AN AUTOMATED INVIGILATION SYSTEM

Application Development Report Submitted
In partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology
in
Information Technology

by

P.NAVEEN REDDY	**23N35A1214**
R.SRIHARI	**23N35A1215**
T.SIDDHARTH REDDY	**22N31A12H8**

Under the esteemed guidance of

Dr.A.LAKSHMAN
Associate Professor

Department of Information Technology

Malla Reddy College of Engineering & Technology

(Autonomous Institution- UGC, Govt. of India)
(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with
'A'Grade)
Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100
website: www.mrcet.ac.in

Malla Reddy College of Engineering & Technology

(Autonomous Institution- UGC, Govt. of India)
(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with 'A' Grade)
Maisammaguda, Kompally, Dhulapally,Secunderabad – 500100
website: www.mrcet.ac.in

CERTIFICATE

This is to certify that this is the bonafide record of the Application Development entitled "An automated invigilation system", submitted by P.Naveen Reddy (23N35A1214),R.SRIHARI (23N35A1215) and T.Siddharth Reddy (22N31A12H8) of B.Tech in the partial fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology, Department of IT during the year 2024-2025.

Internal Guide	**Head of the Department**
Dr.A.Lakshman **Associate Professor**	**Dr. G.Sharada** **Professor**

External Examiner

ABSTRACT

The aim of this project is to provide an effective management of invigilation schedules for the smooth operation of examinations and maintaining academic integrity. This project automates the scheduling process for exam invigilators. The proposed system utilizes a user-friendly interface to manage, organize, and optimize invigilation schedules through web pages.Users can access the system to view their assigned shifts, and receive notifications about updates or reminders.By leveraging modern web technologies, the system aims to improve efficiency, accuracy and transparency in the management of invigilation schedules.The project demonstrates how web-based tools can transform traditional scheduling practices, providing a scalable and adaptable solution for educational institutions.

TABLE OF CONTENTS

INTRODUCTION

1.1 PURPOSE AND OBJECTIVES

PURPOSE

An automated invigilation assignment system serves several key purposes are that it streamlines the process of assigning invigilators to examinations, reducing administrative workload and time spent on manual scheduling. The system can ensure an equitable distribution of invigilation duties among staff, preventing bias and favoritism in assignments.
Automation minimizes human error in scheduling, ensuring that all exams are adequately supervised .It can store and manage records of invigilator assignments, exam schedules, and performance metrics for future reference and reporting .The system can accommodate last-minute changes and communicate updates promptly to all stakeholders, ensuring smooth operations .By maintaining detailed records and following predefined protocols, the system helps ensure compliance with examination regulations and enhances security against cheating and it helps in effectively utilizing available invigilators based on their availability and expertise, improving overall resource management.

OBJECTIVES

The objectives of an automated invigilation assignment system typically include:

- **Streamlining Scheduling**: Automate the scheduling process to reduce the time and effort required for assigning invigilators to exams.
- **Ensuring Fairness**: Create a fair and unbiased distribution of invigilation duties among staff members.
- **Enhancing Accuracy**: Minimize errors in the assignment process to ensure that all exams are properly supervised.
- **Improving Communication**: Facilitate clear and timely communication between administrators, invigilators, and students regarding exam schedules and assignments.
- **Providing Real-time Flexibility**: Allow for quick adjustments to assignments in response to changes, such as staff availability or emergency situations.
- **Optimizing Resource Allocation**: Effectively utilize available invigilators based on their qualifications, availability, and past performance.

These objectives help create a more efficient, fair, and secure examination process.

1.2 EXISTING SYSTEM & PROPOSED SYSTEM

EXISTING SYSTEM

A manual system for scheduling exam invigilation presents several challenges, including time-consuming processes, increased human error, and inefficient resource allocation. Additionally, manual systems provide limited reporting and analytics, making it difficult to assess efficiency and adapt to schedule changes. This inflexibility can result in poor coverage assurance and increased stress for administrative staff, ultimately impacting the integrity of the examination process. An Automated Invigilation Scheduler can effectively address these issues, enhancing overall efficiency and reliability.

PROPOSED SYSTEM

The proposed system for an automated invigilation scheduler system is a comprehensive solution designed to optimize the process of assigning invigilators to examination halls and schedules. The system consists of four main components: the Invigilator Module, Venue Module, student module, and Scheduling Engine. The Invigilator Module manages invigilator profiles, availability, and workload, while the Venue Module manages venue properties, and capacity. The Student Module manages student profiles, requirements, and exam schedules. The Scheduling Engine is the core component that automatically generates schedules based on constraints and preferences.

