

A
Project Report
On
Intranet Based Examination System

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science and Engineering

By

Gaurav Joshi 2261219

Harsh Goswami 2261242

Rahul Koranga 2261454

Divyashish Negi 2261199

Under the Guidance of

Mr. Anubhav Bewerwal

ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O.BHOWALI,

DISTRICT – NAINITAL-263132

2024-2025

STUDENT'S DECLARATION

We, **Gaurav Joshi , Rahul Koranga , Harsh Goswami and Divyashish Negi** hereby declare the work, which is being presented in the project, entitled ‘ **Intranet_Based_Examination_System** ’ in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology (B.Tech.)** in the session **2024-2025**, is an authentic record of my work carried out under the supervision of Mr. Anubhav Bewerwal.

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date:

Gaurav Joshi

Rahul Koranga

Divyashish Negi

Harsh Goswami

CERTIFICATE

The project report entitled “Intranet_Based_Examination_System” being submitted by Gaurav Joshi S/o Mr. Bhuwan Chandra Joshi, 2261219 , Harsh Goswami S/o Mr. Sanjay Giri, 2261242 , Divyashish Negi S/o Mr. Mohan Singh Negi, 2261199 , Rahul Koranga S/o Harendra Singh Koranga, 2261454 of B.Tech.(CSE) to Graphic Era Hill University Bhimtal Campus for the award of bonafide work carried out by them. They have worked under my guidance and supervision and fulfilled the requirement for the submission of a report.

Mr. Anubhav Bewerwal

(Project Guide)

Dr. Ankur Singh Bisht

(Head, CSE)

ACKNOWLEDGEMENT

We take immense pleasure in thanking the Honorable Director '**Prof. (Col.) Anil Nair (Retd.)**', GEHU Bhimtal Campus to permit me and carry out this project work with his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance, and useful suggestions that helped me to develop as a creative researcher and complete the research work, in time.

Words are inadequate in offering my thanks to GOD for providing me with everything that we need. We again want to extend thanks to our president '**Prof. (Dr.) Kamal Ghanshala**' for providing us with all infrastructure and facilities to work in need without which this work could not be possible.

Many thanks to '**Dr. Ankur Singh Bisht**' (Head, Department of Computer Science and Engineering, GEHU Bhimtal Campus), our project guide '**Mr Anubhav Bewerwal**' (Assistant Professor, Department of Computer Science and Engineering, GEHU Bhimtal Campus) and other faculties for their insightful comments, constructive suggestions, valuable advice, and time in reviewing this report.

Finally, yet importantly, We would like to express my heartiest thanks to our beloved parents, for their moral support, affection, and blessings. We would also like to pay our sincere thanks to all my friends and well-wishers for their help and wishes for the successful completion of this project.

Gaurav Joshi , 2261219

Harsh Goswami , 2261242

Rahul Koranga , 2261454

Divyashish Negi , 2261199

Abstract

The Intranet-Based Online Examination System is a secure, fault-tolerant, and load-balanced platform designed to conduct examinations over a local network (LAN) without requiring internet access. Traditional examination methods are often manual, time-consuming, and susceptible to logistical inefficiencies and human error. Existing online systems, while digital, typically rely on stable internet connectivity and centralized architecture, which can become performance bottlenecks during peak usage or experience data loss in case of failures.

This project aims to solve these challenges by creating a web-based examination system that runs entirely on an intranet using open-source technologies such as PHP, MySQL, Apache, and NGINX. It supports features like timed assessments, auto-save mechanisms, question navigation, and session recovery to ensure that exam progress is preserved in case of crashes or refreshes. Additionally, the system uses NGINX as a reverse proxy to distribute load among multiple Apache web servers, ensuring consistent performance and availability for concurrent users.

From the student's perspective, the interface is simple and responsive, allowing easy login, exam selection, question answering, and submission. For administrators, it includes tools for managing questions, viewing student responses, and exporting results in CSV format for offline analysis.

Overall, this project demonstrates a practical, scalable, and cost-effective approach for educational institutions to transition from traditional paper-based exams to a secure digital model within a controlled intranet environment.

