



Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

A CHRISTIAN MINORITY RESIDENTIAL INSTITUTION

AICTE Approved & NAAC Accredited

A SKILL BASED EVALUATION REPORT

SUBMITTED BY

SOORYA K (URK22AI1023)

**COURSE CODE
20CS2056**

**COURSE NAME
WEB TECHNOLOGY**

APRIL 2024



**DIVISION OF DIVISION OF DATA SCIENCE AND CYBER
SECURITY**

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

INDUSTRIAL CERTIFICATION



Statement of Achievement

SOORYA K URK22AI1023

has successfully achieved student level credential for completing the JavaScript Essentials 1 course, provided by Cisco Networking Academy in collaboration with OpenEDG JS Institute.

The graduate is able to proficiently:

- Understand the syntax of the core JavaScript language that allows for working with variables, operators, flow control, and functions.
- Understand the basics of the JavaScript data types system, distinguishing between primitive and complex types.
- Think algorithmically and can analyze a problem using a programmatic conceptual apparatus.
- Choose a data type adequate to the problem being solved and use suitable flow control means.
- Design, develop, and improve very simple JavaScript programs.
- Interpret and handle basic exceptions related to errors in program execution.
- Understand a programmer's work in the software development process and the role of fundamental development tools.
- Understand how a program is interpreted and executed in an actual computer environment, local or remote.



Scan to Verify

January 24, 2024

Laura Quintana
Vice President and General Manager
Cisco Networking Academy



Statement of Achievement

SOORYA K URK22AI1023

has successfully achieved student level credential for completing the JavaScript Essentials 2 course, provided by Cisco Networking Academy in collaboration with OpenEDG JS Institute.

The graduate has studied:

- Techniques for constructing and modifying objects, including the use of prototypes and inheritance.
- Methods for defining and encapsulating class properties and managing array data, including JSON conversion.
- Utilization of the Math object and regular expressions for mathematical and string operations.
- Advanced function techniques and asynchronous programming, including callbacks and iterators.
- Problem analysis and program development using algorithmic thinking and object-oriented principles.



Scan to Verify

January 24, 2024

Laura Quintana
Vice President and General Manager
Cisco Networking Academy

HOSTEL COMPLAINT MANAGEMENT SYSTEM

A REAL TIME APPLICATION REPORT

Submitted by

SOORYA K (URK22AI1023)

HARIHARAN K (URK22AI1048)



DIVISION OF DATA SCIENCE AND CYBER SECURITY

**KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed-to-be-under Sec-3 of the UGC Act,
1956) Karunya Nagar, Coimbatore - 641 114. INDIA**

APRIL 2024

PROJECT REVIEW FORM

Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
MoE, UGC & AICTE Approved
NAAC A++ Accredited
School of Computer Science & Technology
Division of Computer Science & Engineering

Skill-Based Evaluation - Project Zero Review Form

Course Code & Name : **20CS2056- Web Technology**
Academic Year : **2023-24** Semester: **Even Semester**

#	Reg No	Name	Sign
1	URK22AI1023	SOORYA K	<i>su</i>
2	URK22AI1048	HARIHARAN K	<i>vld</i>

Title of the project : Compliant Rising System

Abstract of the project :

The project is to create a website for managing complaints about various maintenance-related concerns for a hostel. The website gives students an easy-to-use way to report issues, monitor their progress, and get in touch with the hostel administration. The website aims to increase facilitate communication and streamline the complaint process. The website will have a polished and organized design and be developed on an appropriate platform. It will have areas dedicated to tracking complaints, submitting complaints, (FAQs). The user-friendly navigation will make it simple for users to get the features they want. The project entails creating the required features, including user access limits, tracking systems for complaints, complaint reporting forms, and security precautions to safeguard user information. To guarantee optimal operation and usability across all devices and browsers, extensive testing will be carried out. The website will be internally launched to the hostel inhabitants upon completion. Users will receive material and training to help them become comfortable with this Idea. To fix issues that have been reported and improve the user experience, regular maintenance and upgrades will be carried out. The hostel's complaint resolution procedure will operate much more efficiently thanks to this website. It will facilitate communication, make filing complaints easier, and let consumers efficiently follow the development of their complaints. This website will be able to better attend to maintenance issues and give the residents a better quality of life by putting this system into place.