APPLICATION DESCRIPTION

2.1 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

• **Computer**: A desktop or laptop computer capable of running the development environment and compiling C programs.

• **Processor**: A processor (CPU) is the logic circuitry that responds to and processes the basic instructions that drive a computer. for example, intel.

• **Memory (RAM)**: Adequate RAM to support the development environment and execution of the program without performance issues.

• **Input Devices**: Keyboard and mouse (or alternative input devices) for interacting with the development environment and running the program.

SOFTWARE REQUIREMENTS

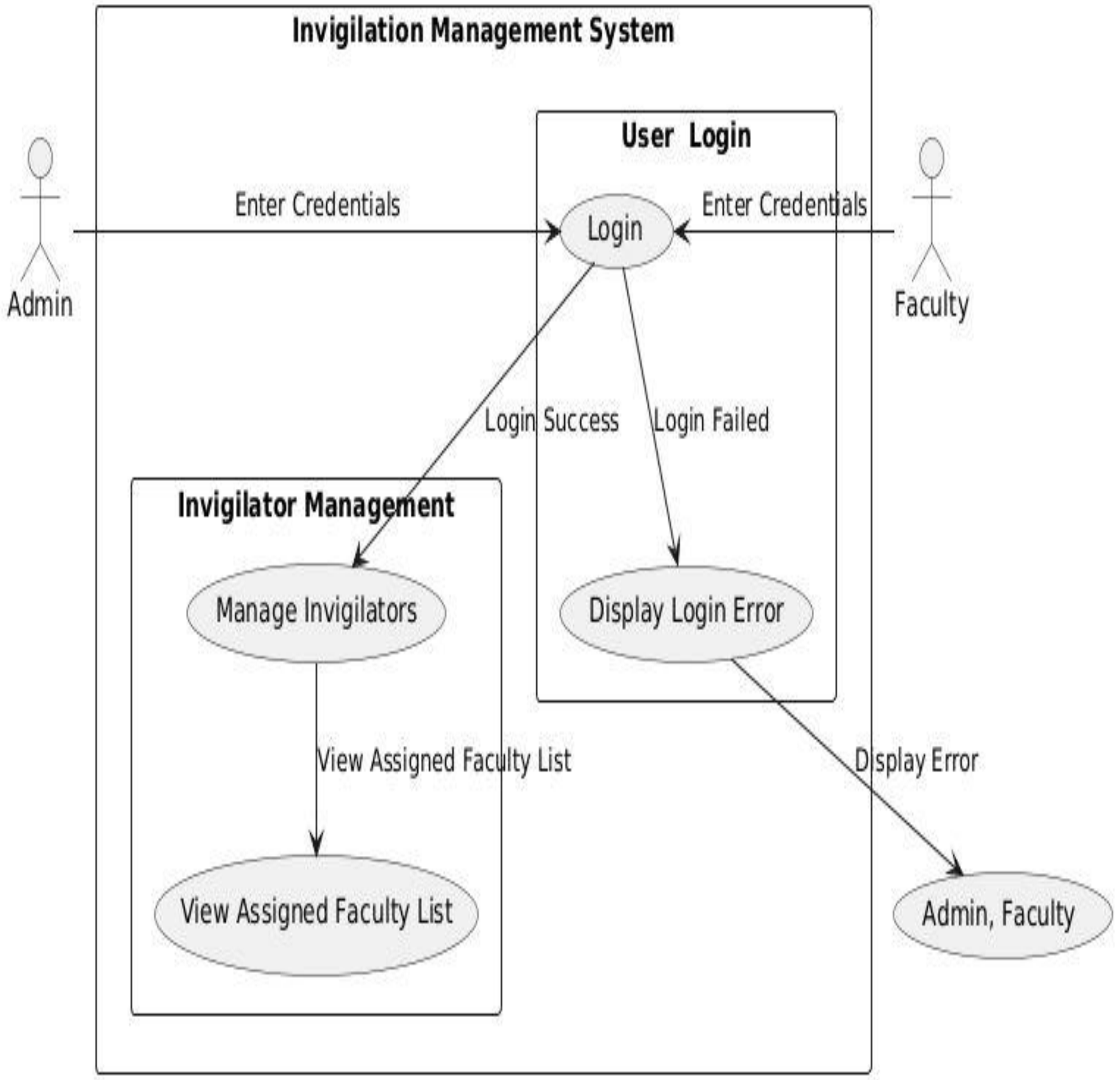
• **Operating System**: A compatible operating system that supports C programming and development tools. This could be Windows or Linux.

