Next-Gen On-Duty Automator

A MINI PROJECT REPORT

Submitted by

ARAVINTH S (221801003)
LIO GODWIN BR (221801029)

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE
AND DATA SCIENCE
ANNA UNIVERSITY, CHENNAI – 602 105

NOV-2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled **"Next-Gen On-Duty Automator**" is the bonafide work of **ARAVINTH S (221801003) and LIO GODWIN BR (221801029)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. J.M. Gnanasekar M.E., Ph.D.,
Professor and Head
Department of AI&DS
Rajalakshmi Engineering College
Chennai – 602 105

Dr. P. Indira Priya M.E., Ph.D.,
Professor
Department of AI&DS
Rajalakshmi Engineering College
Chennai – 602 105

Submitted for the project viva-voce examination held on _____.

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. J.M. GNANASEKAR., M.E., Ph.D.,** Head of the Department, Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our supervisor and coordinator, **Dr. P. INDIRA PRIYA M.E., Ph.D., Professor**, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for her valuable guidance throughout the course of the project.

Finally, we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

It aims to revolutionize the management of student on-duty (OD) requests by developing an automated, digital system. Educational institutions traditionally use a paper-based system for OD requests, where students fill out and submit forms manually and staff respond through direct communication. This method is challenging for tracking and managing requests, time-consuming, and lacks transparency. Using various web framework, the new system will automate the OD request process, making it faster and more accurate. Students can submit requests online, and staff can approve them with a few clicks. Digital records will eliminate the need for physical storage, reduce paper usage, and provide real-time updates to keep everyone informed. This system will enhance efficiency, accuracy, and transparency in handling OD requests while also minimizing paperwork.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Managing ON DUTY leave requests in academic institutions often involves cumbersome manual processes. Students must fill out paper forms and submit them to faculty, leading to delays in approvals and lack of transparency. Faculty members are required to review each request manually, which can be time-consuming and prone to errors. Furthermore, there is no centralized system for tracking leave requests and approvals.

1.2 NEED FOR THE STUDY

With the increasing number of extracurricular and academic events, students require a more efficient way to submit ON DUTY requests. Similarly, faculty members need tools that allow them to approve or reject requests while having full visibility of the student's attendance record. The proposed system automates this process, making it easier to manage and track.

1.3 OVERVIEW OF THE PROJECT

The project aims to develop an automated, digital system to manage student on-duty (OD) requests, replacing the traditional, paper-based process commonly used in educational institutions. Currently, students fill out forms manually, and staff respond through direct communication, making the system time-consuming, difficult to track, and lacking transparency.

This digital solution will use a web framework to streamline the OD request process, allowing students to submit requests online and enabling staff to review and approve them with a few clicks. Real-time updates and notifications will keep students and

staff informed, enhancing communication and transparency. The system will also eliminate the need for physical storage, reduce paper usage, and ensure accurate digital records, improving the efficiency, accuracy, and eco-friendliness of the process.

This project ultimately aims to create a more effective, sustainable, and user-friendly OD request management system for educational institutions.

1.4 OBJECTIVES OF THE STUDY

The objectives of this project are:

1. To develop a web-based platform for managing ON DUTY requests and approvals.
2. To ensure transparency by providing real-time notifications to students and faculty.
3. To implement a multi-level approval workflow involving the class incharge and HOD.
4. To integrate an analysis tool for faculty to review leave patterns over time..

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

Azari et al. [5] discuss the use of ARIMA and LSTM models for cellular traffic prediction and classification, emphasizing the strengths of each model in processing high-dimensional data. Their research highlights that while ARIMA excels in short-term predictions due to its statistical approach, LSTM models with attention mechanisms outperform in complex scenarios requiring adaptability over time. Zhou et al. [6] further support these findings by comparing time-series forecasting methods and suggesting that LSTM's flexibility makes it more suitable for dynamic environments. Together, these studies illustrate how current systems benefit from the integration of traditional statistical models and advanced machine learning techniques, enhancing predictive capabilities in industries like cellular and renewable energy.

Akhawe et al. [1] emphasize the importance of foundational frameworks to secure web applications against evolving cyber threats. By establishing such frameworks, web systems can be protected from common vulnerabilities and potential attacks. Fonseca and Vieira [2,4] expand on this by analyzing security risks tied to software faults, identifying how weak points in code design can lead to serious vulnerabilities. Moreover, researchers from the University of Belgrade [3] explore the role of user behavior in security, highlighting that awareness and preventive behaviors are critical to safeguarding data.

CHAPTER 3

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

The current process for managing ON DUTY (OD) requests in academic institutions is predominantly manual. Students typically submit paper forms to their class in-charge, who manually reviews the requests. This process is prone to several inefficiencies. Tracking and managing multiple requests simultaneously is challenging for faculty, as it requires physical records, which can easily be misplaced or mishandled. Moreover, the manual nature of the system makes it time-consuming and susceptible to human error, such as missing details or incorrect approvals. There is also a lack of transparency for students, as they often do not receive timely updates on the status of their OD requests. As a result, the overall system is inefficient, lacks automation, and fails to provide real-time feedback to users.

3.2 PROPOSED SYSTEM

The proposed system is a web-based ON DUTY management platform designed to address the shortcomings of the existing manual process. The system provides separate login portals for students and faculty, ensuring secure access to relevant functionalities.

- **Student Portal**: Students can log in using their batch and department credentials to submit ON DUTY requests through an intuitive form interface. The system automatically validates the student's details against the database. If valid, the request proceeds to the form submission stage, where the student fills out event details and submits the form for approval. Automated notifications are sent to the student, updating them on the status of their request.

