
    - Modules for the selected year are displayed.
4. Access Question Papers:
    - Clicking a module displays a list of question papers.
    - Students can either preview or download papers.

Lecturer Interaction Workflow:

1. Choose a Course:
    - E.g., DIP ICT.
2. Select a Year:
    - E.g., Third Year.
3. View Modules:
    - Modules for the selected year are displayed.
4. Upload Question Papers:
    - Clicking *Upload* opens an interface for uploading files.
    - Files are validated and uploaded to Firebase Storage, with metadata saved in Firestore.
5. Manage Question Papers:
    - Lecturers can:
        - Preview papers: Open in a viewer.
        - Delete papers: Remove from storage and database.
        - Download papers: Save locally for personal records.

4. Dynamic Access Based on Role

The system uses Firebase Authentication and Firestore rules to enforce role-based access:

1. Students:
    - Restricted to viewing and downloading papers.
    - Cannot upload or delete files.
    - Can only access papers for their assigned course and year.
2. Lecturers:
    - Granted permission to upload, delete, and manage papers for the modules they are responsible for.
    - Have access only to the courses and years assigned by the admin.
3. Admins:
    - Full access to manage users, permissions, courses, and papers.
    - Can add or remove modules, assign courses to lecturers, and manage all uploaded content.

5. Year-Based Access

- The system ensures that students and lecturers can only view or manage papers for their specific year of study.
- This is achieved by:
    - Linking question papers to their respective year in Firestore.
    - Using Firestore rules to restrict access based on the user's profile data.

Summary of Features:

Role	Actions
Student	View and download question papers.
Lecturer	Upload, preview, download, and delete question papers.

The system ensures a smooth and secure flow for each role, enhancing productivity and simplifying academic resource management.

Overall Description (System Interfaces)

The Past Paper Portal interfaces with various system components to ensure a seamless experience for users (students, lecturers, administrators). Below are the key interfaces:

1. User Interface (UI):

o Provides an intuitive front end for students, lecturers, and administrators.

o Includes responsive designs compatible across desktop, tablet, and mobile devices.

o Students can easily access past papers, lecturers can manage uploads, and administrators can control access permissions.

2. Authentication Interface:

o Integrates with Firebase Authentication to securely manage user logins, registration, and role-based access.

o Ensures that students, lecturers, and administrators are authenticated and assigned appropriate permissions upon login.

3. File Management and Storage Interface:

o Provides an interface for uploading, storing, and retrieving past paper files.

o Uses cloud storage to store academic resources securely.

o Ensures data integrity and efficient retrieval processes, enabling students to download papers without delay.

4. Database Interface:

o Connects with a cloud-hosted database for managing metadata associated with past papers, users, and access permissions.

o Supports CRUD operations for adding, viewing, updating, and deleting records as per user roles.

o Ensures that records are organized and retrievable based on filters like course, year, and module.

5. Admin Control Panel Interface:

o Accessible only by administrators, this panel provides tools for managing user roles, permissions, and maintaining database records.

o Includes options to monitor storage usage, manage files, and handle access control settings for enhanced security.

Technical Process (Methods, Tools, and Techniques)

The technical development of the Past Paper Portal involves several key tools, techniques, and methodologies, focusing on user-centered design, modular development, and secure coding practices.

1. Frontend Development:

o Tools: HTML, CSS, JavaScript, and possibly frameworks like React.js for dynamic interfaces.

o Techniques: Responsive design to ensure a seamless experience across various devices; modular JavaScript to separate concerns by creating reusable components (e.g., buttons, dropdowns, and form elements).

2. Backend Development:

o Tools: Node.js, Firebase Authentication, Firebase Firestore (for database needs), and Firebase Storage (for file storage).

o Techniques: RESTful API design to manage data retrieval and storage operations, modular coding practices for separating authentication, database, and storage logic.

3. Database Management:

o Tools: Firebase Firestore or similar NoSQL databases to store user data, permissions, and metadata for past papers.

o Techniques: Data structuring using collections and documents within Firestore to organize data efficiently by course, year, and module.

4. Security Measures:

o Tools: Firebase Authentication for secure user management, SSL encryption to protect data transmitted between client and server.

o Techniques: Implementing role-based access control, sanitizing inputs to prevent SQL injection, and utilizing HTTPS protocols for secure communication.

5. Version Control:

o Tools: Git and GitHub for code versioning and collaboration.

o Techniques: Regular commits and branch management to keep track of development progress, allowing rollback if any issues arise.