• **Development Environment**: An Integrated Development Environment (IDE) or text editor suitable for writing, editing, and compiling C code. Popular options include:
IDEs: Visual Studio.
Text Editors: Visual Studio Code.

• **Front end**: The front end is a combination of two different elements: the graphic design (the look) and the user interface (the feel). most of the technical work going into the user interface using web languages like HTML, CSS, and JavaScript.

• **Back end:** It is the part of the website users see and interact with. The server-side part that processes data and ensures everything works behind the scenes and not visible to users.

2.2 METHODOLOGY



MODULE DESCRIPTION

There are mainly three modules present in an automated invigilation system which includes :

- Main module
- User login module
- Invigilator management module

Main module : In main module we have different type of resources such as Exam halls ,Faculty list ,Contact information and we also have two important things which are admin login and faculty login.

User login module : In user login module a user can either login into administrative login or faculty login using username and password if the username and password matches then it will redirect to the next page where we can schedule the invigilation or view the invigilation.

Invigilator management module : In this particular module we can manage the invigilators and according to their availability we can assaign them to the invigilations . After assigning of the invigilations admin and faculty can view them.

SOURCE CODE

Main.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>An Automated Invigilation</title>
   <link rel="stylesheet" href="Main.css">
</head>
<body>
   <div class="navbar">
     <div class="icon">
       <!-- download-removebg-preview.png -->
       <img src="download-removebg-preview (1).png" class="logo">
       <h2 class="logo-text">MRCET</h2>
     </div>
     <div class="menu">
       <ul>
         <li><a href="Main.html"><abbr title="Main Page">Home</abbr></a></li>
         <li><a href="about.html"><abbr title="details">About</abbr></a></li>
         <li><a href="FacultyList.html"><abbr title="Faculty list">Faculty</abbr></a></li>
         <li><a href="halls.html"><abbr title="Examination Halls">Halls</abbr></a></li>
         <li><a href="contact (1).html"><abbr title="Contact">Contact</abbr></a></li>
       </ul>
     </div>
   </div>

  <div class="main">
   <div class="content">
     <h1>An Automated <br><span>Invigilation</span> <br>System</h1>
     <p class="par">An automated invigilation system helps in organizing and managing<br>
the scheduling of exam invigilators.</p>

     <div class="buttons">
         <button class="cn"><a href="Admin.html" target="_blank">Admin
Login</a></button>
         <button class="cn"><a href="faculty.html" target="_blank">Faculty
Login</a></button>
     </div>
   </div>
  </div>
  <div class="footer1">
<p>© 2024 MRCET.All rights resereved<br>
     TERMS OF USE | PRIVACY POLICY | SITEMAP</p>
   </div>
</body>
</html>
```

Main.css :

```
* {
  margin: 0;
  padding: 0;
}

.main {
  width: 100%;
  background: linear-gradient(to top, rgba(0, 0, 0, 0.5)50%, rgba(0, 0, 0, 0.5)50%),
url(banner1.jpg);
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  background-position: center;
  background-size: cover;
  height: 100vh;
}

.navbar {
  width: 100%;
  height: 75px;
  margin: auto;
  background-color: #002f88;
  display: flex;
  flex-direction: row;
  justify-content: space-between;
}

.icon {
  display: flex;
  flex-direction: row;
  align-items: center;
}


.logo {
  height: 50px;
  width: 70px;
  margin-left: 10px;
}


.logo-text {
  color: white;
  font-size: 35px;
  float: left;
  padding-top: 10px;
  padding-left: 5px;
}