Technology Stack

Front End: HTML, CSS, JAVASCRIPT, JQUERY, ANGULARJS
Back End: NODEJS
Database: MONGODB

Approved / Not Approved *✓*

Faculty Signature *[Signature]*

ABSTRACT

The Hostel Complaint Management System (HCMS) is a web-based platform designed to streamline the process of registering, tracking, and resolving complaints within a hostel environment. Developed using a combination of HTML, CSS, JavaScript, AngularJS, Node.js, MongoDB, and Bootstrap, HCMS provides a user-friendly interface for students to report various issues such as maintenance problems, water leaks, electrical faults, and room-related concerns.

The system features a responsive frontend interface built with AngularJS and Bootstrap, allowing students to easily submit complaints, view their complaint history, and track the status of ongoing issues. Node.js serves as the backend, handling HTTP requests and responses, and connecting to a MongoDB database to store complaint data securely.

HCMS incorporates user authentication and authorization mechanisms to ensure that only registered students and authorized hostel management staff can access and manage complaint data. The system includes separate dashboards for students and hostel management, each tailored to their specific needs. Students can submit complaints, provide additional information if needed, and receive updates on the status of their complaints. Hostel management can view all complaints, assign them to relevant staff members, and monitor the resolution process.

Overall, the Hostel Complaint Management System aims to improve communication and transparency in addressing hostel-related issues, ultimately enhancing the living experience for students and promoting efficient management of hostel facilities.

PROBLEM STATEMENT

Hostels serve as temporary homes for students, providing accommodation and facilities essential for their daily living and academic pursuits. However, maintaining hostel facilities and addressing issues promptly is a constant challenge faced by hostel management. The traditional approach of managing complaints through manual processes, such as paper forms or in-person reporting, often leads to inefficiencies, delays, and communication barriers. To address these challenges and enhance the overall living experience for residents, there is a pressing need for a modern Hostel Complaint Management System (HCMS) that streamlines the process of registering, tracking, and resolving complaints.

Inefficient Complaint Registration: Students encounter difficulties in reporting issues due to the lack of convenient and accessible complaint registration mechanisms. Paper-based forms or in-person reporting require additional effort and time, discouraging students from reporting minor issues promptly.

Limited Communication Channels: The communication between students and hostel management regarding the status of complaints is often inadequate. Students are left in the dark about the progress of their complaints, leading to frustration and dissatisfaction with the hostel's responsiveness to their concerns.

Manual Tracking and Prioritization: Hostel management relies on manual methods for tracking and prioritizing complaints, resulting in delays and inconsistencies in addressing issues. The absence of a centralized system for managing complaints makes it challenging to allocate resources efficiently and address urgent matters promptly.

Data Fragmentation and Inaccuracy: The lack of a centralized database for storing complaint data results in fragmentation and duplication of records. Manual data entry increases the likelihood of errors and inaccuracies, undermining the reliability of complaint tracking and reporting.

Security and Privacy Concerns: Traditional complaint management systems may lack robust security measures to protect sensitive student information. Unauthorized access to complaint records or data breaches could compromise student privacy and confidentiality.

METHODOLOGY / ARCHITECTURE

Methodology

The methodology for developing the Hostel Complaint Management System (HCMS) involves a structured approach to ensure efficient implementation and successful delivery of the project. The chosen methodology should prioritize collaboration, iterative development, and continuous feedback to address evolving requirements and deliver a high-quality solution. An agile methodology such as Scrum or Kanban would be suitable for this project due to its flexibility and emphasis on incremental delivery. Here's an outline of the proposed methodology:

Project Planning: Define project goals, objectives, scope, and deliverables. Identify stakeholders and establish communication channels for regular updates and feedback.

Requirements Gathering: Conduct interviews and surveys to gather requirements from students and hostel management. Prioritize requirements based on their importance and feasibility.

Iteration Planning: Break down requirements into smaller, manageable tasks or user stories. Estimate the time and effort required for each task and prioritize them for implementation.

Iterative Development: Implement features incrementally, focusing on delivering the most valuable functionality first. Hold regular sprint meetings to review progress, discuss any challenges, and adjust priorities as needed.

Testing and Quality Assurance: Conduct thorough testing at each iteration to identify and address any defects or issues. Perform both manual and automated testing to ensure the reliability and stability of the system.

