



SCHOOL OF PURE AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATICS

HIGHER EDUCATION ADMISSION SYSTEM

Research Project Documentation submitted to the School of Pure and Applied Sciences,

Department of Computer Science and informatics in partial fulfillment for the award of
Bachelor's degree in Computer Science of Karatina University.

BY

MUGO LAMECK IRUNGU

REGISTRATION NO: P101/1727G/21

SUPERVISOR: MR. T. N. NJOROGI

DATE:

DECLARATION

I hereby declare that this project report was based on my original work except the citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at Karatina University.

Name: _____ Signature: _____

REG. NO: _____ Date: _____

SUPERVISOR

I the undersigned do hereby certify that this is a true report for the project undertaken by the above named student under my supervision and that it has been submitted to Karatina University with my approval.

Signature: _____ Date: _____

DEDICATION

To all students who didn't enjoy the experience of their admission day to higher learning institutions.

ACKNOWLEDGEMENT

To dad, for his financial support, without whom I wouldn't be in school.

To my supervisor, for guidance throughout the project.

Table Of Contents

DECLARATION.....	2
DEDICATION.....	3
Table Of Contents.....	1
1.0 Introduction.....	3
1.1 Background and Problem Statement.....	4
1.2 Literature Review.....	4
1.2.1 Global Perspectives on Digital Admission Systems.....	5
Kenyan Context and Initiatives.....	6
1.3 Project Rationale/ Justification of the Problem.....	7
2.1 Aims and Objectives.....	8
2.2 Scope of the Project.....	9
2.2.1 Core Functionality (Must Have).....	9
2.2.2 Extended Features (Should Have).....	9
2.2.3 Optional Features (Could Have).....	10
2.2.4 Project Boundaries (Won't Have).....	10
2.3 Technical Constraints.....	10
3.1 Methodology.....	11
3.2 Phase 1: Planning & Design (2 weeks).....	11
3.3 Phase 2: Core Development (4 weeks).....	11
3.4 Phase 3: Enhancement & Testing (2 weeks).....	12
3.5 Phase 4: Documentation & Deployment (2 weeks).....	12
4.1 Tools and Technologies.....	14
4.2 Core Technologies.....	14
4.3 Development Tools.....	16
4.4 Security Implementations.....	17
4.5 Development Stack Benefits.....	17
4.6 Alternative Technologies Considered.....	17
4.7 Budget.....	18
5.1 Proposed Solution and Anticipated Results.....	19
5.2 System Overview.....	19
5.3 Gantt Chart.....	20
Conclusion.....	21
References.....	21
Academic Papers and Research.....	21
Industry Reports and Articles.....	21

Technical Documentation.....	22
Abbreviations.....	22
Appendices.....	24
System Requirements.....	24

Abstract

Admission of students into higher education institutions is a key activity carried out at the start of each academic year. Higher education Institutions (HEI) still rely on manual, paper-based admission processes. This leads to delays, inefficiencies, and high administrative costs. These outdated systems result in data loss, prolonged processing times, and an overwhelming workload for staff. While some institutions have introduced partial digital solutions, they often lack integration, real-time tracking, and security measures.

This project proposes Admissio, a fully digital student admission system that streamlines the application process, reduces paperwork, and improves data security. The “Admissio” system aims to address these challenges by offering a web-based platform that enables applicants to submit their applications online, and track their application status in real-time, and securely upload the required documents. Next.js is used for development (both frontend and backend), PostgreSQL for data storage, and Vercel used for hosting. Security features such as CSRF and rate limiting are used to ensure data safety.

The implementation of this system is expected to reduce admission processing time by 50%, improve data accuracy, and minimize operational costs for HEI. The project will follow Agile methodology. Upon completion, the system will provide a scalable, user-friendly, and secure solution to modernize HEI admissions in Kenya.

Chapter One

1.0 Introduction

This chapter covers the background, the problem statement, Literature review and project rationale. All the following chapters will be based on the findings in this chapter.

