

# Customizable Customer Journey Mapper



Skill Up. Level Up

## 1. Company Profile - GU VI HCL

GU VI HCL is a collaborative initiative between GU VI (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.

GU VI, 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, GU VI has empowered over 10 lakh learners through online courses, coding bootcamps, and career programs. The platform specializes in programming, full-stack 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 GU VI to offer industry-relevant programs like the GU VI-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.

The GU VI-HCL partnership focuses on transforming aspiring students into skilled and job-ready 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 COLLEGE 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 "Internship training report" submitted by **HARSHITHAGANGARAPU (Regd.No.:22781A0538)** is work done here and submitted during 2024 – 2025 academic year in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**, at the **HCL GUVI**.

**Mr. P. RAGAVAN**

**GUVI - HCL**

**(TECHNICAL TRAINER)**

**Dr. P. JYOTHEESWARI**

**Head of The Department**

**Department Of CSE**

# **INDEX**

<b>S.NO</b>	<b>CONTENT</b>	<b>Page No</b>
1	Abstract	4
2	Aim	5
3	Algorithm	6
4	System requirements	7
5	Program code	8-11
6	Output screenshot/output	12
7	conclusion	13

## ABSTRACT

In the modern business environment, customer experience has emerged as a key differentiator that directly influences brand loyalty, conversion rates, and overall business performance. However, organizations often struggle to gain a holistic understanding of how customers interact with their products or services across various channels. The **Customizable Customer Journey Mapper** project addresses this challenge by developing an intelligent and adaptive tool that enables businesses to visualize, analyze, and optimize the entire customer journey in a dynamic and personalized manner.

The proposed system allows users to design customer journeys through an interactive interface that supports drag-and-drop functionality for defining touchpoints, stages, and customer personas. Unlike traditional static mapping tools, this solution offers **customizability**, enabling users to tailor journeys to specific industries, marketing strategies, or customer segments. It integrates with external data sources such as CRM systems, web analytics platforms, and customer feedback databases to ensure that the journey maps are grounded in real-time behavioral data.

Additionally, the system includes built-in **analytics and reporting modules** that identify key performance indicators (KPIs), detect pain points, and recommend actionable insights to enhance user experience. Through data visualization and AI-driven insights, the tool empowers decision-makers to better understand customer emotions, motivations, and barriers at each interaction stage.

## AIM

To develop a customizable Customer Journey Mapper using Java and MongoDB that helps businesses visualize and manage customer experience stages. The program enables users to add, view, edit, and delete journey stages dynamically.

It provides flexibility to define unique touchpoints and descriptions for each stage. The goal is to create an interactive, data-driven system for tracking and improving customer engagement.

## OBJECTIVE

1. **To design an interactive and customizable platform** that enables businesses to create, visualize, and modify customer journey maps according to specific goals and customer segments.
2. **To integrate real-time data sources** such as CRM systems, analytics tools, and customer feedback platforms for accurate and up-to-date journey mapping.
3. **To utilize data analytics and AI-driven insights** to identify pain points, customer behavior patterns, and areas for experience improvement.
4. **To enhance decision-making and collaboration** among marketing, product, and customer service teams through a centralized, visual, and easy-to-use dashboard.
5. **To improve overall customer satisfaction, retention, and loyalty** by enabling organizations to design personalized and seamless customer experiences.
6. **To provide scalable and industry-flexible solutions** that can adapt to different business domains, including e-commerce, banking, healthcare, and education.

# ALGORITHM

1. **Start the program** and connect to the MongoDB database (customer\_journey\_db) and collection (journey\_stages).
2. **Display a menu** with options to add, view, edit, delete, or exit.
3. **Accept user input** to choose an operation.
4. **Perform the selected CRUD operation:**
  - Add → Insert a new stage document.
  - View → Retrieve and display all stages.
  - Edit → Update details of a chosen stage.
  - Delete → Remove a stage by name.
5. **Repeat the menu** until the user selects Exit, then **close the database connection** and end the program.
6. **Start the program** and connect to the MongoDB database (customer\_journey\_db) and collection (journey\_stages).
7. **Display a menu** with options to add, view, edit, delete, or exit.
8. **Accept user input** to choose an operation.
9. **Perform the selected CRUD operation:**
  - Add → Insert a new stage document.
  - View → Retrieve and display all stages.
  - Edit → Update details of a chosen stage.
  - Delete → Remove a stage by name.
10. **Repeat the menu** until the user selects Exit, then **close the database connection** and end the program.