● **Faculty Portal:** Faculty members, including the class in-charge and Head of Department (HOD), log in to review pending ON DUTY requests. The system provides a dashboard that allows faculty to view the student's academic performance and previous OD records before making an informed decision to approve or reject the request. Notifications are sent to both the student and the next level of authority upon each approval.

● **Multi-Level Approval Workflow:** The system includes a multi-level approval workflow where the class in-charge first reviews the request, followed by final approval from the HOD. Each step is tracked, and real-time notifications are sent to keep students and faculty updated.

● **Certificate Submission:** After the event, the student uploads a participation certificate and geotagged photos as proof of attendance. This ensures that OD leaves are granted only for valid events. Once verified by a mentor or faculty member, the OD is officially granted.

● **Data Analytics and Reporting:** The system includes an analysis feature for faculty, enabling them to generate reports on OD requests for a specified time period. This tool helps in reviewing trends, monitoring student attendance, and making data-driven decisions.

3.3 FEASIBILITY STUDY

A feasibility study was conducted to evaluate the proposed system's viability in terms of technical, operational, and economic factors:

● **Technical Feasibility:** The system is designed using widely available technologies, such as HTML, CSS, JavaScript for the frontend, and Node JS,

Express with Mongodb for the backend. These technologies are scalable and can handle the expected load of multiple users.

- **Operational Feasibility:** The system simplifies the process for both students and faculty by automating ON-DUTY and providing real-time tracking. Faculty members benefit from the data analytics feature, which reduces their administrative burden, while students gain transparency and quick feedback. The system is user-friendly, and minimal training is required for faculty and students to use it effectively.

- **Economic Feasibility:** The proposed system is cost-effective to implement, requiring only moderate hardware and software resources. By reducing paperwork and streamlining approvals, the system minimizes operational costs and administrative overhead. The longterm savings in time and resources make the investment worthwhile for academic institutions.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 SOFTWARE REQUIREMENTS

1. Frontend:

1. HTML5, CSS3, and JavaScript
2. Bootstrap.

1. Backend:

1. Node.js.
2. Express.js.

1. Database:

- MongoDB

1. Notification System:

- Email

1. Development Tools:

- Node Package Manager (NPM)
- Visual Studio Code

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

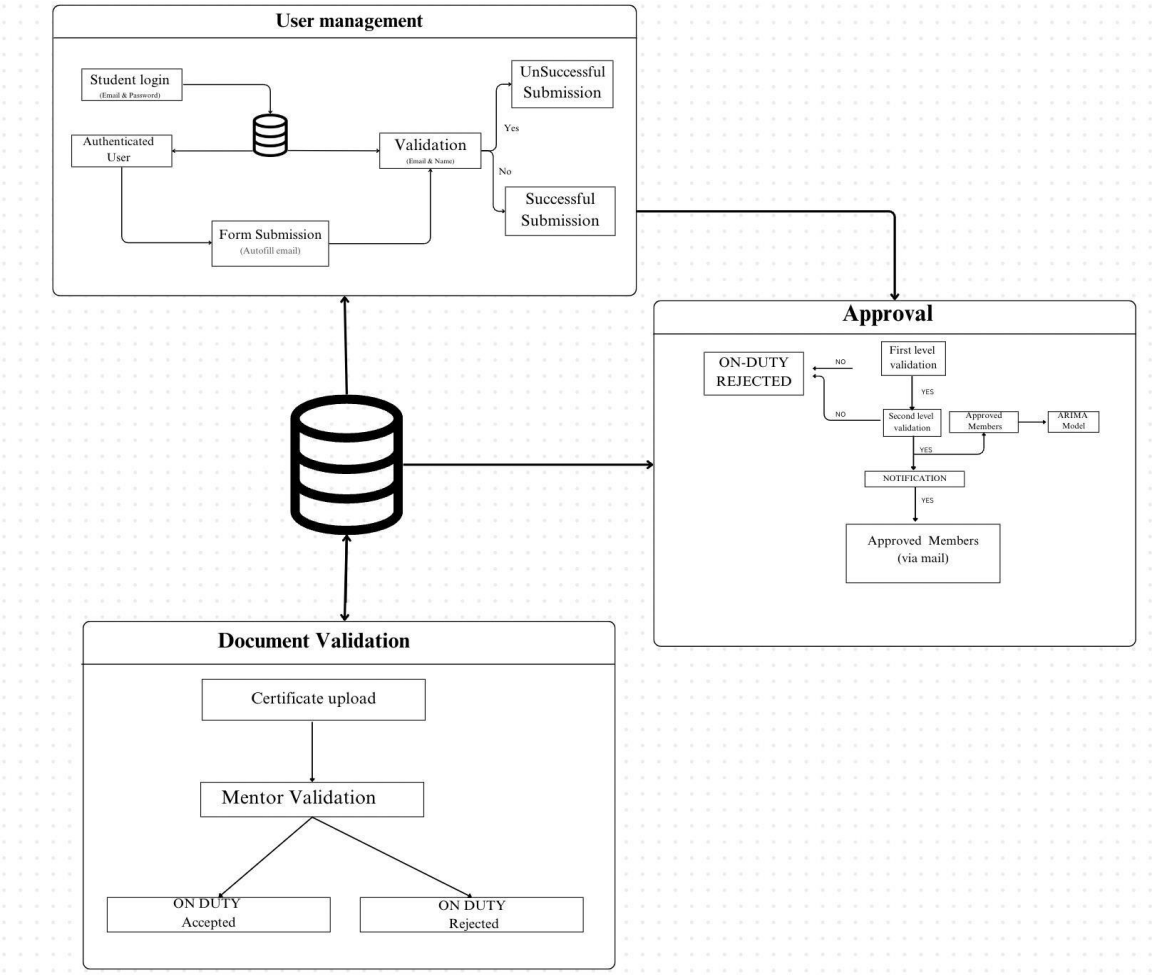


Figure 5.1: System Architecture

The system consists of three main modules:

1. **User Management Module:** Handles user registration, login, and validation.
2. **Document Validation Module:** Processes document uploads, certificate validation, and mentor validation.

