

BASAVARAJESWARI GROUP OF INSTITUTIONS



**Ballari Institute of Technology & Management**



AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITY JNANASANGAMA,

BELAGAVI 590018

## **INTERNSHIP**

### **Report On**

### **Server Health Monitor**

Submitted in partial fulfillment of the requirements for the award of degree of

## **Bachelor of Engineering**

### **In**

## **COMPUTER SCIENCE AND ENGINEERING – ARTIFICIAL INTELLIGENCE**

### **Submitted by**

**Shubham yadav**

**3BR22CA049**

### **Internship Carried Out**

### **By**

**EZ TRAININGS & TECHNOLOGIES PVT.LTD**

**HYDERABAD**

### **Internal Guide**

**Mr. K Ramesh Kumar**  
Assistant Professor ,CSE

### **External Guide**

**Mr. S SANKARNARAYANAN**

### **BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT**

NACC Accredited Institution\*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"Jnana Gangotri" Campus, No.873/2, Ballari-  
Hospet Road, Allipur, Ballari-  
583104 (Karnataka) (India) Ph: 08392-  
237100/237190, Fax: 08392-237197

**2024-2025**

BASAVARAJESWARI GROUP OF INSTITUTIONS  
**BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT**  
AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITY JNANASANGAMA,  
BELAGAVI 590018



NACC Accredited Institution\*  
(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya  
Technological University, Belagavi)  
"Jnana Gangotri" Campus, No. 873/2, Ballari-Hospet Road, Allipur,  
Ballari-583104 (Karnataka) (India)  
Ph: 08392-237100/237190, Fax: 08392-237197



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING –  
ARTIFICIAL INTELLIGENCE**

# **CERTIFICATE**

This is to certify that the Internship entitled **“Server Health Monitor”** has been successfully completed by **Shubham vadav** bearing USN **3BR22CA049** a bonafide student of Ballari Institute of Technology and Management, Ballari. For the partial fulfillment of the requirements for the **Bachelor’s Degree in Computer Science and Engineering-Artificial Intelligence** of the Ballari Institute of Technology & Management Autonomous Institute Under Visvesvaraya Technology University, Belagavi during the academic year 2024-2025.

**Signature of Internship**

**Co-ordinators**

**Mr. K Ramesh Kumar**  
Assistant Professor ,CSE-AI

**Signature of HOD**

**Dr. Yeresime Suresh**  
Professor & HOD(CSE-AI)

## **DECLARATION**

I, **Shubham yadav**, third year student of Computer Science and Engineering-Artificial Intelligence, Ballari Institute of Technology, Ballari, declare that Internship entitled **Infrastructure Cost Optimizer** is a part of Internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT.LTD, Hyderabad** at **“BITM,BALLARI”**. This report is submitted in partial fulfillment of the requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering-Artificial Intelligence of the Visvesvaraya Technological University, Belagavi.

**Date : 08–03-2025**

**Place : Ballari**

**Signature of the Student**

## **ACKNOWLEDGEMENT**

The satisfactions that a company the successful completion of my internship on “ **Server Health Monitor** ” would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance ,encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to my respective coordinator “**Mr. Ramesh Kumar(Asst.prof,CSE-AI)** ” for their noble gesture ,support co-ordination and valuable suggestions given to me in the completion of Internship.

I also thank **Dr.Suresh Yeresime**, HOD, Department of **Computer Science and Engineering-Artificial Intelligence** for extending all his valuable support and encouragement.

## **Table of Contents**

<b>Chapter No.</b>	<b>Chapter Name</b>	<b>Page No.</b>
<b>1</b>	<b>Company Profile</b>	<b>1</b>
<b>2</b>	<b>Day to day activity(student diary extract)</b>	<b>2</b>
<b>3</b>	<b>Abstract</b>	<b>3</b>
<b>4</b>	<b>Introduction of the project</b>	<b>4</b>
<b>5</b>	<b>Module Description</b>	<b>5-6</b>
<b>6</b>	<b>Algorithm</b>	<b>7-8</b>
<b>7</b>	<b>Output</b>	<b>9</b>
<b>8</b>	<b>Conclusion</b>	<b>10</b>
<b>9</b>	<b>References</b>	<b>11</b>

## CHAPTER-1

**COMPANY PROFILE**

**Company Name : EZ Trainings and Technologies Pvt. Ltd.**

**Introduction:**

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

**Mission:**

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

**Services:****College Trainings:**

- Tailored training programs designed to enhance the employability of students.
- Industry-aligned curriculum covering technical and soft skills.
- Placement assistance and career guidance.