TABLE OF CONTENTS

Declaration.....	i
Certificate.....	ii
Acknowledgement.....	iii
Abstract.....	iv
Table of Contents.....	v
List of Abbreviations.....	vi

CHAPTER 1 INTRODUCTION.....	08
1.1 Prologue.....	08
2.1 Background and Motivations.....	08
3.1 Problem Statement.....	08
4.1 Objectives and Research Methodology.....	09
5.1 Project Organization.....	09
CHAPTER 2 PHASES OF SOFTWARE DEVELOPMENT CYCLE	
1.1 Hardware Requirements.....	11
2.1 Software Requirements.....	11
CHAPTER 3 CODING OF FUNCTIONS.....	12
CHAPTER 4 SNAPSHOT.....	14
CHAPTER 5 LIMITATIONS (WITH PROJECT)	16
CHAPTER 6 ENHANCEMENTS.....	17
CHAPTER 7 CONCLUSION.....	19
REFERENCES.....	20

LIST OF ABBREVIATIONS

LAN: Local Area Network – Used in this project to host the exam server without internet.

PHP: Hypertext Preprocessor – It powers the backend of the examination system.

SQL: Structured Query Language – Used in the project for managing exam questions, students, and results.

NGINX: Engine X – A high-performance web server and reverse proxy used for load balancing multiple Apache servers in the system.

UI: User Interface – The part of the system that interacts with users (students/admins). Built using HTML, CSS, and JavaScript for a smooth experience

Chapter 1

INTRODUCTION

1.1 Prologue

Over the past decades, educational institutions have increasingly adopted digital solutions to streamline administration and enhance the exam-taking process. Traditional paper-based examinations are not only labor-intensive but also prone to errors and logistical challenges. In contrast, an online examination system offers benefits such as quicker grading, automated record keeping, and better security. This project develops an Intranet-Based Online Examination System designed for local network use, ensuring that exams can be conducted efficiently even without internet connectivity. The system incorporates fault tolerance, auto-save functionality, and load-balancing across multiple servers, thereby ensuring high availability and robustness during peak usage or server outages.

2.1 Background and Motivation

The digitization of examinations aims to reduce the administrative burden and potential for human error in marking and record management. However, many existing online systems depend on internet connectivity, leaving institutions vulnerable during network outages. Furthermore, high student volume during exams can overload a single server, leading to delays or system failures. This project addresses these challenges by:

- Operating on a secure local (intranet) network, which removes dependency on the internet and enhances data security.
- Incorporating load balancing, which distributes student requests across several backend servers to prevent any single point of failure.
- Implementing fault tolerance features, such as auto-saving of exam progress and session recovery, to ensure that student data is not lost in the event of a browser crash or server interruption.

3.1 Problem Statement

The primary challenge is to build a robust online examination system that functions reliably on an intranet. The system must support fault tolerance, ensuring that if parts of the system or the user's browser crash, students do not lose their work. It must also incorporate load balancing to handle simultaneous exam takers without performance degradation. Additionally, the administration should be able to monitor and view exam results in real-time.

4.1 Objectives and Research Methodology

Objectives:

- Develop a secure web-based exam platform using PHP and MySQL.
- Design a fault-tolerant system with features such as auto-save, resume, and session recovery.
- Implement load balancing using NGINX and Apache virtual hosts.
- Create an intuitive and responsive UI for students and administrators.
- Enable real-time result monitoring and CSV export for administrative purposes.

Research Methodology:

- **Literature Review:** Study existing online examination systems, fault tolerance methods, and load balancing techniques.
- **Requirement Analysis:** Identify the functional and non-functional requirements (e.g., usability, security, reliability) for the system.
- **Design and Prototyping:** Develop initial wireframes, system architecture diagrams, and database schemas.
- **Implementation:** Code the application using PHP, JavaScript, HTML/CSS, and configure the server environment using XAMPP and NGINX.
- **Testing:** Perform unit testing, integration testing, and user acceptance testing (UAT) on an intranet environment.
- **Documentation:** Prepare comprehensive documentation and user manuals for further development and maintenance.

5.1 Project Organization

The project is organized into the following chapters:

- **Chapter 1:** Introduction – Outlining the background, objectives, and overall organization.
- **Chapter 2:** Phases of Software Development – Detailing hardware/software requirements and the development lifecycle.

- **Chapter 3:** Coding of Functions – Explaining key code segments for student login, exam management, fault tolerance features, and auto-save functions.
- **Chapter 4:** Snapshots – Providing screenshots of the application (login screen, exam interface, admin panel, etc.).
- **Chapter 5:** Limitations – Discussing inherent limitations of the current design.
- **Chapter 6:** Enhancements – Proposing possible improvements for future iterations.
- **Chapter 7:** Conclusion – Summarizing the project outcomes and future outlook.
- **References:** Listing sources and documentation used during development.