3. **Approval Module:** Reviews submissions, makes approval decisions, and updates the system status accordingly.

5.2 MODULE DESCRIPTION

This section provides an overview of each module involved in the crop prediction system, explaining how each module contributes to the system's functionality and how they interact with each other. Each module is structured to align with the system's design and enhance the predictive accuracy of crop recommendations.

5.2.1 User Management Module

Step1: Prompt user to enter login credentials**.**

Step2: Collect user input (e.g., email, password).

Step3: Verify credentials against stored information in the database**.**

Step4**:** If credentials are correct, grant access to the system.

Step5**:** If credentials are incorrect, display an error message and prompt the user to retry.

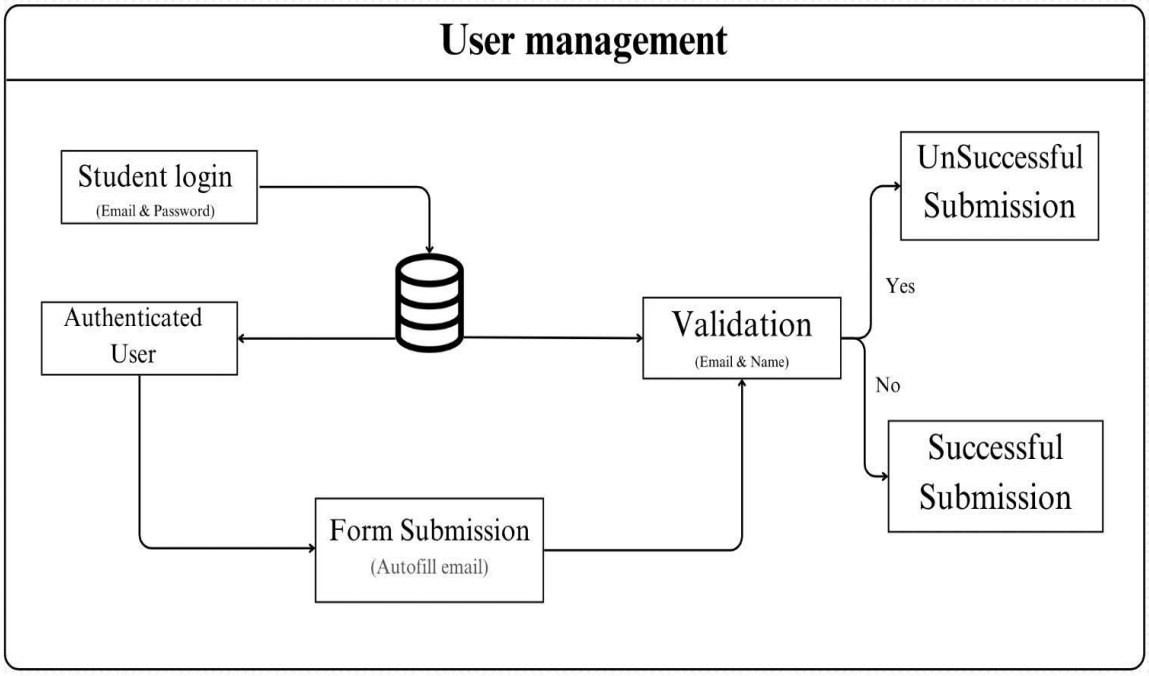


Figure 5.2.1 : User Management Module

5.2.2 Document Validation

Step 1: Prompt the student to upload required certificates (e.g., academic transcripts, proof of identity).

Step 2: Collect the uploaded documents from the student.

Step 3: Validate the uploaded certificates, including verifying file format and checking for completeness.

Step 4: Integrate with external verification services (if available) to check certificate authenticity and validity.

Step 5: If the certificate is validated, proceed to the mentor validation step otherwise, notify the student of any issues.

Step 6: Route the validated documents to the assigned mentor for further review and approval (if required).

Step 7: Allow mentors to review the documents, provide feedback, or reject submissions if they do not meet criteria.

Step 8: Update the student on the approval or rejection status.