6. Testing and Debugging:

o Tools: Testing libraries like Jest for JavaScript unit testing, Firebase Emulator for local testing.

o Techniques: Unit tests for validating individual functions, integration testing for ensuring components work well together, and user acceptance testing (UAT) to confirm that the application meets user expectations.

Design and Implementation Constraints

The project is subject to certain constraints that will guide the design and implementation phases:

1. Data Privacy and Security:

o Must comply with data protection regulations (e.g., GDPR) by safeguarding student data, including personal information and access permissions.

o Role-based access control is essential, limiting access to sensitive resources based on user role (student, lecturer, or admin).

o Data encryption is required for secure file storage and transmission.

2. Scalability:

o The system should support a growing user base and increasing data volume over time.

o The chosen backend architecture (Firebase) should allow for horizontal scaling as the number of users and files grows.

3. Cross-Platform Compatibility:

o The portal must be accessible on multiple devices and browsers (e.g., Chrome, Firefox, Safari).

o Responsive design principles must be applied to ensure the application adapts seamlessly to different screen sizes.

4. Real-Time Data Synchronization:

o Real-time access to resources is required, especially for student downloads and lecturer uploads.

o Firebase's real-time database features will be leveraged to update users immediately upon any data changes.

5. Performance and Load Handling:

o The application should handle multiple concurrent users without latency issues.

o Efficient data retrieval strategies and caching mechanisms are necessary to ensure that users do not experience long load times.

6. Storage and Cost Constraints:

o Firebase's free tier will be used initially, but storage limits must be monitored.

o As the project scales, there may be a need for paid cloud storage and hosting solutions, which could impact budget considerations.

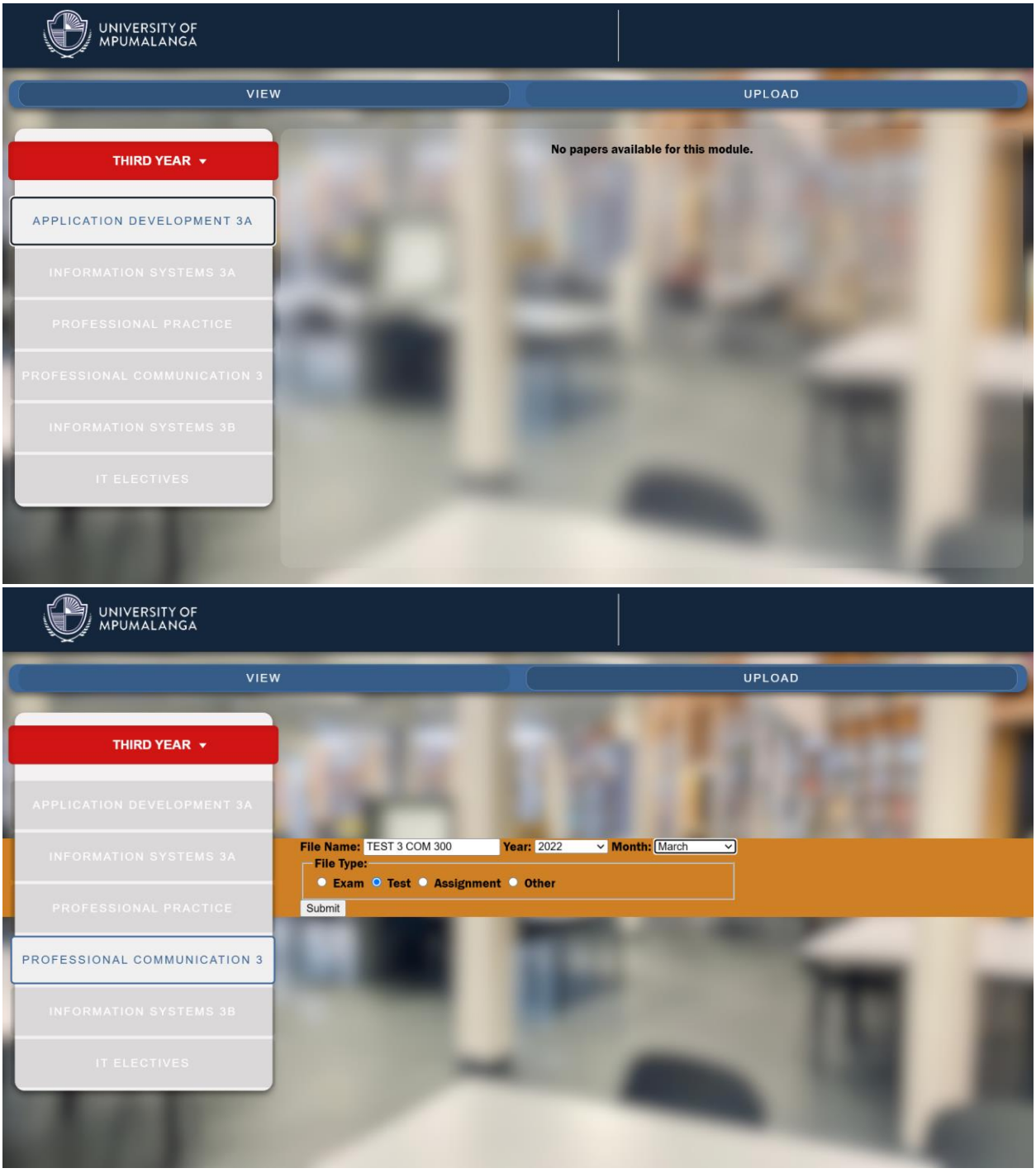
7. Maintenance and Support:

