Inter Departmental Collaboration Hub

1. Company Profile – GUVI HCL

GUVI HCL is a collaborative initiative between GUVI (Grab Ur Vernacular Imprint) and HCL Technologies, designed to bridge the gap between academic learning and industry skills through hands-on technical training and real-world exposure.

GUVI, an edtech platform incubated by IIT Madras and IIM Ahmedabad, was founded in 2014 with the mission of making technology education accessible to everyone in vernacular languages such as Tamil, Telugu, Hindi, and Kannada. Headquartered in Chennai, India, GUVI has empowered over 10 lakh learners through online courses, coding bootcamps, and career programs. The platform specializes in programming, fullstack development, artificial intelligence, cloud computing, and data science, offering training aligned with current IT industry needs.

HCL Technologies, a global technology leader with decades of experience in IT services and consulting, partners with GUVI to offer industry-relevant programs like the GUVI–HCL Tech Career Program and HCL Career Launchpad. Through this collaboration, students gain exposure to enterprise-level technologies, mentorship from HCL professionals, and opportunities to work on real-time industrial projects.

GUVI–HCL partnership focuses on transforming aspiring students into skilled and jobready IT professionals by integrating theoretical learning with practical implementation. Together, they aim to create a new generation of tech talent that is proficient, confident, and ready to contribute to India's fast-growing digital and innovation ecosystem.

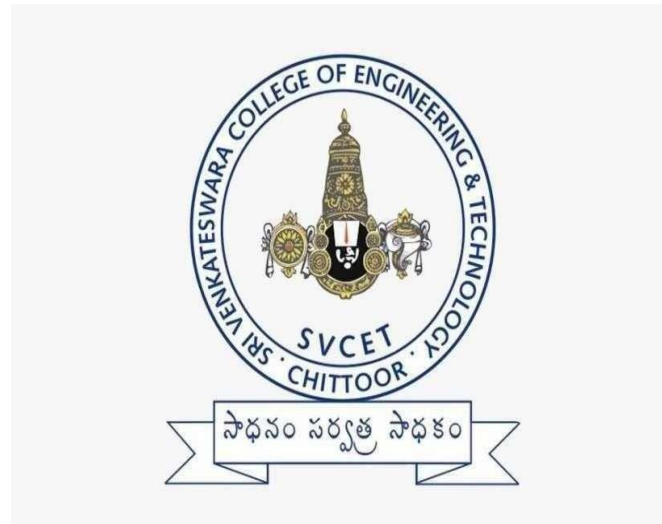# SRI VENKATESWARA COLLGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

R.V.S. Nagar, Chittoor-517127.(A.P)

(Approved by AICTE, New Delhi, Affiliated to JNTUA, Anantapur)

(Accredited by NBA, New Delhi C NAAC, Bangalore)

(An ISO 9001:2000 Certified Institution)

2025-2026

This is to certify that the project report submitted by NEERUGATTI KARTHIK Regd.No.: 22781A0588, is the original work carried out by him during the Academic year 2025-2026 as part of the Java Training Program conducted by HCL GUVI

P. Ragavan   P. Jyotheeswari

Technical Trainer   Head of the department

(COMPUTER SCIENCE & ENGINEERING)

Abstract

In today's dynamic institutional and organizational environment, efficient coordination and collaboration between departments are crucial for achieving common goals and improving operational efficiency. However, managing inter-departmental communication, shared projects, and related data manually can be time-consuming and prone to mismanagement.

The Inter Departmental Collaboration Hub, developed in **Java**, addresses this challenge by providing an automated solution to manage and streamline departmental collaboration efficiently. The system allows multiple departments within an institution to store and access collaboration records through a centralized **MongoDB** database, enabling seamless data sharing and real-time interaction. It supports essential operations such as **adding, updating, deleting, and viewing departmental collaborations**, along with analytical features to measure collaboration frequency and engagement.

By integrating a doubly linked list data structure with database operations, the project ensures efficient data management, real-time synchronization, and quick access to departmental records. This approach enhances both performance and data reliability, making the collaboration process smoother and more transparent. The application demonstrates the practical implementation **of Java programming, object-oriented principles, and MongoDB connectivity** in building scalable, datadriven management systems.

Index

9	Output screenshot/output	15
10	Conclusion	16

Aim:

The main aim of the *Inter Departmental Collaboration Hu*b is to create a Java-based interactive platform that facilitates seamless communication, coordination, and data sharing among multiple departments within an institution.This system enables real-time collaboration, project tracking, and data management using an integrated MongoDB database and in-memory data structures like Doubly Linked Lists.It ensures efficient storage, retrieval, and modification of department-related collaboration records in a centralized environment.

Algorithm:

1. **Start the program**.

2. **Establish a connection** to the MongoDB database (collaboration_db) and access the collection (departments).

3. **Load existing department collaboration** records from MongoDB into a **Doubly Linked List** (DLL) for efficient in-memory data handling.