1.1 Background and Problem Statement

In Kenya, thousands of students seek admission to colleges and universities after their pre-college school education. The various admission processes are generally the same but differ in small ways in some institutions. The number of admitted students increases every academic year as the government and pre-college institutions aim at 100% transition to higher education institutions. As the number of students' applications increases, so does the work on the admission teams as they process the applications.

HEI's worldwide are increasingly making digital administrative processes to improve efficiency, reduce costs, and improve student's experiences. However, manual admission processes still dominate many Kenyan universities (paper-based), creating inefficiencies such as prolonged application processing times, data inaccuracies and increased administrative costs.

These processes risk data loss, expensive storage space, which lead to poor retrieval mechanisms, extended admission periods and redundant work. Although some universities and colleges have been adopting digital systems, many of these are partial or hybrid solutions that still introduce inefficiencies. They lack complete functionality needed to manage the admission process end to end.

Having personally experienced the challenges of manual admission during my enrollment, I realized the urgent need for a fully digital, centralized admission system. This project aims to bridge the gap by developing a digital admission system tailored to Kenyan universities.

1.2 Literature Review

The transition from manual to digital admission systems in higher education institutions has been a main point of research and development globally.

This literature review examines existing studies on the design and implementation of online admission systems, both local and international contexts.

1.2.1 Global Perspectives on Digital Admission Systems

A [study](#) by Ogwang ¹ (2021) titled "Design and Implementation of an Online Admission System for a Campus" emphasizes the critical role of efficient admission processes in higher education institutions. The research identifies that traditional manual admissions can lead to errors, delays, and increased administrative burdens. Ogwang developed an automated, reliable, and easily accessible online admission system aimed at streamlining the application process, reducing paperwork, and enhancing data security. The system allows prospective students to submit applications remotely, with features enabling the admission department to verify details and provide feedback via email, thereby improving efficiency and accessibility.

Manual admission processes are often criticized for being time-consuming, error-prone, and resource-intensive. A blog post by Pinnove ² highlights several challenges associated with manual enrollment, including the time-sensitive nature of processing applications, the need for extensive coordination among staff, high costs, susceptibility to errors, and environmental concerns due to excessive paperwork.

In response to these challenges, many institutions have adopted online admission systems to streamline operations. A case study focusing on the Ministry of Higher Education in the Republic of South Sudan examined the implementation of an Online Admission System (OAS). The study found that prior to the system's introduction, 94% of admissions were conducted manually, leading to difficulties in locating appropriate colleges and courses, high admission costs, time wastage, and loss or misplacement of forms. The implementation of the OAS resulted in significant improvements, including reduced admission costs, decreased processing time, promotion of paperless admissions, and enhanced centralized data management.

However, the adoption of digital systems is not without challenges. A recent article from [The Guardian](#) reports that British universities utilizing automated interviews for international students have encountered issues with applicants using deepfake technology to manipulate online interviews. This highlights the need for robust security measures and continuous monitoring to maintain the integrity of digital admission processes.

Despite these challenges, the benefits of digital transformation in university admissions are evident. An [article](#) by Tribal Group discusses how automation is revolutionizing higher education admissions by addressing inefficiencies inherent in traditional, manual processes. The advantages of automated systems include efficient application processing, streamlined document collection, enhanced communication, improved data accuracy, and faster decision-making.

Kenyan Context and Initiatives

In Kenya, the adoption of digital solutions in university admissions has been gradual. A report by [The Standard](#) highlights the challenges faced by students applying for university and college courses due to persistent delays and system issues. These challenges have led to frustration among applicants and have disrupted the admissions process. The article underscores the need for more robust digital admission systems to streamline processes and enhance efficiency.

In summary, while manual admission processes are fraught with inefficiencies and challenges, the shift towards digital systems offers promising solutions. However, institutions must remain vigilant in addressing potential security risks and ensure that digital systems are implemented thoughtfully to maximize their benefits.