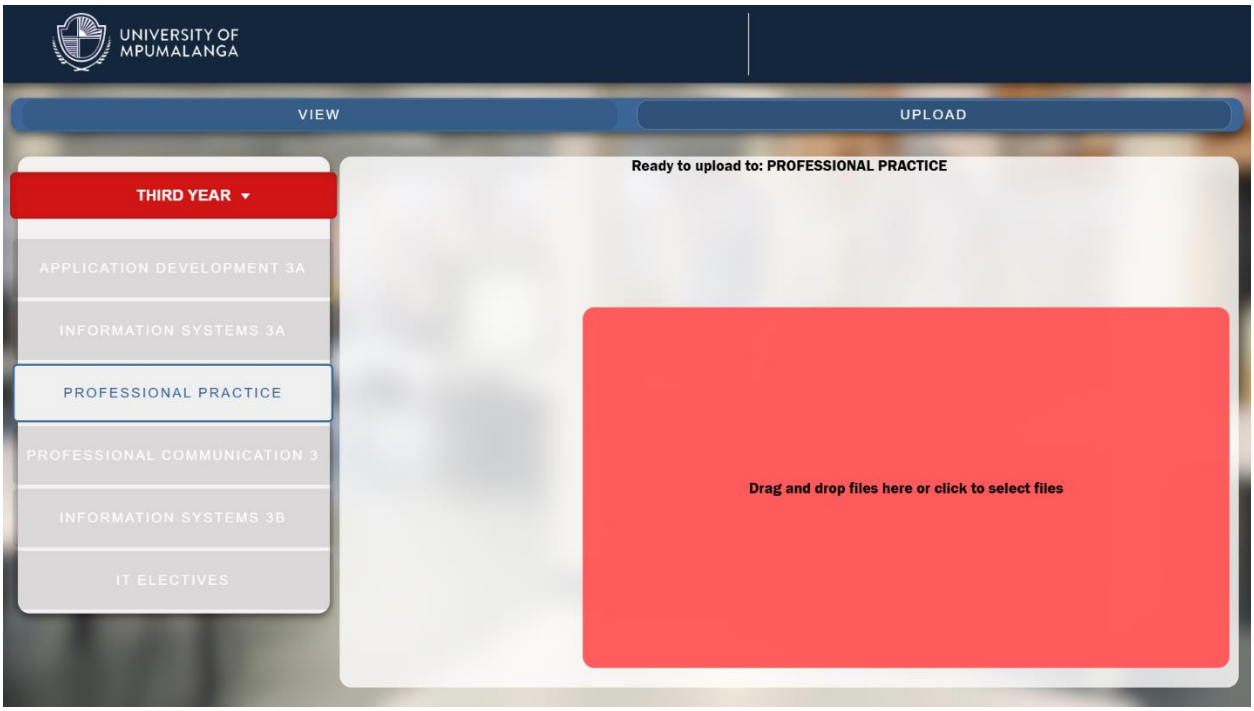
o The application requires regular maintenance to apply security patches, update dependencies, and monitor Firebase service changes.