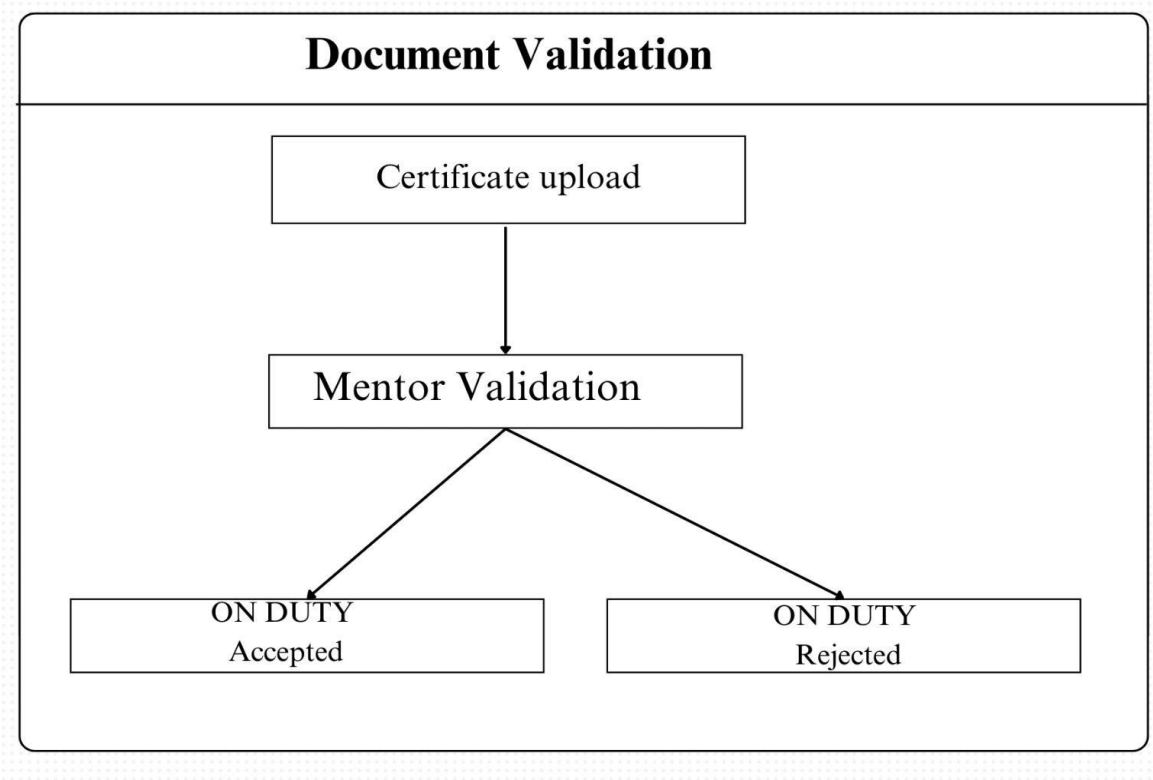


Figure 5.2.2 Document Module

5.2.3 Approval

Step 1: Collect and organize student submissions for review by the designated reviewer (e.g., mentor or administrative staff).

Step 2: Display each submission with relevant details for the reviewer to examine all required documents and information.

Step 3: Evaluate each submission against the established criteria to determine if it meets the standards for approval.

Step 4: If the submission meets all criteria, approve it and update the student's status to reflect approval.

Step 5: If issues or discrepancies are found, reject the submission, recording the reason for rejection and providing feedback for the student.

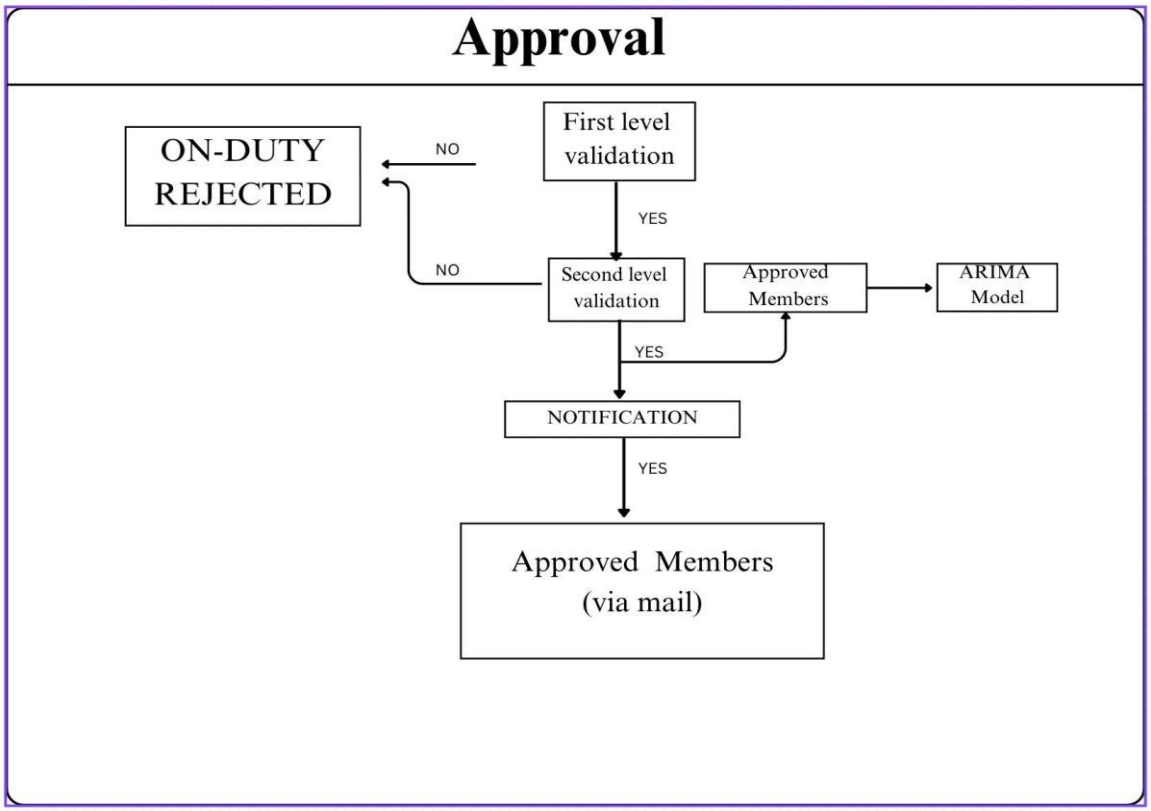


Figure 5.2.3 Approval Module

CHAPTER 6

RESULTS AND DISCUSSIONS

The AutoRegressive Integrated Moving Average (ARIMA) model was chosen for this project due to its exceptional suitability for time series data, particularly in contexts where patterns and trends must be extracted from historical data. ARIMA's core strength lies in its ability to model and forecast data with temporal dependencies, making it the perfect fit for analyzing and predicting ON DUTY (OD) request patterns. OD data typically exhibits trends over time,such as increased requests during specific periods like exams or sports events. ARIMA's components—Autoregressive (AR), Integrated (I), and Moving Average (MA)—work in harmony to identify and model these patterns, enabling accurate predictions based on past behavior. This capability is critical for administrators managing OD requests, as it allows them to anticipate peak periods, streamline resource allocation, and enhance overall efficiency.