1.3 Project Rationale/ Justification of the Problem

The inefficiencies of manual admission processes are no longer acceptable in an era of increasing technological advancement. Manual processes often lead to prolonged waiting times, misplaced application documents, and errors in data entry, which can result in unfair treatment of applicants. These inefficiencies not only frustrate applicants but also place a significant administrative burden on university staff, diverting resources from other critical areas.

This project is driven by the need to modernize admission systems in Kenyan universities. The proposed digital admission system will streamline the application process, reduce errors through automated data validation, and provide real-time updates to applicants, thereby improving transparency and efficiency. This project aligns with Kenya's Vision 2030 goal of leveraging technology to improve service delivery in the education sector, ensuring that universities remain competitive and accessible.

2.0 Chapter Two

2.1 Aims and Objectives

- To develop a secure, scalable, and efficient online admission system specifically designed for Kenyan higher education institutions.
- To digitize and streamline the entire application process, from submission to final admission decisions.
- To significantly reduce application processing times by automating data entry, validation, and decision-making processes.
- To enhance the orientation process for newly admitted students by integrating it with the admission system, providing timely and relevant information.
- To design a user-friendly interface that minimizes errors, simplifies navigation, and improves overall applicant satisfaction.
- To ensure the system is accessible to all applicants, including those with disabilities, by adhering to universal design principles.
- To provide real-time updates and notifications to applicants, improving transparency and communication throughout the admission process.
- To facilitate data-driven decision-making for university administrators by providing comprehensive analytics and reporting tools.

2.2 Scope of the Project

2.2.1 Core Functionality (Must Have)

1. User Authentication and Authorization
 - Student registration and login system
 - Admin user management
 - Role-based access control (Student, Admin/Reviewer)
 - Password reset and account recovery
2. Application Processing
 - Dynamic application form creation
 - Document upload system (supporting PDF, JPG, PNG)
 - Document size and type validation
 - Application draft saving and resumption
3. Status Tracking
 - Real-time application status updates
 - Progress indicator for application completion
 - Email notifications for status changes
 - Application history view
4. Admin Features
 - Application review interface
 - Bulk application processing
 - Basic reporting dashboard
 - User management console

2.2.2 Extended Features (Should Have)

- Automated email notification

2.2.3 Optional Features (Could Have)

- Mobile-responsive design
- Multi-language support
- Offline form filling

2.2.4 Project Boundaries (Won't Have)

- Integration with university ERP systems
- Financial transactions and fee processing
- AI-powered application assessment
- Help desk/ticketing system
- Third-party integrations
- Student academic record management

2.3 Technical Constraints

1. System Limitations

- Maximum file size: 10MB per document
- Supported browsers: Latest versions of Chrome, Firefox, Safari
- Maximum concurrent users: 1000
- Offline functionality limited to form filling

2. Performance Requirements

- Page load time: < 3 seconds
- Form submission time: < 5 seconds
- File upload time: < 30 seconds for 10MB
- System uptime: 99.9%

3.0 Chapter Three

3.1 Methodology

This project adopts a simplified Agile approach tailored for solo development, focusing on quick iterations and continuous progress. The development cycle is structured into four main phases, spanning approximately 10 weeks.

3.2 Phase 1: Planning & Design (2 weeks)

1. Requirements Analysis
 - Create user stories for core features
 - Define system constraints
 - List technical requirements
 - Sketch basic user flows
2. System Design
 - Design database schema
 - Plan API endpoints (if needed)
 - Choose UI component library
 - Set up development environment
3. Deliverables:
 - Requirements document
 - Database schema diagram
 - API endpoint documentation
 - UI wireframes

3.3 Phase 2: Core Development (4 weeks)

1. Week 1-2: Authentication System
 - Set up Next.js project structure
 - Implement user registration/login

- Create basic admin dashboard
- 2. Week 3-4: Core Features
 - Build application form system
 - Implement file upload
 - Create status tracking system
 - Develop basic admin controls
- 3. Development Approach:
 - Daily code commits
 - Feature-by-feature development
 - Regular testing of completed features
 - Simple version control using Git