Deployment and Feedback: Deploy the system to a staging environment for user acceptance testing (UAT). Gather feedback from students and hostel management and incorporate any necessary changes or enhancements.

Iterative Improvements: Continuously monitor and evaluate the system's performance and usability. Iterate on feedback and make improvements based on user experience and changing requirements.

Architecture

The architecture of the Hostel Complaint Management System (HCMS) should be designed to be scalable, secure, and maintainable. It should incorporate both frontend and backend components to provide a seamless user experience and efficient data management. Here's a proposed architecture for HCMS:

Frontend Components:

HTML, CSS, JavaScript: Responsible for creating the user interface and handling client-side interactions.

AngularJS: A framework for building dynamic web applications, used for implementing interactive features and data binding.

Bootstrap: A front-end framework for designing responsive and mobile-friendly interfaces, ensuring consistency and usability across different devices.

Backend Components:

Node.js: A JavaScript runtime environment for building scalable and high-performance server-side applications. Used for handling HTTP requests, routing, and business logic implementation.

MongoDB: A NoSQL database for storing complaint data in a flexible and scalable manner. Utilizes document-based storage for easy retrieval and manipulation of data.

Integration and Communication:

RESTful APIs: Define and implement a set of RESTful APIs for communication between the frontend and backend components. These APIs should support CRUD operations (Create, Read, Update, Delete) for managing complaints and user authentication.

Infrastructure and Deployment:

Cloud Hosting Platform: Deploy the application to a cloud hosting platform such as AWS (Amazon Web Services) or Heroku for scalability and reliability.

Containerization: Use containerization technology such as Docker to package the application and its dependencies into portable containers for easy deployment and management.

Continuous Integration/Continuous Deployment (CI/CD): Implement CI/CD pipelines to automate the build, test, and deployment processes, ensuring rapid and reliable delivery of updates and improvements

IMPLEMENTATION – CODING AND OUTPUT SCREENSHOT

Index.html

```
<!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">
    <link rel="stylesheet" href="index.css">
    <script src="index.js"></script>
    <title>Login</title>
  </head>
  <body>
    <div id="a3"></div>
    <div class="a">
      <div class="a2">
        <div class="a22">
          <h1>LOGIN</h1>
          <div class="tbox">
            <label for="u">REG NO</label>
            <input type="text" id="u" placeholder="Reg_no">
          </div>
          <div class="tbox">
            <label for="p">DOB</label>
            <input type="password" id="p" placeholder="Date of birth">
          </div>
          <button class="b1" onclick="login()">Login</button>
        </div>
      </div>
    </div>
  </body>
</html>
```

Room.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<link rel="stylesheet" href="home.css">
<script src="room.js"></script>
<title>Document</title>
<style>
.form-label {
    color: #ffe7e7;
}
.btn-primary {
    background-color: #F4BC1C;
    border-color: #F4BC1C;
}
.btn-primary:hover {
    background-color: #F4BC1C;
    border-color: #F4BC1C;
}
.table {
    background-color: #3a3a3a;
}
.table th,
.table td {
    color: #FFFFFF;
}
#cc1 {
    border: 2px dashed #ccc;
    border-radius: 10px;
    padding: 20px;
    margin-top: 20px;
}
</style>
</head>
<body>
```