**Development Projects:**

- End-to-end development services, from ideation to execution.
- Expertise in diverse technologies and frameworks.
- Custom solutions to meet specific business needs.

**Locations:** Hyderabad | Delhi NCR

At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence



**CHAPTER-3****ABSTRACT**

The Server Management Dashboard is a real-time monitoring and management system that tracks server performance metrics such as CPU and memory usage. Built using Python, Flask, MongoDB, and Flask-SocketIO, this project enables CRUD operations for server records, provides real-time updates through Socket.IO, and visualizes historical data using Chart.js. The system not only simulates server metric changes every 5 seconds (updating the MongoDB records) but also offers a dynamic web interface for administrators to monitor performance, predict potential failures, and schedule maintenance. This report outlines the architecture, module usage, and functional as well as non-functional requirements of the project, demonstrating its scalability, reliability, and user-friendly design



## INTRODUCTION OF THE PROJECT

In today's rapidly evolving IT landscape, continuous monitoring and proactive management of server infrastructure are essential to ensure optimal performance and minimal downtime. This project, the Server Management Dashboard, addresses these challenges by providing a comprehensive, web-based solution for real-time server monitoring and management.

The system is built using Python and Flask, with MongoDB serving as the primary data storage for server records. It integrates Flask-SocketIO to facilitate real-time communication between the server and the client, allowing updates to critical metrics—such as CPU and memory usage—to be pushed live to the dashboard every 5 seconds. This ensures that administrators have immediate access to the latest server performance data without needing to refresh the entire page.

In addition to basic CRUD operations for managing server data, the dashboard features interactive visualizations powered by Chart.js. These visualizations display historical trends of CPU and memory usage side by side, enabling users to gain deeper insights into server performance over time. The project also incorporates predictive analytics to flag potential server failures and schedules maintenance automatically, thereby facilitating proactive infrastructure management.

Overall, the Server Management Dashboard not only streamlines server monitoring and management but also provides a scalable and user-friendly foundation for future enhancements such as advanced analytics, notifications, and security features.

## CHAPTER-5

**MODULE DESCRIPTION**

The Server Management Dashboard is organized into several interrelated modules, each responsible for specific functionalities:

**1. Server Management Module:**

- Purpose: Handle CRUD operations on server records.
- Functionality:
  - Add new servers with attributes such as name, CPU usage, memory usage, and logs.
  - Retrieve and display server details.
  - Update existing server data.
  - Delete server records from the database.
  - Technology: Flask routes with RESTful endpoints using PyMongo for database interactions.

**2. Real-Time Update Module:**

- Purpose: Simulate and broadcast live server metric updates.
- Functionality:
  - A background thread periodically (every 5 seconds) generates random values for CPU and memory usage.
  - Updates the corresponding MongoDB documents in real time.
  - Emits updates via Flask-SocketIO to connected clients.
  - Technology: Python's threading module, time.sleep(), and Flask-SocketIO.

**3. Data Visualization Module:**

- Purpose: Provide interactive graphical representations of server performance.
- Functionality:
  - Visualize historical CPU and memory usage data using interactive charts.
  - Allow side-by-side comparison of CPU and memory trends.
  - Update charts in real time when new data is received.
  - Technology: Chart.js integrated with JavaScript and HTML5 Canvas.

#### 4. Predictive Analytics & Maintenance Module:

- Purpose: Offer basic predictive insights and schedule maintenance.
- Functionality:
  - Predict potential server failures based on CPU usage thresholds.
  - Generate maintenance schedules (e.g., 7 days ahead) for proactive service management.
  - Technology: Simple rule-based functions using Python's datetime and conditional logic.

#### 5. Database Management Module:

- Purpose: Persist and manage server data.
- Functionality:
  - Utilize MongoDB for storing server records and their real-time metrics.
  - Ensure data consistency during frequent updates.
  - Technology: MongoDB via PyMongo for seamless data handling.

#### 6. User Interface Module:

- Purpose: Provide a user-friendly, interactive web dashboard.
- Functionality:
  - Display a dynamic server list with real-time updates.
  - Offer controls for adding, updating, deleting, and visualizing server data.
  - Include a refresh button to manually update the server list.
  - Technology: HTML, CSS, JavaScript, and integration with Socket.IO for real-time feedback.