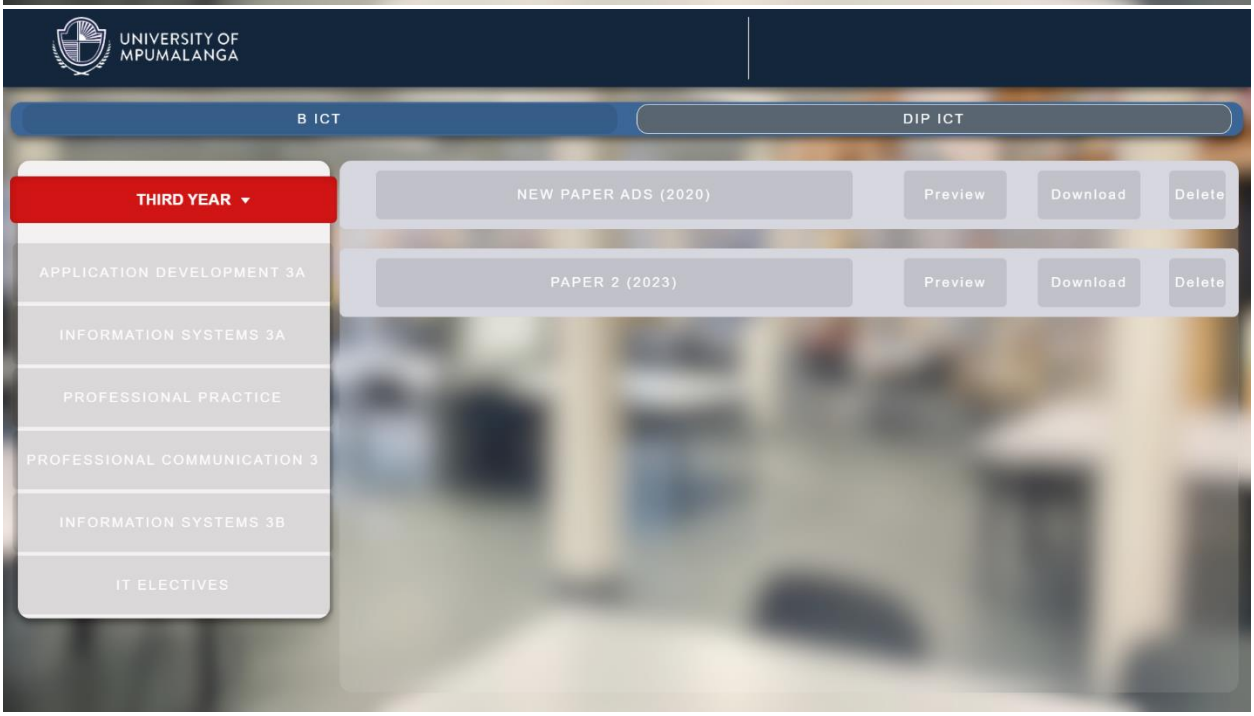
o Detailed documentation is necessary to support future development and updates, making it easier for administrators to manage changes over time.

Conclusion

The Past Paper Portal is more than just a digital repository—it is a tool for academic empowerment. By creating a secure, efficient, and user-friendly platform, this project aims to bridge the gap between students' resource needs and the institution's capacity to provide organized and accessible academic materials. This project also aligns with broader trends toward digital transformation in education, preparing institutions and students alike for a future of learning that is increasingly digital and data-driven.




UNIVERSITY OF MPUMALANGA
VIEW
UPLOAD
THIRD YEAR
APPLICATION DEVELOPMENT 3A
INFORMATION SYSTEMS 3A
PROFESSIONAL PRACTICE
PROFESSIONAL COMMUNICATION 3
INFORMATION SYSTEMS 3B
IT ELECTIVES
Ready to upload to: PROFESSIONAL PRACTICE
Drag and drop files here or click to select files


UNIVERSITY OF MPUMALANGA
B ICT
DIP ICT
THIRD YEAR
APPLICATION DEVELOPMENT 3A
INFORMATION SYSTEMS 3A
PROFESSIONAL PRACTICE
PROFESSIONAL COMMUNICATION 3
INFORMATION SYSTEMS 3B
IT ELECTIVES
NEW PAPER ADS (2020)
Preview
Download
Delete
PAPER 2 (2023)
Preview
Download
Delete

Lecturer Portal

Sign Up

Full Name

Email

Password

Confirm Password

Sign Up

Already have an account? Sign In

Welcome to Academic Portal

Lecturer Icon

Lecturer

Access course management and resource upload tools

Student Icon

Student

Access course materials and resources