Chapter 2

HARDWARE AND SOFTWARE REQUIREMENTS

1.1 Hardware Requirements

- **Processor:** Intel i3 or higher
- **RAM:** Minimum 4 GB
- **Storage:** 1 GB free space
- **Network:** Reliable LAN connection (Ethernet or Wi-Fi)

2.1 Software Requirements

- **Operating System:** Windows 10/11
- **Server Platform:** XAMPP (includes Apache, PHP 8.x, and MySQL/MariaDB)
- **Load Balancing:** NGINX configured as a reverse proxy
- **Development Tools:** Visual Studio Code or Sublime Text
- **Browser:** Google Chrome or Mozilla Firefox

Chapter 3

CODING OF FUNCTIONS

3.1 Overview of Key Modules

- Login Module (login.php):
 - Authenticates students using a universal password.
 - Initiates a session storing student ID, username, and exam selection.
- Dashboard (dashboard.php & start_exam.php):
 - Displays available exams for selection.
 - Validates exam selection and initializes an exam session.
 - Uses Apache Virtual Hosts for multi-server deployment.
- Exam Interface (exam.php):
 - Displays exam questions one at a time with navigation controls (Next/Previous buttons).
 - Implements a countdown timer with fault tolerance to resume after refresh or crash.
 - Employs AJAX for auto-save functionality to continuously store answers in the database.
- Autosave Mechanism (autosave.php & save_progress.php):
 - Captures student input periodically (every 15 seconds) to update their progress in real time.
 - Uses session data to associate saved responses with the correct student and exam session.
- Fault Tolerance Module:
 - Client-Side: Utilizes localStorage (or server-side session management) to store the exam start time and current question index so that the exam can resume accurately after a browser refresh or crash.
 - Server-Side: Implements load balancing through NGINX, distributing requests among Apache servers (exam_server1, exam_server2, exam_server3), ensuring high availability.
- Results Module (results.php & export_csv.php):
 - Allows the admin to view detailed results, showing the total number of correct answers, student data, and answer breakdown.
 - Provides CSV export functionality for result data.

3.2 Notable Code Snippets

Fault tolerance cliend side timer:

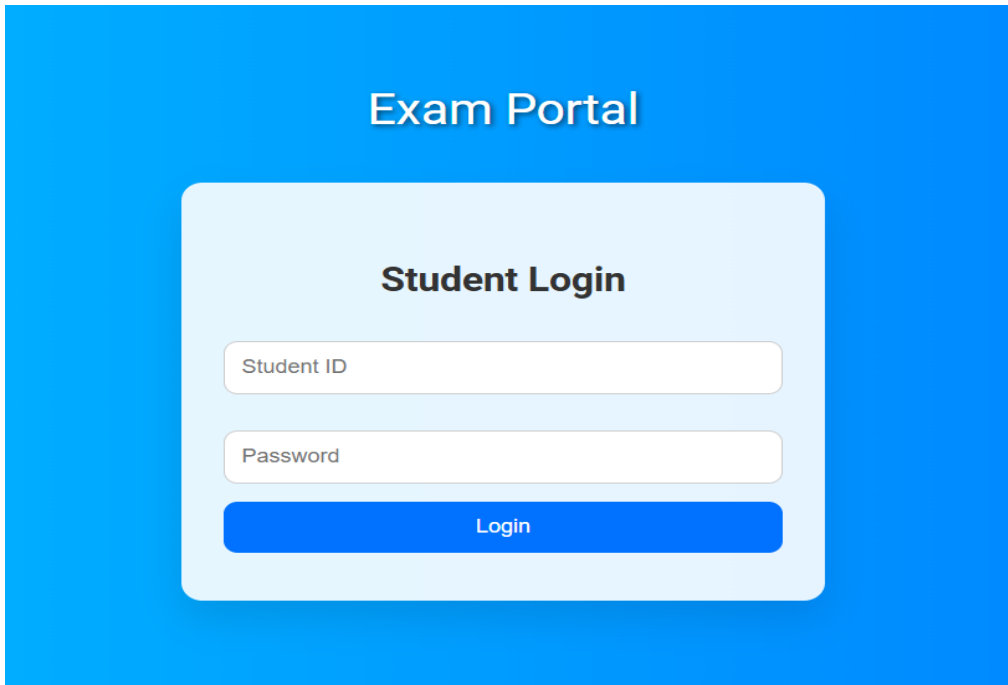
```
$start_time = $session['start_time'];  
$start_timestamp = strtotime($start_time);  
$now = time();  
$duration_minutes = 10;  
$elapsed = $now - $start_timestamp;  
$remaining_time = max(0, ($duration_minutes * 60) - $elapsed);
```