4. **Display the main menu** with the following options:

    1. Add Department Collaboration

    2. Display All Collaborations

    3. Search Collaboration by Department ID

    4. Update Collaboration Details

    5. Delete Collaboration

6. Analyze Collaboration Count per Department

7. Exit

5. **Accept the user's choice** and perform the corresponding operation:

6.1 If choice = 1 (Add Department Collaboration):

1. Prompt for Department ID, Department Name, Collaborating Department, Project Title.

2. Create a new collaboration node and insert it at the end of the DLL.

3. Insert the same record into the MongoDB collection.

4. Display a success message confirming insertion.

6.2 If choice = 2 (Display All Collaborations):

1. Traverse the DLL from head to tail.

2. Display all collaboration details (ID, Department Name, Partner Department, Project Title).

3. If the list is empty, display "No collaborations available."

6.3 If choice = 3 (Search Collaboration by Department ID):

1. Traverse the DLL to find a matching Department ID.

2. If found, display the collaboration details.

3. If not found, show "Department not found."

6.4 If choice = 4 (Update Collaboration Details):

1. Search for the Department ID in the DLL.

2. If found, prompt for updated Partner Department or Project Title.

3. Modify both the DLL node and the MongoDB document.

4. Display a success message.

6.5 If choice = 5 (Delete Collaboration):

1. Search for the Department ID in the DLL.

2. If found, remove the node from the DLL.

3. Delete the corresponding record from MongoDB.

4. Display a confirmation message.

6.6 If choice = 6 (Analyze Collaboration Count per Department):

1. Traverse the DLL and count the number of collaborations for each department.

2. Display the department name and corresponding collaboration count.

3. Highlight the department with the highest number of collaborations.

6.7 If choice = 7 (Exit):

1. Display an exit message.

2. Close the MongoDB connection and terminate the program.

System Requirements:

The Real-Time Customer Feedback Aggregator is a Java-based application integrated with a MongoDB database. It requires both hardware and software components to function incidently. The following are the minimum and recommended system requirements for developing and executing the project.

Hardware Requirements:

| Component	Minimum Requirement	Recommended Requirement
Processor	Intel Core i3 or equivalent	Intel Core i5 / i7 or higher
RAM	4 GB	8 GB or more
Hard Disk	500 MB free space	1 GB free space
Monitor	1024 × 768 resolution	1920 × 1080 resolution
Keyboard & Mouse	Standard input devices	Standard input devices

Software Requirements:

Component	Specification
Operating System	Windows 10 / 11, Linux, or macOS
Programming Language	Java (JDK 8 or higher)
Database	MongoDB (version 4.0 or higher)
IDE / Editor	IntelliJ IDEA, Eclipse, or NetBeans
Build Tool (optional)	Apache Maven or Gradle
Driver Library	MongoDB Java Driver (mongodb-driver-sync)
Additional Libraries	org.bson and com.mongodb.client packages
Testing Tool	MongoDB Compass / Command Prompt