.menu {
  width: 670px;
  float: left;
  height: 70px;
}

ul {
  float: left;
  display: flex;
  justify-content: center;
  align-items: center;
}

ul li {
  list-style: none;
  margin-left: 62px;
  margin-top: 27px;
  font-size: 15px;
}

ul li a {
  text-decoration: none;
  color: #fff;
  font-family: Arial;
  font-weight: bold;
  transition: 0.4s ease-in-out;
}

abbr {
  text-decoration: none;
}

.footer1 {
  font-family: Arial;
  height: 50px;
  padding-top: 20px;
  text-align: center;
  color: white;
  background-color: #002f88;
}
```

Admin.html :

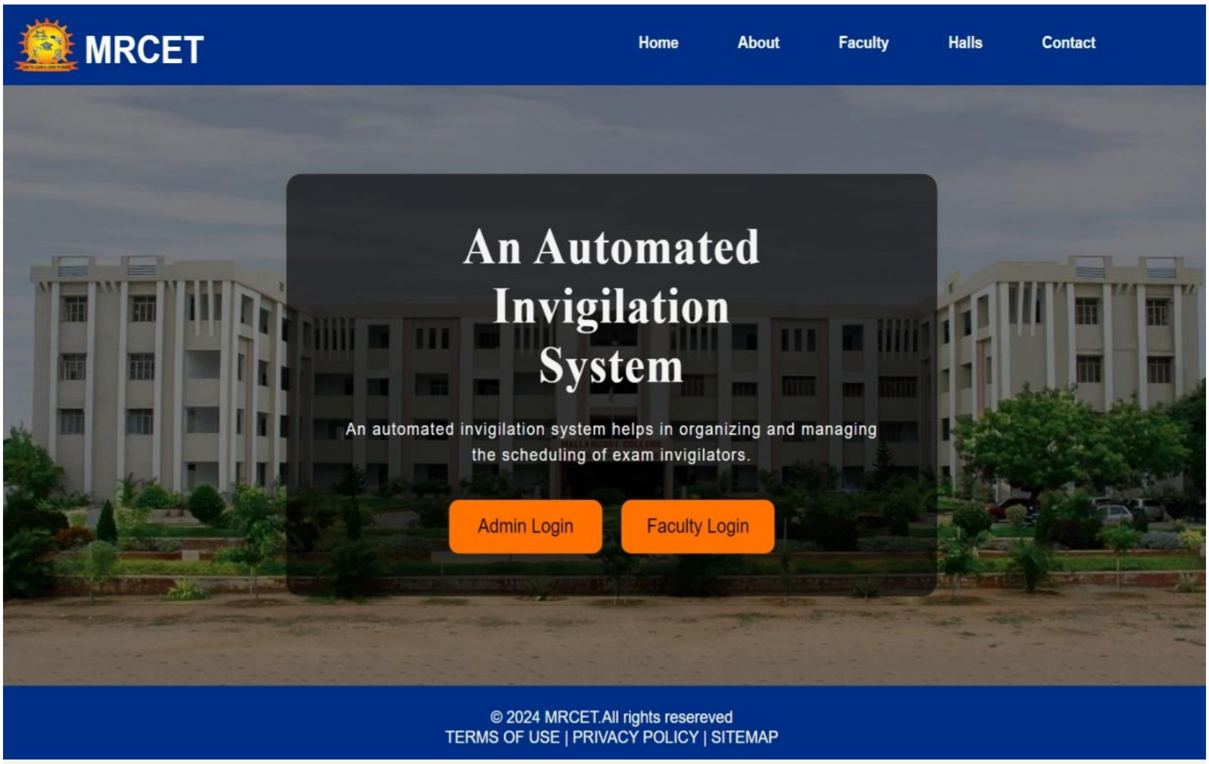
```
<DOCTYPE HTML!>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>An Automated Invigilation</title>
 <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=
swap"rel="stylesheet" />
    <link rel="stylesheet" href="navbar.css" />
    <link rel="stylesheet" href="Change.css" />
  </head>
  <body>
    <div class="navbar">
      <div class="icon">
        <img src="download-removebg-preview (1).png" class="logo" />
        <h2 class="logo-text">MRCET</h2>
      </div>
      <div class="menu">
        <ul>
          <li>
            <a href="Main.html"><abbr title="Main Page">Home</abbr></a>
          </li>
          <li>
            <a href="about.html"><abbr title="details">About</abbr></a>
          </li>
          <li>
            <a href="FacultyList.html"><abbr title="Faculty list">Faculty</abbr></a>
          </li>
          <li>
          <a href="halls.html"><abbr title="Examination Halls">Halls</abbr></a >
          </li>
          <li>
            <a href="contact (1).html"><abbr title="Contact">Contact</abbr></a>
          </li>
        </ul>
      </div>
    </div>
    <div class="container">
      <h1 style="color: #007bff">An Automated Invigilation</h1>
      <label for="examDate">Enter Examination Date:</label>
      <input type="date" id="examDate" required />
      <fieldset>
        <legend>Select Exam Time:</legend>
        <label>
          <input
            type="checkbox"
            value="Forenoon"
            onchange="updateSessionSelection()"/>
          Forenoon
        </label>
        <label>
          <input
            type="checkbox"
            name="session"
            value="Afternoon"
            onchange="updateSessionSelection()"/>
          Afternoon
        </label>
      </fieldset>
      <label for="numForenoon">Number of Faculty Needed for Forenoon:</label>
      <input type="number" id="numForenoon" min="1" required />
      <label for="numAfternoon">Number of Faculty Needed for Afternoon:</label>
      <input type="number" id="numAfternoon" min="1" required />
      <label for="facultyNames">Select Faculty Names:</label>
      <div class="dropdown">
        <input
          type="text"
          id="facultyInput"
          placeholder="Select Faculty"
          readonly
          onclick="toggleDropdown()"/>
        </div>
      </div>