## Algorithm

**The following steps outline the process flow for the Server Management Dashboard:**

### **1. Initialization:**

- Establish a connection to the MongoDB database.
- Initialize the Flask application and configure REST API endpoints.
- Set up Flask-SocketIO for real-time communication.
- Start a background thread dedicated to simulating server metric updates.

### **2. Background Metric Simulation:**

- Every 5 seconds, execute the following steps for each server record:
- Generate new random values for CPU and memory usage.
- Update the corresponding server document in MongoDB using an update query.
- Construct an update message containing the server ID, new metric values, and a timestamp.
- Emit the update message via Socket.IO to all connected clients.
- Log the update details in the server console.

### **3. Client-Side Processing:**

- On page load, use a REST API call to fetch the current list of servers and display them on the dashboard.
- Establish a Socket.IO connection to listen for real-time update events.
- When an update event is received:
  - Locate the corresponding server element on the dashboard.
  - Update the displayed CPU and memory usage values.
  - If the server is currently being visualized, append the new data points to the respective Chart.js graphs while maintaining a fixed number of data points (e.g., last 10 updates).

#### 4. User Interactions:

- Adding/Updating/Deleting Servers:

Users interact with form elements to perform CRUD operations. These actions trigger corresponding REST API calls that update MongoDB and refresh the server list.

- Visualizing Server Data:

When a user clicks the "Visualize" button for a server:

- Send a request to the `/server_history/<server_id>` endpoint to retrieve historical data.
- Update the CPU and memory charts with the retrieved data.
- Set the selected server as the current context for real-time chart updates

#### 5. Predictive and Maintenance Functions:

- The system calculates server failure predictions based on a predetermined CPU usage threshold.
- It also generates maintenance schedules by adding a fixed interval (e.g., 7 days) to the current date.
- These calculations are exposed via dedicated REST endpoints for external use or integration into the dashboa

CHAPTER 7

OUTPUT:

Server Management Dashboard

Add Server

Refresh Server List

Server List

shubham

CPU Usage: 87%

Memory Usage: 56%

Delete

Update

Predict Failure

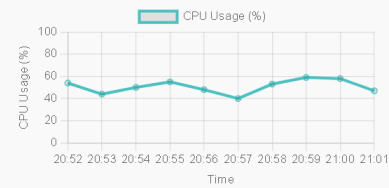
Maintenance Schedule

Visualize

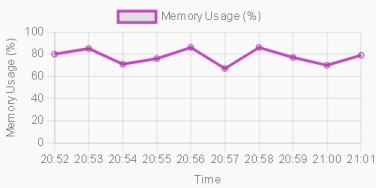
Predict Failure: No  
Maintenance Date: 2025-03-14T21:02:39.826303

Server Data Visualization

CPU Usage for shubham



Memory Usage for shubham



## CHAPTER 8

**CONCLUSION**

In conclusion, the Server Management Dashboard successfully demonstrates a robust solution for real-time monitoring and management of server performance. By integrating Flask for REST API development, PyMongo for database interactions, and Flask-SocketIO for live data updates, the project provides an efficient and scalable platform that updates critical metrics such as CPU and memory usage every 5 seconds directly in MongoDB. This live data is seamlessly pushed to the user interface, ensuring that administrators have immediate access to the latest server information.

Interactive visualizations powered by Chart.js further enhance the user experience by offering detailed, side-by-side charts for both CPU and memory usage. These visual tools allow users to easily track historical trends and quickly identify potential issues, such as servers at risk of failure based on predefined thresholds.

The modular design of the project not only facilitates basic CRUD operations and real-time updates but also lays a solid foundation for future enhancements, including advanced analytics, automated notifications, and role-based access control. Overall, this project delivers a comprehensive, user-friendly solution for proactive server management and maintenance scheduling, ultimately contributing to improved operational efficiency and reduced downtime in IT infrastructures.

CHAPTER-9

## REFERENCES

- <https://github.com/shubhamyadav018>
- <https://flask.palletsprojects.com/>
- <https://pymongo.readthedocs.io/>
- <https://flask-socketio.readthedocs.io/>
- <https://www.chartjs.org/>
- <https://socket.io/>
- <https://docs.mongodb.com/>