The choice of ARIMA over other algorithms stems from several key considerations. Machine learning models, such as Long Short-Term Memory (LSTM) networks, are often used for time series analysis but require extensive computational resources and large datasets. These models are also more complex to implement and interpret, presenting challenges for projects that need immediate and actionable insights. On the other hand, traditional regression models and exponential smoothing methods, while simpler, lack the ability to capture intricate temporal dependencies and seasonal variations present in OD request data. ARIMA strikes the perfect balance, offering a robust framework for analyzing time-dependent data with moderate resource requirements and high interpretability.

A major advantage of ARIMA is its adaptability to non-stationary data, which is common in OD requests due to variations in patterns over time. The Integrated (I)

component of the model ensures that the data is transformed into a stationary format, enabling reliable analysis and forecasting. This feature is particularly important in the context of the ON DUTY Management System, where the data may include both long-term trends and short-term fluctuations. By stabilizing the data, ARIMA ensures that the predictions generated are both accurate and meaningful.

Another critical factor in selecting ARIMA is its computational efficiency. Unlike more resource-intensive algorithms, ARIMA performs well even with moderatesized datasets, making it an accessible option for projects with limited computational power. Additionally, the model's intuitive parameters (p, d, q) make it easier for stakeholders to understand and trust the results. This level of transparency is essential in an administrative setting, where decisions based on predictions must be well-supported and explainable.

The implementation of ARIMA in this project has allowed for several significant activities. It has enabled the detection of trends in OD requests, revealing patterns that align with academic and extracurricular calendars. For example, ARIMA identified increased requests during exam seasons and sports events, providing actionable insights for administrators. Furthermore, the model's forecasting capabilities have empowered the system to predict future peaks in OD requests, allowing for proactive resource planning and ensuring timely responses to student needs. By accurately modeling the data's temporal dependencies, ARIMA has proven its value as a reliable tool for optimizing the ON DUTY management process.

When compared to other time series analysis techniques, ARIMA stands out due to its versatility and effectiveness. While advanced machine learning models may offer comparable forecasting abilities, they often act as "black boxes," making it difficult to interpret their decision-making processes. In contrast, ARIMA's 14 structure is inherently interpretable, offering a clear view of how past data points influence

future predictions. Furthermore, ARIMA's flexibility allows for customization, enabling the model to adapt to various types of time series data and improving its accuracy over time.

In conclusion, the application of ARIMA in the ON DUTY Management System has demonstrated its efficacy as a powerful tool for time series forecasting. By leveraging historical OD request data, the model has provided valuable insights into patterns and trends, enabling administrators to anticipate and prepare for high-demand periods. Its computational efficiency, interpretability, and proven effectiveness in handling non-stationary data make ARIMA an ideal choice for this project. The results achieved with ARIMA not only validate its selection but also highlight its potential for broader applications in time-sensitive operational management. As the system evolves, incorporating real-time data and exploring multivariate predictors could further enhance the model's capabilities, paving the way for even more effective and responsive OD management solutions.