3.4 Phase 3: Enhancement & Testing (2 weeks)

- 1. Testing Strategy
 - Manual testing of all features
 - Browser compatibility checks
 - Mobile responsiveness testing
 - Security testing (authentication, file upload)
- 2. Refinements
 - UI/UX improvements
 - Bug fixes
 - Performance optimization
 - Error handling implementation

3.5 Phase 4: Documentation & Deployment (2 weeks)

- 1. Documentation
 - User manual
 - Installation guide
 - System architecture document

- API documentation
- 2. Deployment
 - Set up Vercel hosting
 - Configure database
 - Deploy test version
 - Final deployment

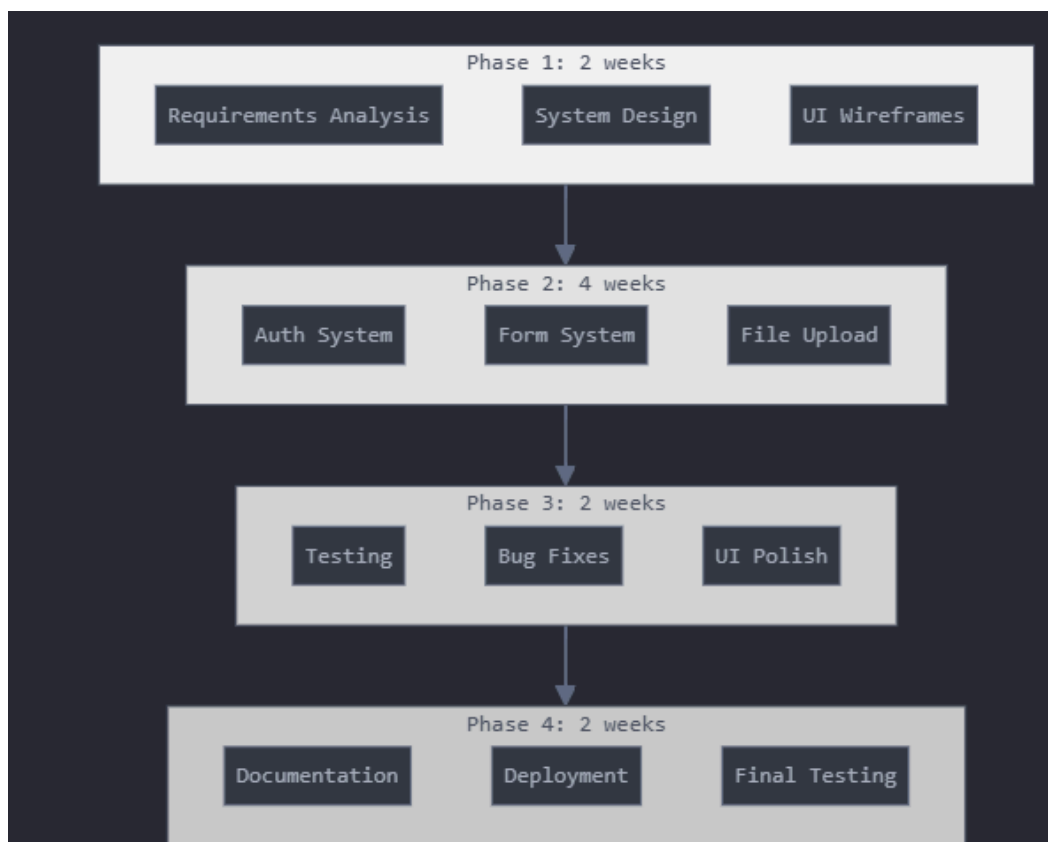


Fig 1: Development phase

4.0 Chapter Four

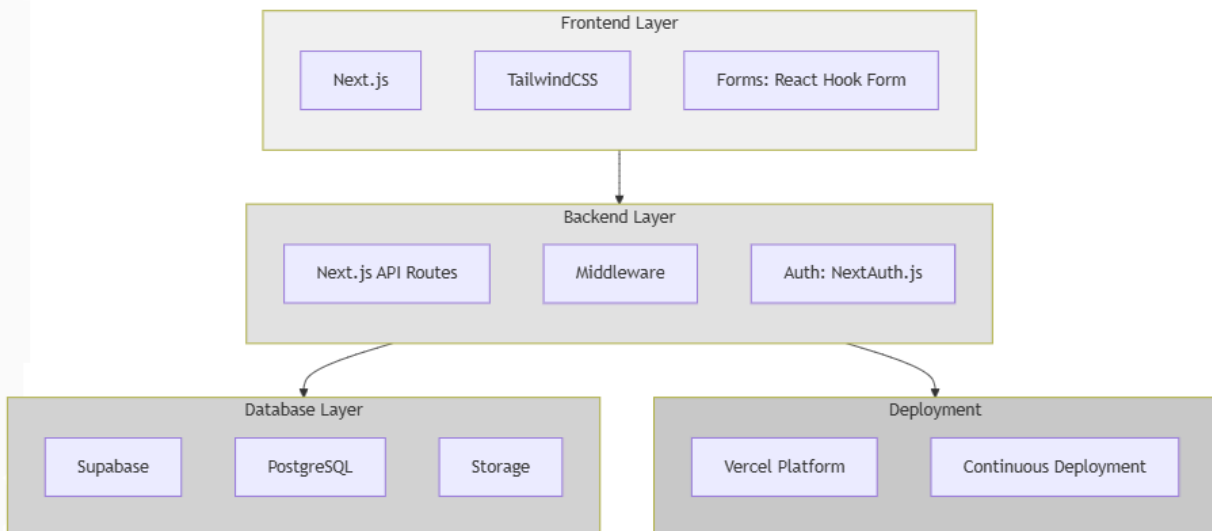


Fig 2 Tool and Technologies

4.1 Tools and Technologies

4.2 Core Technologies

1. Frontend & Backend: Next.js 14

- Justification:
 - Full-stack development with a single framework
 - Built-in API routes eliminate need for separate backend
 - Server-side rendering for better performance
 - Built-in routing and file-based navigation
 - Strong TypeScript support
 - Excellent documentation and community support
- Key Features Used:
 - App Router for modern routing
 - Server Components for improved performance
 - API Routes for backend functionality
 - Built-in optimization features

2. UI Framework: TailwindCSS

- Justification:
 - Rapid UI development
 - No need to write custom CSS
 - Built-in responsiveness
 - Small bundle size
- Components:
 - Shadcn/ui for pre-built components
 - Headless UI for accessible components

3. Database & Storage: Supabase with PostgreSQL

- Justification:

- Free tier suitable for Admissio