Load Balancing Configuration:

```
worker_processes 1;  
  
events {  
    worker_connections 1024;  
}  
  
http {  
    upstream exam_servers {  
        server 127.0.0.1:8081;  
        server 127.0.0.1:8082;  
        server 127.0.0.1:8083;  
    }  
  
    server {  
        listen 8000;  
        server_name localhost;  
  
        location / {  
            proxy_pass http://exam_servers;  
            proxy_set_header Host $host;  
            proxy_set_header X-Real-IP $remote_addr;  
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
            proxy_set_header X-Forwarded-Proto $scheme;  
        }  
    }  
}
```

Chapter 4

SNAPSHOTS



The image shows a 'Student Login' form within a blue 'Exam Portal' container. The form has a light blue background and contains two input fields: 'Student ID' and 'Password'. Below these fields is a blue 'Login' button.

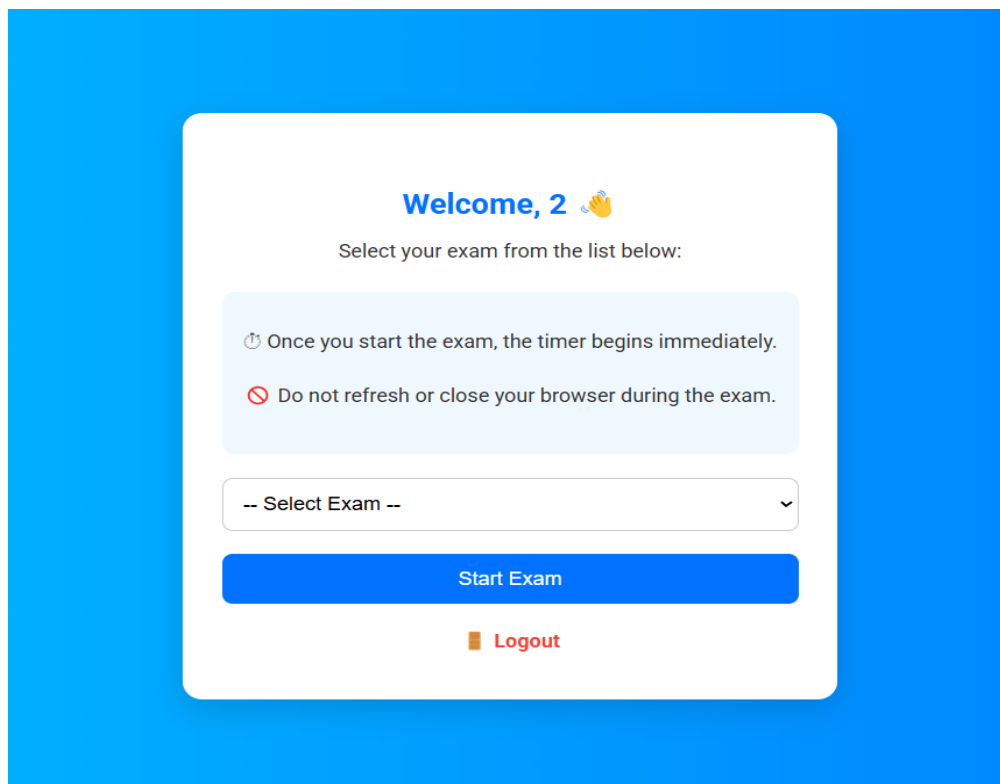
Exam Portal

Student Login

Student ID

Password

Login



The image shows a 'Welcome' screen within a blue container. It features a white card with the text 'Welcome, 2 🙌' and 'Select your exam from the list below:'. Below this is a light blue box containing two instructions: 'Once you start the exam, the timer begins immediately.' and 'Do not refresh or close your browser during the exam.' Below the instructions is a dropdown menu labeled '-- Select Exam --' and a blue 'Start Exam' button. At the bottom of the card is a 'Logout' link with a red icon.

Welcome, 2 🙌

Select your exam from the list below:

⌚ Once you start the exam, the timer begins immediately.