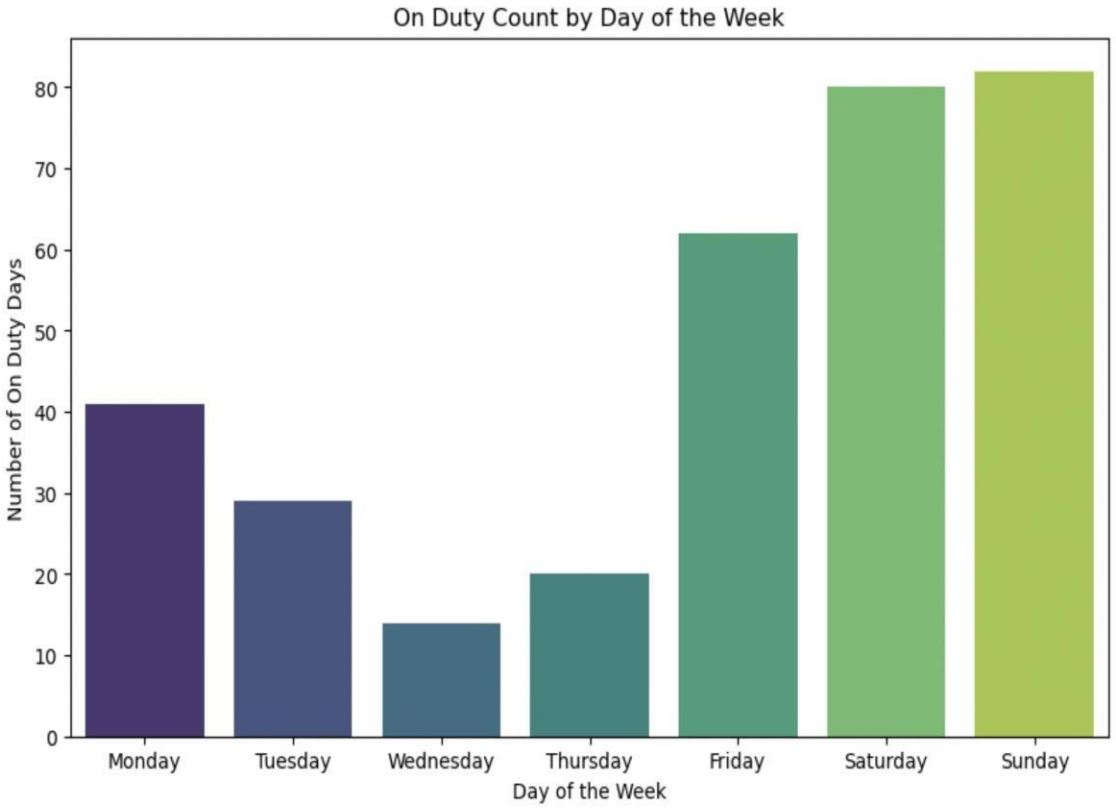


Figure 6.1: Arima Analysis

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The ON DUTY management system successfully addressed the inefficiencies and limitations of the traditional manual process. Through automation, real- time notifications, and a transparent approval workflow, the system greatly improved both student and faculty experiences. The data analytics feature enabled faculty to monitor ON-DUTY patterns and make data-driven decisions, while the post-event validation step ensured accountability in ON- DUTY usage. Despite some challenges, the system demonstrated its potential to significantly enhance the ON-DUTY management process in academic institutions, paving the way for future enhancements.

7.2 FUTURE ENHANCEMENT

Future enhancements for the ON DUTY management system include developing a mobile application to improve accessibility and user experience for both students and faculty. Additionally, implementing advanced data analytics, such as predictive modeling, could provide deeper insights into ON- DUTY patterns and help in making proactive decisions. Integration with other institutional systems, like the student information system, could provide a more holistic view of student performance and ON-DUTY records. Lastly, optimizing the backend for better scalability and performance will ensure the system can handle larger user bases efficiently.

APPENDIX

A.1.FRONTEND

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Animated Login & Registration Form | Codehal</title>
<link rel="stylesheet" type="text/css" href="login1.css">
</head>
<body>
<div class="wrapper">
<div class="form-wrapper sign-in">
<form action="/login" method="POST">
<h2>Login</h2>
<div class="input-group">
<input type="email" name="Email" required>
<label for="Email">Email</label>
</div>
<div class="input-group">
<input type="password" id="login-password" name="Password" required>
<label for="Password">Password</label>
<span class="show-password" onclick="togglePasswordVisibility('login-
password')">👁</span>
</div>
<div class="remember">
```

```
<label><input type="checkbox"> Remember me</label>
</div>
<button type="submit">Login</button>
<div class="signUp-link">
<p>Don't have an account? <a href="#" class="signUpBtn-link">Sign Up</a></p>
</div>
</form>
</div>
<div class="form-wrapper sign-up">
<form action="#" id="signup-form">
<h2>Sign Up</h2>
<div class="input-group">
<input type="email" id="signupEmail" name="Email" required>
<label for="Email">Email</label>
</div>
<div class="input-group">
<input type="password" id="passwordField" name="Password" required>
<label for="Password">Password</label>
<span class="show-password"
onclick="togglePasswordVisibility('passwordField')">👁</span>
</div>
<div class="input-group">
<input type="password" id="confirmPassword" name="ConfirmPassword"
required>
<label for="ConfirmPassword">Confirm Password</label>
```

```
<span class="show-password"
onclick="togglePasswordVisibility('confirmPassword')">👁</span>
</div>
<button type="submit">Sign Up</button>
<div class="signUp-link">
<p>Already have an account? <a href="#" class="signInBtn-link">Sign
In</a></p>
</div>
</form>
</div>
</div>

<script>
//added currently

function togglePasswordVisibility(id) {
const passwordInput = document.getElementById(id);
passwordInput.type = (passwordInput.type === 'password') ? 'text' : 'password';
}

document.querySelector('.signUpBtn-link').addEventListener('click', () => {
</script>
</html>
```

BACKEND

```
const express = require('express'); const mongoose = require('mongoose');
const bodyParser = require('body-parser'); const path = require('path');
const axios = require('axios'); const cors = require('cors');
const nodemailer=require('nodemailer')
const app = express(); // Initialize express app
app.use(cors()); // Apply cors middleware after app initialization
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true })); const port = 3000;
// MongoDB Connection
mongoose.connect('mongodb://localhost:27017/final_student_details')
.then(() => console.log('MongoDB connected successfully'))
.catch(err => console.error('MongoDB connection error:', err));
const studentDetailsDb =
mongoose.createConnection('mongodb://localhost:27017/student_details');
studentDetailsDb.on('error', err => console.error('MongoDB connection
error for student_details:', err));
studentDetailsDb.once('open', () => console.log('MongoDB connected
successfully for student_details'));

// Define Schema
const studentSchema = new mongoose.Schema({ name: String,
registerNumber: String,
```

```
year: String, email: String, cgpa: Number, eventType: String,
collegeName: String, startDate: Date, endDate: Date,
file1: String, file2: String,
description: String,
});
const Student = mongoose.model('Student', studentSchema); const
studentDetailsSchema = new mongoose.Schema({ email: String,
name: String,
});
const StudentDetails = studentDetailsDb.model('twentymembers',
studentDetailsSchema);
// Middleware to serve static files app.use(express.static(path.join(
dirname)));
// Serve HTML file app.get('/', (req, res) => {
res.sendFile(path.join(_dirname, 'gemini.html'));
});
const transporter = nodemailer.createTransport({ service: 'Gmail',
auth: {
user: 'aravinthsubbaiah3@gmail.com', pass:'mheu wpuw gzuz xiha'
}
});
// Send Email Function
async function sendEmail(to, subject, text) { try {
await transporter.sendMail({
from: 'aravinthsubbaiah3@gmail.com', to: to,
subject: subject, text: text
});
console.log('Email sent to:', to);
```

```
} catch (error) {
console.error('Error sending email:', error);
}
}
// Handle form submissions app.post('/register', async (req, res) => { try {
// Log the incoming data console.log('Received data:', req.body);