# SYSTEM REQUIREMENTS

## Software Requirements:

- **Operating System:** Windows 10 / 11 or Linux
- **Programming Language:** Java (JDK 8 or higher)
- **Database:** MySQL
- **IDE:** Eclipse / NetBeans / IntelliJ IDEA
- **Connector:** MySQL JDBC Driver

## Hardware Requirements:

- **Processor:** Intel i5 or higher
- **RAM:** Minimum 8 GB
- **Storage:** Minimum 500 GB
- **Display:** 1366 × 768 resolution

## PROGRAM CODE

```
import com.mongodb.client.*;
import com.mongodb.client.model.Filters;
import org.bson.Document;

import java.util.Scanner;

public class CustomerJourneyMongo {
    private static final String DATABASE_NAME = "customer_journey_db";
    private static final String COLLECTION_NAME = "journey_stages";

    public static void main(String[] args) {
        try (MongoClient mongoClient =
MongoClients.create("mongodb://localhost:27017")) {
            MongoDatabase database =
mongoClient.getDatabase(DATABASE_NAME);
            MongoCollection<Document> collection =
database.getCollection(COLLECTION_NAME);

            Scanner sc = new Scanner(System.in);
            int choice;

            do {
                System.out.println("\n--- Customer Journey Mapper (MongoDB) ---");
                System.out.println("1. Add a new stage");
                System.out.println("2. View all stages");
                System.out.println("3. Edit a stage");
                System.out.println("4. Delete a stage");
                System.out.println("5. Exit");
                System.out.print("Enter your choice: ");
                choice = sc.nextInt();
            }
        }
    }
}
```

```

sc.nextLine();

switch (choice) {
    case 1:
        addStage(sc, collection);
        break;
    case 2:
        viewStages(collection);
        break;
    case 3:
        editStage(sc, collection);
        break;
    case 4:
        deleteStage(sc, collection);
        break;
    case 5:
        System.out.println("Exiting... Thank you!");
        break;
    default:
        System.out.println("Invalid choice!");
}
} while (choice != 5);
}
}

```

```

private static void addStage(Scanner sc, MongoCollection<Document>
collection) {
    System.out.print("Enter Stage Name: ");
    String name = sc.nextLine();
    System.out.print("Enter Description: ");
    String desc = sc.nextLine();
    System.out.print("Enter Touchpoint: ");
    String touch = sc.nextLine();

```

```

    Document stage = new Document("stageName", name)
        .append("description", desc)

```

```

.append("touchpoint", touch);

collection.insertOne(stage);
System.out.println("☑ Stage added successfully to MongoDB!");
}

private static void viewStages(MongoCollection<Document> collection) {
    System.out.println("\n📋 Customer Journey Stages:");
    try (MongoCursor<Document> cursor = collection.find().iterator()) {
        int count = 0;
        while (cursor.hasNext()) {
            Document stage = cursor.next();
            count++;
            System.out.println(count + ". Stage: " + stage.getString("stageName"));
            System.out.println(" Description: " + stage.getString("description"));
            System.out.println(" Touchpoint: " + stage.getString("touchpoint"));
            System.out.println();
        }
        if (count == 0)
            System.out.println("No stages found.");
    }
}

private static void editStage(Scanner sc, MongoCollection<Document>
collection) {
    System.out.print("Enter the stage name to edit: ");
    String stageName = sc.nextLine();

    Document existingStage = collection.find(Filters.eq("stageName",
stageName)).first();
    if (existingStage == null) {
        System.out.println("Stage not found!");
        return;
    }

    System.out.print("Enter new description (leave blank to keep): ");
}