```
<div class="a3">
```

```
<a href="index.html"></a>

<div class="pt-5"></div>
<div class="pt-5"></div>

<div><div class="container" id="cc1">
    <a href="hist.html" class="pr-5" style="color: #F4BC1C;">History</a>
    <a href="room.htm" id="n" style="color: #F4BC1C;">Complaint</a>
</div><br><form action="http://localhost:5000/room1" method="post">
    <div class="container">
        <div class="row">
            <div class="col-md-6">
                <div class="form-pair">
                    <label for="hn" class="form-label">Hostel Name :</label>
                    <input type="text" class="form-control" name="hn" required placeholder="Hostel Name" id="hn">
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-pair">
                    <label for="fn" class="form-label">Floor No :</label>
                    <input type="text" class="form-control" name="fn" required placeholder="Your Room No" id="fn">
                    <label for="rn" class="form-label">Room No :</label>
                    <input type="number" class="form-control" name="rn" required placeholder="Enter Room No" id="rn">
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-pair">
                    <label for="name" class="form-label">Name :</label>
                    <input type="text" class="form-control" name="name" required placeholder="Enter Your Name" id="name">
                    <label for="phn" class="form-label">Contact :</label>
                    <input type="tel" class="form-control" name="phn" required placeholder="Enter Your Phone Number" id="phn" pattern="[0-9]{10,}" minlength="10" maxlength="10">
                </div>
            </div>
        </div>
    </div>
</form>
```


Output Screenshot

The screenshot shows the Karunya University login interface. At the top left is the university logo. The main area is titled "LOGIN". It contains two input fields: "REG NO" with the value "URK22AI1023" and "DOB" with a masked input field showing "*****-**-**". Below these is a yellow "LOGIN" button.

The screenshot shows the "COMMON" section of the Karunya University application. It includes fields for "Hostel Name" (AR), "Floor No" (2), "Room No" (229), "Name" (SOORYA), "Contact" (9943051504), and "Reg No" (URK22AI1023). A "Complaint" field contains "FAN PROBLEM". Below the form are buttons for "Image", "Choose File", and "Screenshot ... 48.47PM". A "Submit" button is at the bottom.

The screenshot shows a table titled "ROOM" under the "COMMON" section. The table lists complaints from AR students. The columns are Hostel Name, Floor No, Room No, Name, Contact, Reg No, and Additional Info.

Hostel Name	Floor No	Room No	Name	Contact	Reg No	Additional Info
AR	2	229	SOORYA	9943051504	URK22AI1023	FAN PROBLEM
AR	2	229	HARI	7092003788	URK22AI1048	SOCKET PROBLEM

Hostel Name : AR

Floor : 2

Complaint: BLUB

Image Choose File Screenshot ... 48.47 PM

Submit

MongoDB Compass - localhost:27017/hostel.rooms

localhost:27017 ...

My Queries Soorya Employee Databases hostel rooms

hostel > rooms

Documents 5 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1 - 2 of 2

```

_id: ObjectId('66196e3eba4fa02befc2d2aa')
hn: "AR"
fn: "2"
rn: "229"
name: "SOORYA"
phn: "9943051594"
regn: "URK22A11023"
additionalInfo: "FAN PROBLEM"
__v: 0

_id: ObjectId('66196e8dba4fa02befc2d2aa')
hn: "AR"
fn: "2"
rn: "229"
name: "HARI"
phn: "7092003788"
regn: "URK22A11048"
additionalInfo: "SOCKET PROBLEM"
__v: 0

```

CONCLUSION

The Hostel Complaint Management System (HCMS) project represents a significant step forward in addressing the challenges associated with traditional complaint management processes in hostel environments. By leveraging modern web technologies and adopting an agile methodology, HCMS aims to streamline the process of registering, tracking, and resolving complaints, ultimately enhancing the living experience for hostel residents and promoting efficient management of hostel facilities.

Efficient Complaint Registration: HCMS provides students with a user-friendly interface for conveniently registering complaints online. The system allows students to provide detailed information about the nature and location of the issue, facilitating prompt resolution by hostel management.

Transparent Communication: HCMS fosters transparency and accountability by enabling real-time tracking and updates on the status of complaints for both students and hostel management. Through the centralized dashboard, students can monitor the progress of their complaints and receive notifications when issues are resolved.

Streamlined Management: Hostel management benefits from HCMS's centralized dashboard, which allows for efficient tracking, prioritization, and assignment of complaints. The system provides tools for allocating resources effectively and ensuring timely resolution of issues, thereby improving overall operational efficiency.

Data Integrity and Security: HCMS maintains data integrity and security by utilizing a centralized database for storing complaint data and implementing robust authentication and authorization mechanisms. Student information is protected from unauthorized access, ensuring privacy and confidentiality.

Enhanced User Experience: By incorporating responsive design principles and intuitive user interfaces, HCMS delivers an enhanced user experience for both students and hostel management. The system's accessibility and ease of use contribute to higher levels of user satisfaction and engagement.

EVALUATION SHEET

Reg.No : URK22AI1023

Name: SOORYA K

Course code: 20CS2056

Course Name: Web Technology

S.No	Rubrics	Maximum Marks	Marks Obtained
1	Industrial Certification	10	
2	Real – Time Application Design	30	
	Total	40	

Rubrics	Excellent	Good	Average	Below Average
Design of website				
Input validation				
Integration of front end and back end				
Presentation and Viva				
GitHub repository				
Report				

Signature of the Faculty-in-charge