      <label for="blockName">*Enter Block Name:</label>
      <input type="text" id="blockName" placeholder="e.g., IT,CSE,ECE" />

      <button onclick="assignInvigilation()">Assign Invigilation</button>

      <div class="selected-faculty" id="selectedFaculty"></div>
    </div>

    <script src="Change.js"></script>
  </body>
</html>
```

Change.js :

```
function toggleDropdown() {
  document.getElementById("facultyDropdown").classList.toggle("show");
}
function updateSelectedFaculties() {
  const checkboxes = document.querySelectorAll(
    '#facultyDropdown input[type="checkbox"]'
  );
  const selectedFaculties = Array.from(checkboxes)
    .filter((checkbox) => checkbox.checked && checkbox.value !== "Select All")
    .map((checkbox) => checkbox.value);

  document.getElementById("facultyInput").value = selectedFaculties.join(", ");
  document.getElementById(
    "selectedFaculty"
  ).innerText = Selected Faculty: ${selectedFaculties.join(", ")};

  // Update "Select All" checkbox state
  const allChecked = selectedFaculties.length === checkboxes.length - 1; // -1 for the "Select
All" checkbox
  document.getElementById("selectAll").checked = allChecked;
}

function toggleSelectAll() {
  const selectAllCheckbox = document.getElementById("selectAll");
  const checkboxes = document.querySelectorAll(
    '#facultyDropdown input[type="checkbox"]'
  );
  checkboxes.forEach((checkbox) => {
    if (checkbox.value !== "Select All") {
      checkbox.checked = selectAllCheckbox.checked;
    }
  });
  updateSelectedFaculties();
}

function updateSessionSelection() {
  const sessionCheckboxes = document.querySelectorAll('input[name="session"]');
  const selectedSessions = Array.from(sessionCheckboxes)
    .filter((checkbox) => checkbox.checked)
    .map((checkbox) => checkbox.value);

  // Store the selected sessions for later use
  localStorage.setItem("selectedSessions", JSON.stringify(selectedSessions));
}

function assignInvigilation() {
  const examDate = document.getElementById("examDate").value;
  const numForenoon = parseInt(document.getElementById("numForenoon").value);
  const numAfternoon = parseInt(document.getElementById("numAfternoon").value);
  const selectedFaculties = document
    .getElementById("facultyInput")
    .value.split(", ")
    .filter((name) => name !== "");
  const blockName = document.getElementById("blockName").value || "No Block";

  const selectedSessions =
    JSON.parse(localStorage.getItem("selectedSessions")) || [];

  if (
    !examDate ||
    selectedFaculties.length === 0 ||
    selectedSessions.length === 0
  ) {
    alert(
      "Please fill in all required fields: Exam Date, Selected Faculty, and Session."
    );
    return;
  }

  const assignments = [];

  selectedSessions.forEach((session) => {
    const numNeeded = session === "Forenoon" ? numForenoon : numAfternoon;
    for (let i = 0; i < numNeeded; i++) {
      if (selectedFaculties.length === 0) {
        alert("Not enough faculty available for the required sessions.");
        break;
      }
      const assignedFaculty =
        selectedFaculties.splice(
          Math.floor(Math.random() * selectedFaculties.length),
          1
        )[0] || "No Faculty";
      assignments.push({
        date: examDate,
        session: session,
        block: blockName,
        faculty: assignedFaculty,
      });
    }
  });

  localStorage.setItem("assignmentResults", JSON.stringify(assignments));
  window.location.href = "results.html";
}
```