```

```

String desc = sc.nextLine();
System.out.print("Enter new touchpoint (leave blank to keep): ");
String touch = sc.nextLine();

if (!desc.isEmpty())
    existingStage.put("description", desc);
if (!touch.isEmpty())
    existingStage.put("touchpoint", touch);

collection.replaceOne(Filters.eq("stageName", stageName), existingStage);
System.out.println("☑ Stage updated successfully!");
}

private static void deleteStage(Scanner sc, MongoCollection<Document>
collection) {
    System.out.print("Enter the stage name to delete: ");
    String stageName = sc.nextLine();

    long deletedCount = collection.deleteOne(Filters.eq("stageName",
stageName)).getDeletedCount();
    if (deletedCount > 0)
        System.out.println("🗑 Stage deleted successfully!");
    else
        System.out.println("Stage not found.");
}
}

```

# OUTPUT

```
C:\Windows\System32\cmd.e x + v
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh\Downloads\customer>javac -cp ".;mongodb-driver-sync-4.11.0.jar;mongodb-driver-core-4.11.0.jar;bson-4.11.0.jar" CustomerJourneyMapper.java
C:\Users\harsh\Downloads\customer>java -cp ".;mongodb-driver-sync-4.11.0.jar;mongodb-driver-core-4.11.0.jar;bson-4.11.0.jar" CustomerJourneyMapper

--- Customer Journey Mapper ---
1. Add a new stage
2. View all stages
3. Edit a stage
4. Delete a stage
5. Exit
Enter your choice: 2
No stages in the journey yet.

--- Customer Journey Mapper ---
1. Add a new stage
2. View all stages
3. Edit a stage
4. Delete a stage
5. Exit
Enter your choice: 1
Enter Stage Name: chittoor
Enter Description: college
Enter Customer Touchpoint: maps
? Stage added successfully!

--- Customer Journey Mapper ---
1. Add a new stage
2. View all stages
3. Edit a stage
4. Delete a stage
5. Exit
Enter your choice: 3
Customer Journey Map:
Stage 1:
Stage: chittoor
Description: college
Touchpoint: maps

Enter stage number to edit: 1
Enter new Stage Name (chittoor): tirupati
Enter new Description (college): college
Enter new Touchpoint (maps): google
? Stage updated successfully!

--- Customer Journey Mapper ---
1. Add a new stage
2. View all stages
3. Edit a stage
4. Delete a stage
5. Exit
Enter your choice: |
```

MongoDB Compass - localhost:27017/customer\_journey\_db.journey\_stages

Connections Edit View Collection Help

**Compass**

My Queries

CONNECTIONS (1)

localhost:27017

- admin
- company
- config
- customer\_journey\_db
  - journey\_stages
- integrationHub
- libraryDB
- local
- marketing
  - customers

journey\_stages

localhost:27017 > customer\_journey\_db > journey\_stages

Documents 3 Aggregations Schema Indexes 1 Validation

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

[EXPLAIN](#) [Reset](#) [Find](#) [Options](#)

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

25 1-3 of 3

```
_id: ObjectId('68ebd2ce62b17b7c849801c1')
stageName: "ygeygo"
description: "Customer compares products"
touchpoint: "website"
```

```
_id: ObjectId('68ec68f633c5b524b368a665')
stageName: "chittoor "
description: "college"
touchpoint: "maps"
```

```
_id: ObjectId('68ec69ad136cb65b3791bda2')
stageName: "Awareness"
description: "Customer becomes aware of our product"
touchpoint: "social media"
```

# CONCLUSION

The Customizable Customer Journey Mapper serves as an innovative solution for businesses aiming to better understand and enhance their customers' experiences across multiple touchpoints. By combining data analytics, visualization, and user-driven customization, the system provides a comprehensive view of the customer's path, helping organizations identify pain points, optimize engagement, and tailor services to meet individual needs. Its intuitive interface and integration with real-time data sources make it a powerful tool for improving decision-making and aligning customer interactions with business objectives.

Overall, this project demonstrates how technology can transform the traditional approach to customer experience management. By offering a flexible and data- driven platform, the Customizable Customer Journey Mapper empowers organizations to design more effective, personalized, and seamless journeys. This not only improves customer satisfaction and loyalty but also supports long-term business growth through continuous insight-driven improvement.