Output:
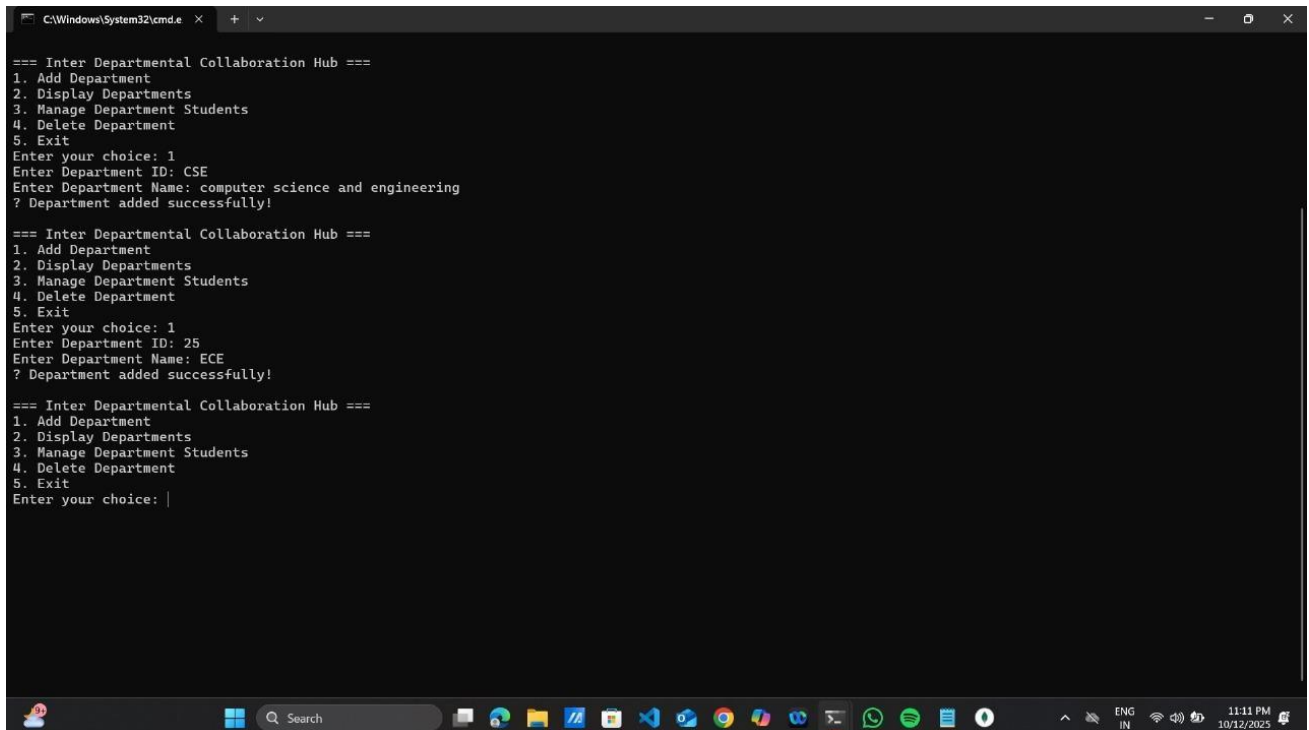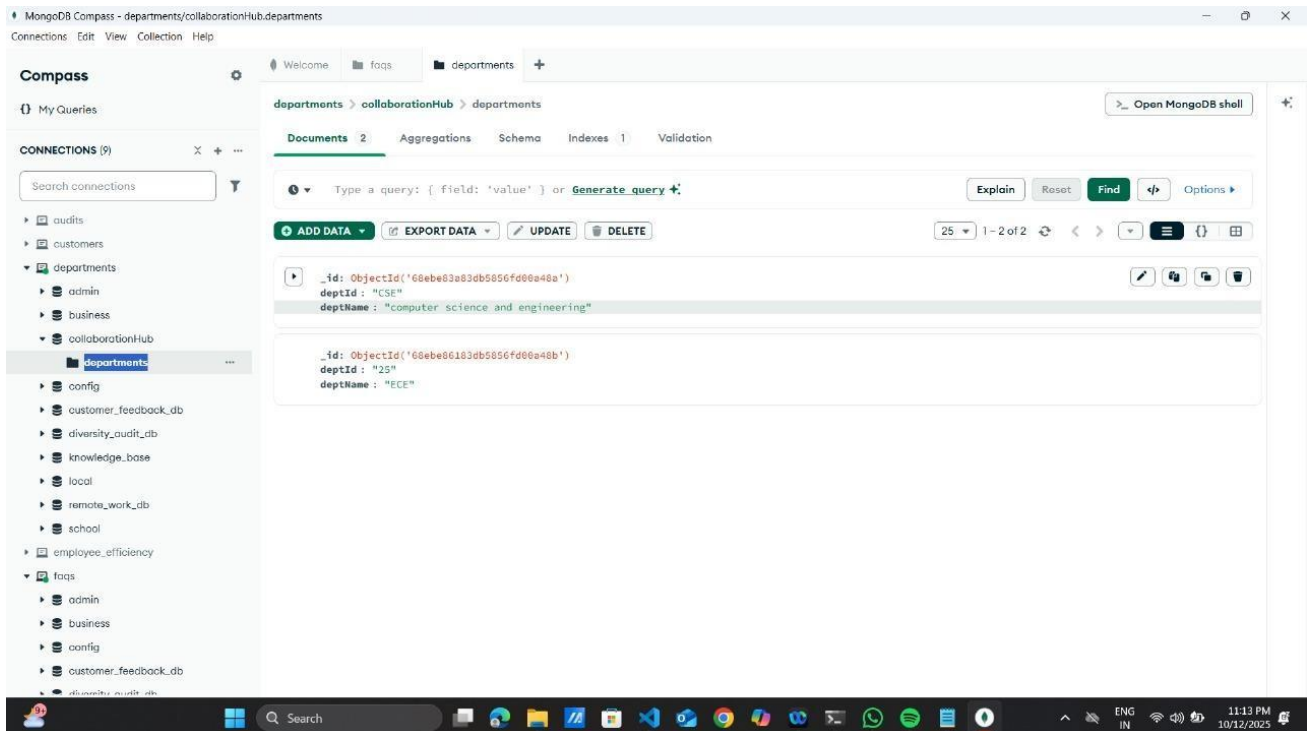
Conclusion:

The Inter Departmental Collaboration Hub successfully demonstrates how technology can enhance coordination, communication, and data management across various departments within an organization or institution. By integrating **Java** with a **MongoDB** database and employing a **doubly linked list** structure, the system provides a reliable and efficient platform for managing interdepartmental collaborations in real time.

The project not only simplifies the process of adding, updating, and retrieving departmental data but also ensures transparency and accessibility of shared information. Through its structured approach and interactive console-based interface, the application showcases the practical application of **objectoriented programming concepts, database connectivity**, and **data structure utilization.**

Overall, this system serves as a scalable foundation for future enhancements, such as adding user authentication, notifications, and analytics dashboards. It contributes toward building a more connected, collaborative, and efficient organizational environment, paving the way for smarter communication and better decision-making.