```
window.onclick = function (event) {
  if (!event.target.matches("#facultyInput")) {
    const dropdowns = document.getElementsByClassName("dropdown-content");
    for (let i = 0; i < dropdowns.length; i++) {
      const openDropdown = dropdowns[i];
      if (openDropdown.classList.contains("show")) {
        openDropdown.classList.remove("show");
      }
    }
  }
};
```

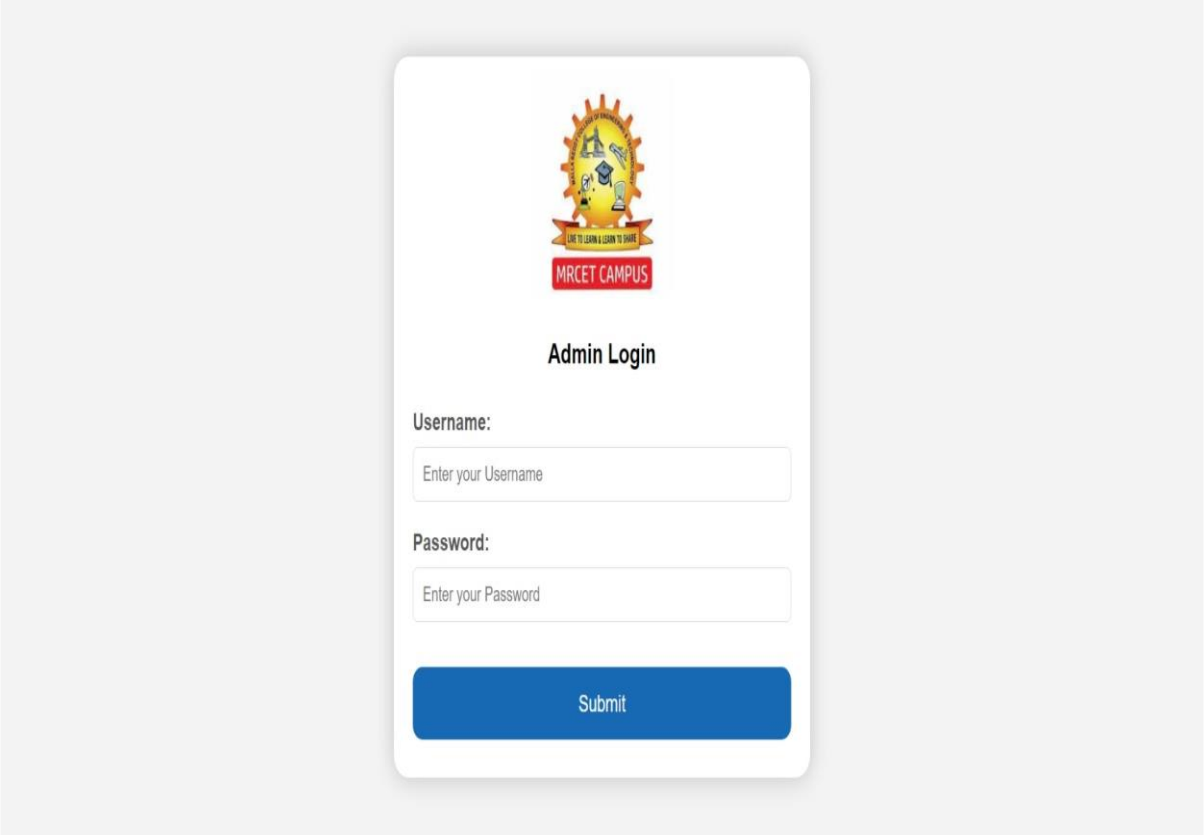
RESULTS



An Automated Invigilation System

An automated invigilation system helps in organizing and managing the scheduling of exam invigilators.

© 2024 MRCET.All rights resereved
TERMS OF USE | PRIVACY POLICY | SITEMAP



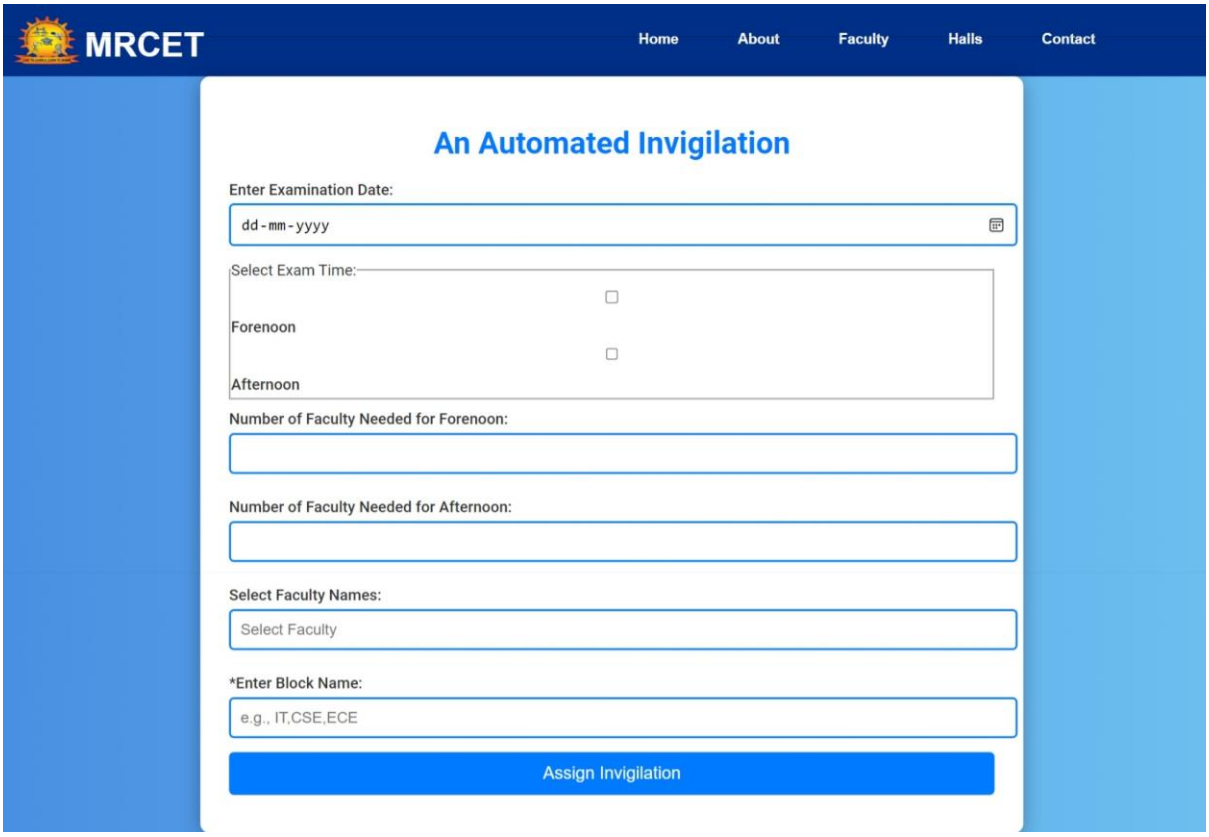
Admin Login

Username:

Enter your Username

Password:

Enter your Password



An Automated Invigilation

Enter Examination Date:

dd-mm-yyyy

Select Exam Time:

□ Forenoon

□ Afternoon

Number of Faculty Needed for Forenoon:

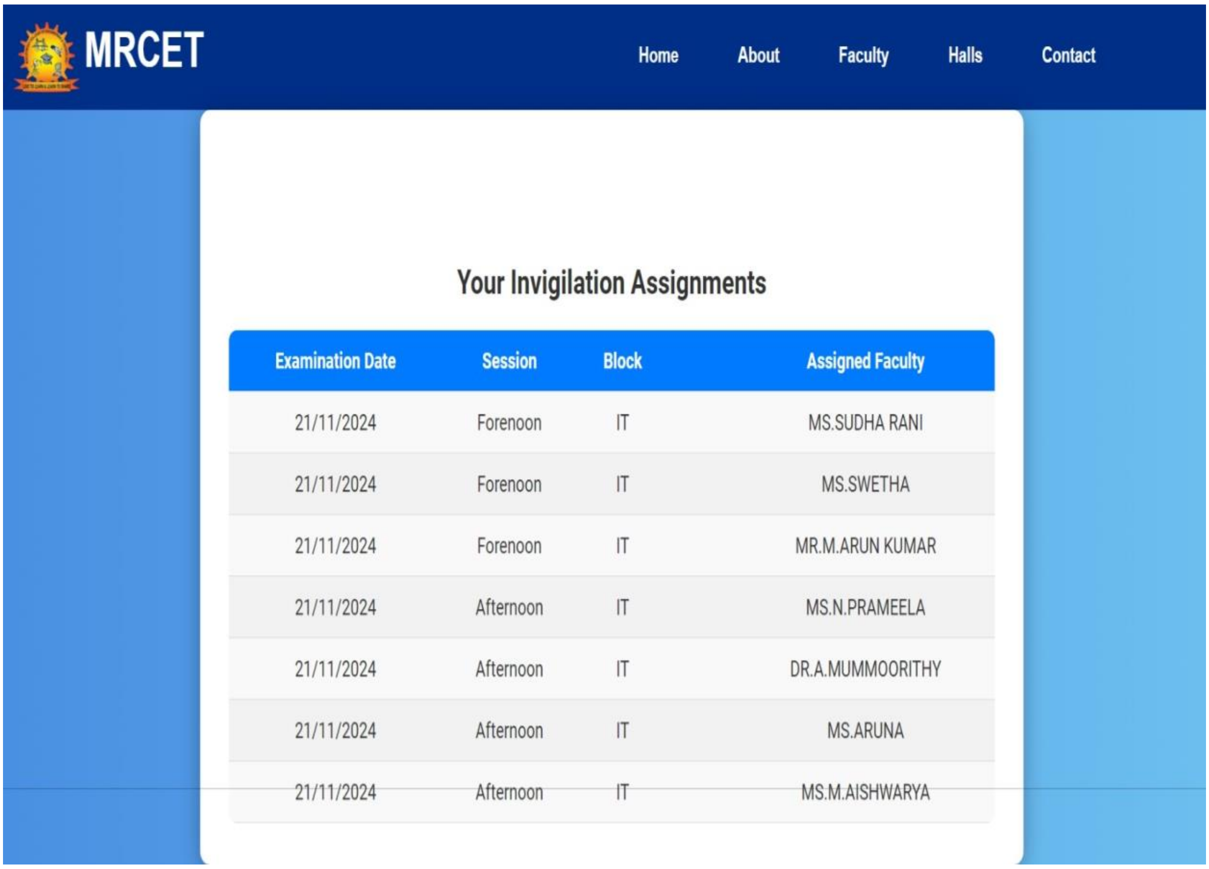
Number of Faculty Needed for Afternoon:

Select Faculty Names:

Select Faculty

*Enter Block Name:

e.g., IT,CSE,ECE



Your Invigilation Assignments