// Extract form data
const { name, registerNumber, year, email, cgpa, eventType, collegeName,
startDate, endDate, file1, file2, description } = req.body;
// Save to MongoDB
const newStudent = new Student({ name,
registerNumber, year,
email, cgpa, eventType,
collegeName, startDate, endDate, file1,
file2, description,
});
await newStudent.save();
// Synchronize with mentor backend
await axios.post('http://localhost:3001/syncStudent', { name,
registerNumber, cgpa,
startDate, endDate, collegeName, file1,
file2, email
});
res.status(200).json({ message: 'Form submitted successfully!' });
// res.redirect('/success.html');
} catch (error) {
```

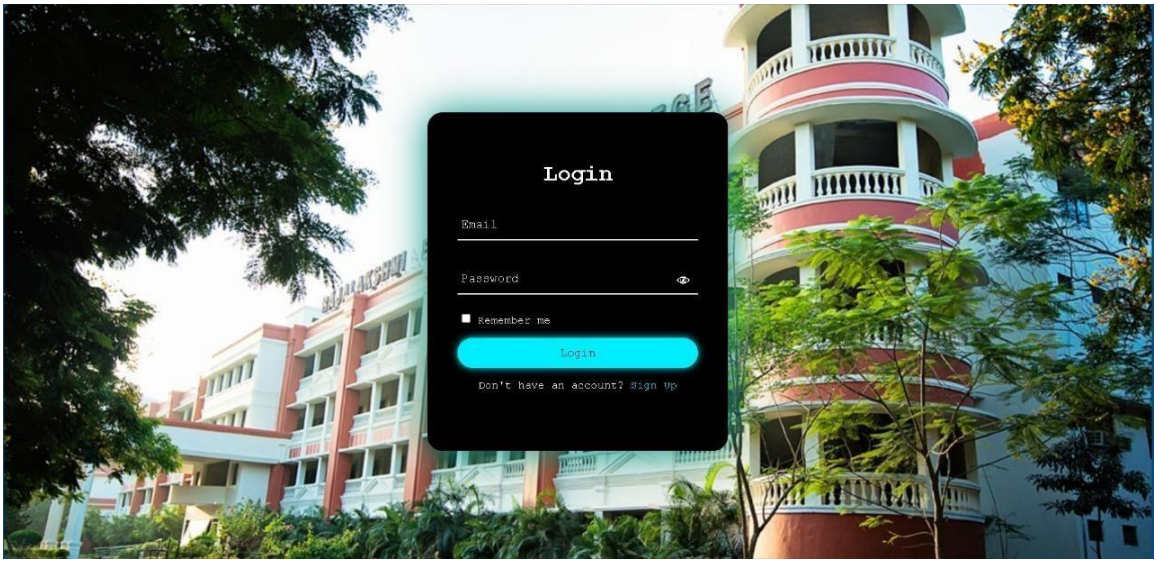
```
console.error('Error submitting form:', error); res.status(500).send('Error
submitting form.');
}
});
app.post('/validateName', async (req, res) => { try {
const { email, name } = req.body;
console.log(`Received for validation - Email: ${email}, Name: ${name}`);

// Normalize the name for comparison
const formattedName = name.trim().toUpperCase();

// Find student details by email
const studentDetails = await StudentDetails.findOne({ email: email });

if (studentDetails) {
const storedName = studentDetails.name.trim().toUpperCase();
console.log(`Stored name: ${storedName}`);
const isValidName = storedName === formattedName;

// Log database details
console.log(`Validation result for email ${email}:`, { email,
storedName,
enteredName: formattedName, isValid: isValidName
});

res.json({ valid: isValidName });
} else {
```

```
console.log(`No student found for email: ${email}`); res.json({ valid: false
});
}
} catch (error) {
console.error('Error validating name:', error); res.status(500).send('Error
validating name.');
}
});
// Retrieve student data for mentor.html app.get('/getStudents', async (req,
res) => { try {
// Fetch students with only specific fields
const students = await Student.find({}, 'name registerNumber cgpa startDate
endDate collegeName file1 file2');
console.log('Fetched students:', students); // Log data for debugging
res.json(students); // Send the filtered data as JSON response
} catch (error) {
console.error('Error retrieving students:', error); res.status(500).send('Error
retrieving students.');
}
});
app.listen(port, () => {
console.log(`Server running at http://localhost:${port}`);
});
```

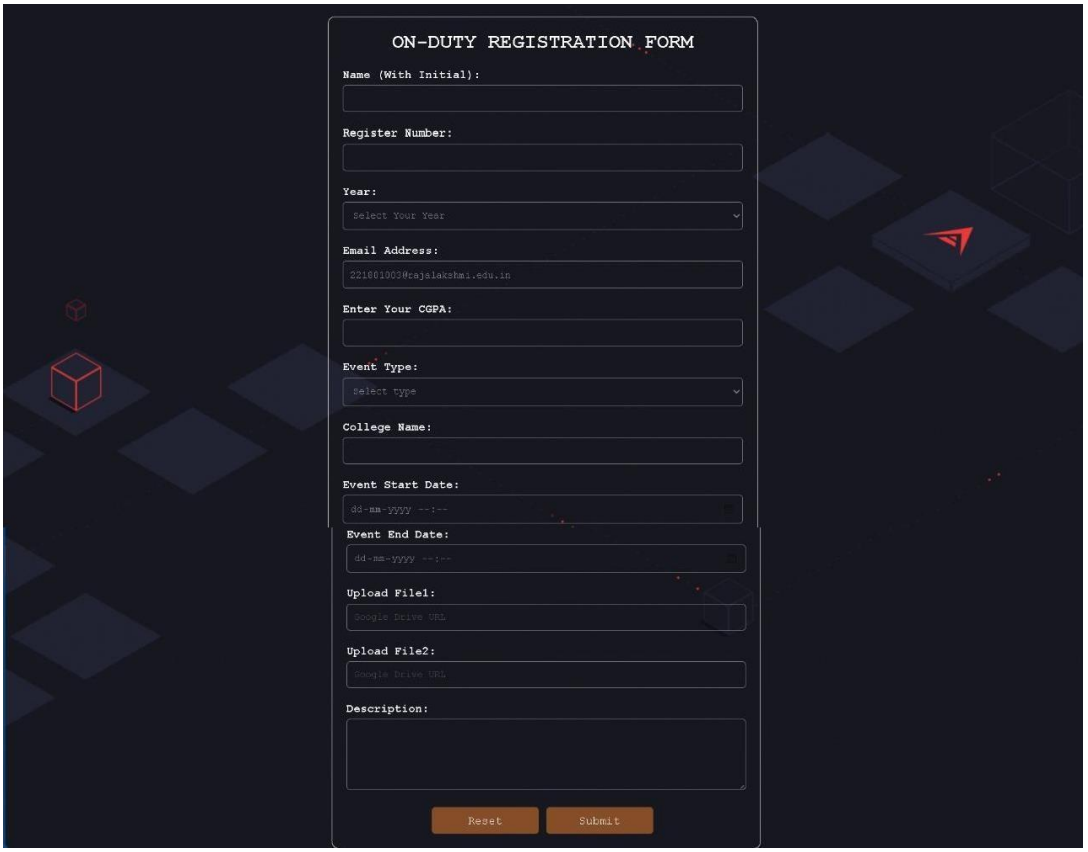
3.ARIMA