🚫 Do not refresh or close your browser during the exam.

-- Select Exam --

Start Exam

Logout

1

2

3

4

5

6

7

8

9

10

Exam

Time Left: 09:59

Q1: Which of the following is a regular language

- ☐ $a^n b^n$
- ☐ $a^n b^n c^n$
- ☐ a^*b^*
- ☐ $a^n b^{n+1}$

Previous

Next

Submit



Admin Login

Admin Username

Password

Login

Chapter 5

LIMITATIONS

While this system is robust and feature-complete, there are some limitations:

- **Limited Scalability:** The system currently runs on an intranet with a limited number of students. Enterprise-level scalability would require cloud infrastructure.
- **Dependence on Local Network:** The system is not accessible over the internet, limiting its use to local environments.
- **Simplified Authentication:** The system uses a universal password for demo purposes; a production system would require a more secure, individual login mechanism.
- **No Question Randomization:** All students receive the same questions in the same order. Randomization would improve exam integrity.
- **Basic UI/UX:** Although functional, the user interface can be enhanced with more advanced design patterns and responsive behavior.

Chapter 6

ENHANCEMENTS

1. Question Pool and Randomization

- Implement a dynamic question bank that supports random question generation per student.
- Support randomized options (A/B/C/D order) to minimize cheating.
- Define difficulty levels and categories for intelligent question selection.

2. Role-Based Access Control (RBAC)

- Introduce multi-level access roles:
 - **Student:** Attempt exams
 - **Faculty:** Upload/edit questions
 - **Admin:** View reports, manage exams/users
- Secure pages with authentication checks and restrict unauthorized access.

3. Real-Time Proctoring Integration

- Integrate WebRTC or webcam access to monitor candidates during the exam.
- Add screenshot logging or face detection for suspicious behavior.

4. Offline Exam Sync Mechanism

- Allow students to take exams offline (e.g., in a desktop app or Electron JS app), and sync responses when reconnected to the intranet.
- Use a background task to store responses locally if the server is temporarily unreachable.

5. Cloud Migration (Optional for Scalable Use)

- Port the application to cloud infrastructure (AWS, Azure, GCP) with:
 - Auto-scaling servers
 - Distributed database (e.g., Amazon RDS)
 - Serverless architecture (e.g., Lambda for auto-save)

6. Analytics & Insights

- Graphical dashboards for performance visualization:
 - Pie charts, bar graphs, average scores
 - Student-wise and question-wise analytics
- Track question difficulty by analyzing success rates.

7. Email/SMS Notification

- Send exam start reminders, completion alerts, and result summaries to students.
- Alert admins when a student encounters submission or login issues.

8. Multilingual Support

- Add language selection for UI and questions (Hindi, English, regional languages).
- Use UTF-8 encoding and dynamic translation tables.

9. Enhanced UI/UX

- Fully responsive design using frameworks like Bootstrap or Tailwind CSS.
- Add dark mode, keyboard navigation, and accessibility features (WCAG 2.1 compliance).

10. Advanced Security Features

- Secure session management and auto logout after inactivity.
- Encrypt answer submissions using HTTPS.
- Hash passwords (currently stored as NULL or plain) using bcrypt.

CONCLUSION

This project demonstrates how to build a resilient, intranet-based examination system using open-source technologies. By incorporating fault tolerance via auto-save and session recovery along with load balancing across multiple Apache servers via NGINX, the system is able to handle common issues like unexpected browser refreshes, crashes, or server failures without loss of data. Although designed primarily for a local network, this project lays the groundwork for scalable and secure examination systems that could be adapted for larger deployments. Future enhancements will focus on increased security, advanced authentication, mobile responsiveness, and cloud-based scalability.

REFERENCES

1. Tanenbaum, A. S., & Bos, H. (2015). *Modern operating systems* (4th ed.). Pearson. <https://www.pearson.com/en-us/subject-catalog/p/modern-operating-systems/>
2. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating system concepts* (10th ed.). Wiley. <https://www.wiley.com/en-us/Operating+System+Concepts%2C+10th+Edition-p-9781119320913>
3. Sharma, R. (2022). Load balancing techniques in distributed systems. *International Journal of Computer Science and Engineering*, https://www.ijcseonline.org/full_paper_view.php?paper_id=6724
4. Microsoft. (n.d.). Introduction to multithreading. Retrieved April 8, 2025, from <https://learn.microsoft.com/en-us/dotnet/standard/threading/>