Examination Date	Session	Block	Assigned Faculty
21/11/2024	Forenoon	IT	MS.SUDHA RANI
21/11/2024	Forenoon	IT	MS.SWETHA
21/11/2024	Forenoon	IT	MR.M.ARUN KUMAR
21/11/2024	Afternoon	IT	MS.N.PRAMEELA
21/11/2024	Afternoon	IT	DR.A.MUMMOORITHY
21/11/2024	Afternoon	IT	MS.ARUNA
21/11/2024	Afternoon	IT	MS.M.AISHWARYA

CONCLUSION

In conclusion, transitioning from a manual system to an Automated Invigilation Scheduler can significantly enhance the efficiency and effectiveness of exam management. By addressing the inherent challenges of manual scheduling such as time consumption, human error, and resource misallocation this automated solution streamlines processes, ensures adequate coverage, and improves communication among staff. Ultimately, adopting such technology not only alleviates administrative burdens but also upholds the integrity of the examination process, allowing institutions to focus on their core mission of providing quality education.

ENHANCEMENTS

Enhancing an automated invigilation scheduler can improve its efficiency, user experience, and adaptability such as Developing a clean, intuitive interface for both administrators and invigilators, with easy navigation and clear instructions. Implement machine learning algorithms that optimize scheduling based on historical data, such as peak demand times and invigilator availability. Include a notification system that alerts invigilators of upcoming assignments, changes, or urgent requests via email, SMS, or app notifications. Allow users to set specific rules for scheduling (e.g., minimum rest periods between shifts, maximum number of exams in a day) to ensure fairness and compliance. Provide comprehensive reporting tools to analyze scheduling patterns, invigilator workload, and exam outcomes to improve future scheduling. Develop a mobile app or mobile-friendly website that allows invigilators to view their schedules and make updates on the go.Enable integration with existing academic systems (like learning management systems or student databases) for seamless data flow and accuracy.

BIBLOGRAPHY

https://www.crio.do/

https://www.guvi.in/

https://www.nuclino.com/solutions/project-documentation

https://github.com