```
import pandas as pd import matplotlib.pyplot as plt
 import seaborn as sns # Load the dataset
file_path = '/Users/liogodwinbr/Downloads/student_data.csv' # Replace with your file path
df = pd.read_csv(file_path) # Convert the 'Date' column to datetime format, coercing errors to handle out-ofbounds dates
 df['startDate'] = pd.to_datetime(df['startDate'], errors='coerce') # Drop rows where 'startDate' could not be converted to a valid date
df = df.dropna(subset=['startDate']) # Extract day of the week from the 'Date' column
df['Day_of_Week'] = df['startDate'].dt.day_name() # Calculate the count of 'On Duty' by day of the week 23 od_counts = df['Day_of_Week'].value_counts().reindex([ 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday' ]) # Plot the trend of 'On Duty' counts by day of the week
plt.figure(figsize=(10, 6))
sns.barplot(x=od_counts.index, y=od_counts.values, palette='viridis')
plt.title('On Duty Count by Day of the Week') plt.xlabel('Day of the Week')
plt.ylabel('Number of On Duty Days")
plt.show()
```

A.2 OUTPUT SCREENSHOTS



A.7.4.1 Login Page



A.7.4.2 Registration Form

ARAVINTH S 16:30	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 16:30	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 16:35	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 16:50	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 16:51	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 16:52	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 17:00	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 17:05	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 17:10	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 17:15	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 17:30	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 17:40	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 17:50	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject
ARAVINTH S 17:51	221801003	8	13-Sep-2024 4:59 p.m.	21-Sep-2024 4:59 p.m.	REC	View File1	View File2	0	Approve Reject

Hod

Name	Register No	Event Start Date	Event End Date	College Name	Actions
ARAVINTH.S	03	05-Oct-2024 8:58 a.m.	05-Oct-2024 8:58 a.m.	rec	Approve Reject

A.7.4.3 Multilevel Approval Workflow

The registration for on duty has been **accepted**. Here are the details:

- **Name:** ARAVINTH S
- **Register Number:** 221801003
- **Event Start Date and Time:** 18/7/2024, 8:00:00 am
- **Event End Date and Time:** 19/7/2024, 3:10:00 am

The registration for on duty request has been **rejected**. Here are the details:

- **Name:** ARAVINTH S 14:20
- **Register Number:** 01
- **Event Start Date and Time:** 13/9/2024, 4:23:00 pm
- **Event End Date and Time:** 11/9/2024, 5:24:00 pm

A.7.4.4 Notification

REFERENCES

[1] Devdatta Akhawe, Adam Barth, Peifung E. Lam, John Mitchell, and Dawn Song, "Towards a Formal Foundation of Web Security," University of California, Berkeley.

[2]J. Fonseca, N. Seixas, M. Vieira, and H. Madeira, "Analysis of Field Data on Web Security Vulnerabilities," VOL. 11, NO. 2, MARCH/APRIL 2014.

[3]Faculty of Security Studies, University of Belgrade, 11000 Belgrade, Serbia, "Factors Related to Cyber Security Behavior," received June 27, 2020, accepted July 4, 2020, published July 8, 2020.

[4]Azari, A.; Papapetrou, P.; Denic, S.; Peters, G. "Cellular Traffic Prediction and Classification: A Comparative Evaluation of LSTM and ARIMA." Discovery Science. Springer International Publishing, 2019, pp. 129–144

[5]Zhou, K.; Wang, W.Y.; Hu, T.; Wu, C.H. "Comparison of Time Series Forecasting Based on Statistical ARIMA Model and LSTM with Attention Mechanism." J. Phys. Conf. Ser., 2020, 1631, 012141.

[6]Biswas, A.K.; Ahmed, S.I.; Bankefa, T.; Ranganathan, P.; Salehfar, H. "Performance analysis of short and mid-term wind power prediction using ARIMA and hybrid models." Proceedings of the 2021 IEEE Power and Energy Conference at Illinois (PECI), Urbana, IL, USA, 1–2 April 2021, pp. 1–7.