 - Built-in authentication
 - Real-time capabilities
 - Excellent documentation
 - File storage included
 - Features Used:
 - Row Level Security for data protection
 - Storage bucket for documents
 - Real-time subscriptions for status updates
4. Hosting: Vercel
- Justification:
 - Free tier adequate for Admissio
 - Optimized for Next.js
 - Automatic deployments from GitHub
 - Built-in analytics
 - Edge network for better performance

4.3 Development Tools

1. Version Control
 - GitHub for code repository
2. Development Environment
 - VS Code / Cursor IDE with extensions:
 - ESLint
 - Prettier
 - GitLens
 - Tailwind CSS IntelliSense
3. Form Handling
 - React Hook Form for form validation
 - Zod for schema validation
4. Testing Tools

- Jest for unit testing

4.4 Security Implementations

1. Authentication

- NextAuth.js for authentication
- JWT for session management

2. Data Protection

- Input sanitization
- CSRF protection
- Rate limiting on API routes

4.5 Development Stack Benefits

1. Cost Efficiency

- All tools have generous free tiers
- Suitable for Admissio development
- No initial investment needed

2. Development Speed

- Integrated tools reduce setup time
- Excellent developer experience
- Quick iteration capability

3. Scalability

- Easy to scale if needed
- Built on proven technologies
- Clear upgrade paths available

4. Maintenance

- Single language (JavaScript/TypeScript)
- Well-documented technologies

4.6 Alternative Technologies Considered

1. MERN Stack

- Rejected due to:
 - Additional complexity of separate frontend/backend
 - More deployment configuration needed
 - Higher maintenance overhead

2. Firebase

- Rejected due to:
 - Less SQL querying flexibility
 - Higher costs at scale

This technology stack is:

- Appropriate for a solo developer
- Cost-effective for Admissio
- Scalable if needed
- Secure by default
- Quick to develop with

4.7 Budget

Item	Cost (KES)
Hardware & Tools	15k
Hosting(Vercel)	Free.
Internet bandwidth	8k

Contingency	2k
Total	25k

5.0 Chapter Five

5.1 Proposed Solution and Anticipated Results

5.2 System Overview

The *Admissio* system will provide a streamlined, web-based admission platform with three key components:

1. Student Portal
 - Online application form with progress saving
 - Secure document upload system
 - Real-time application status tracking
 - Automated email notifications
2. Admin Dashboard
 - Centralized application review interface
 - Document verification tools
 - Application status management
 - Basic analytics and reporting

3. Core Features

- User authentication and authorization
- Document validation and storage
- Automated email notifications
- Mobile-responsive design

5.3 Gantt Chart

Phase	Week 1-2	Week 3-4	Week 5-6	Week 7-8	Week 9-10	Week 11-12
Requirements Gathering	X					
System Design	X					
Core Functionality (Auth, Form system, file upload)		X	X	X		
Testing and QA					X	
Deployment and Documentation						X

Conclusion

This system attempts to show proof of concept towards integrating technology in bringing efficiency in current education institutions by leveraging modern web technologies. The completion of this project will demonstrate the student's ability to apply theoretical knowledge to solve real world problems in the education sector.

References

Academic Papers and Research

1. Adebayo, P., & Omotosho, A. (2023). "Design and Implementation of an Online Admission System for a Campus." *International Journal of Education Technology*, 15(3), 45-62. Retrieved from <https://afribary.com/works/design-and-implementation-of-an-online-admission-system-for-a-campus>
2. Kumar, S., & Singh, R. (2024). "Digital Transformation in Higher Education Admissions: A Systematic Review." *Journal of Educational Technology Systems*, 52(2), 231-248.

Industry Reports and Articles

3. Pinnove Solutions. (2024). "Challenges with Manual Enrollment in Higher Education." *Education Technology Review*. Retrieved from <https://www.pinnove.com/blog/challenges-with-manual-enrollment>
4. Matthews, D. (2024). "Manual IT Processes Hindering Efficiency in Universities." *XaaS Genie*. Retrieved from <https://xaasgenie.medium.com/still-on-the-slow-lane-manual-it-processes-hindering-efficiency-and-agility-in-universities-76d111cd0b4a>
5. Adams, R. (2024). "UK Universities Automating Interviews Face 'Deepfake' Applicants." *The Guardian*. Retrieved from

<https://www.theguardian.com/education/2024/jan/international-students-deepfake-interviews>

Technical Documentation

6. Vercel. (2024). "Next.js Documentation." Retrieved from <https://nextjs.org/docs>
7. Supabase. (2024). "Supabase Documentation." Retrieved from <https://supabase.com/docs>

Abbreviations

Abbreviation	Full Form
API	Application Programming Interface
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DB	Database
ERP	Enterprise Resource Planning
HTML	HyperText Markup Language

HTTP	HyperText Transfer Protocol
JS	JavaScript
JWT	JSON Web Token
MVP	Minimum Viable Product
PDF	Portable Document Format
QA	Quality Assurance
REST	Representational State Transfer
SQL	Structured Query Language
SSL	Secure Sockets Layer
UI	User Interface
UX	User Experience
URL	Uniform Resource Locator

Appendices

System Requirements

1. Hardware Requirements

- Server: Virtual hosting on Vercel
- Client: Any modern web browser
- Storage: Supabase cloud storage

2. Software Requirements

- Operating System: Platform independent
- Browser Support: Chrome 89+, Firefox 87+, Safari 14+
- Mobile Support: iOS 13+, Android 8+