National College of Computer Studies Tribhuvan University

**IOST**



**Project Report On**

**“Survey System”**

**Submitted To:**

### National College of Computer Studies Department of Bachelor of Science in Computer Science and Information Technology (BSc.CSIT)

### Paknajol, Kathmandu

**In partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology (BSc. CSIT)**

**Submitted By:** Atullya Maharjan CSIT 4th Semester Roll No: 5

### Sec: B

# Abstract

The Survey System is a software solution developed in Java that provides an engaging and interactive experience for users taking surveys. Unlike traditional methods that use paper forms, this system uses modern technology to allow users to complete surveys online. This removes the need for physical paper forms, making it easier and more convenient.

The main goal of this system was to create an easy-to-use and efficient platform for surveys, improving the way people gather and share opinions. The development of this system shows the strength of Java programming in creating tools that improve user experiences.

A key feature of the Survey System is that it gives participants immediate results after they complete surveys. This quick feedback makes the experience more interactive and encourages more people to take part. It helps collect data and insights effectively.

In summary, the Survey System achieves its goal of providing a simple, effective, and enjoyable survey experience.

# Acknowledgement

I am sincerely grateful to **NCCS College** for providing me with the resources and support to undertake this project. I extend my heartfelt thanks to **Mr. Sumit Ghishing**, our esteemed lecturer for their invaluable guidance and expertise. Their continuous support has shaped the direction and quality of our work.

I also want to express my appreciation to my family, friends, and all the participants for their unwavering encouragement and contributions, which have been a constant source of motivation throughout this endeavor.

In conclusion, I am deeply thankful to NCCS College, Mr. Sumit Ghishing, and everyone involved for making this project a reality. Your support has been instrumental in its success.

Thanking You, Atullya Maharjan

CSIT 4th SEM (11th Batch)

**Table of Contents**

[Abstract i](#_Toc166934853)

[Acknowledgement ii](#_Toc166934854)

[Chapter 1: Introduction 1](#_Toc166934855)

[1.1 Introduction to the project 1](#_Toc166934856)

[1.2 Problem Statement 1](#_Toc166934857)

[1.3 Objectives 1](#_Toc166934858)

[Chapter 2: Implementation 2](#_Toc166934859)

[2.1 Tools used 2](#_Toc166934860)

[2.2 Implementation detail of module 2](#_Toc166934861)

[Chapter 3: Source Code 4](#_Toc166934862)

[Chapter 4: Expected Output 8](#_Toc166934863)

[Chapter 5: Conclusion 9](#_Toc166934864)

[References 10](#_Toc166934865)

# Chapter 1: Introduction

## Introduction to the project

The Survey System is software created using Java and SQL that improves how surveys are conducted and managed. It helps simplify the entire process from setting up surveys to filling them out, doing away with old paper-based methods and manual data handling.

The primary aim of the Survey System is to offer a straightforward and effective way for users to participate in surveys. It provides various tools, such as creating new surveys, adding questions, and answering surveys. Users can also see their results immediately after completing surveys, which adds to the convenience.

Moreover, the system emphasizes security and proper management of data.. The Survey System is designed to make survey participation not just easier but also more secure and enjoyable for all involved.

## Problem Statement

Before the Survey System was developed, managing surveys was often a clumsy and inefficient process. It involved extensive paperwork and significant time commitments, frequently resulting in issues like data loss. This old method made it difficult to keep survey records updated and secure.

The Survey System addressed these issues by introducing a secure and user-friendly platform. Additionally, the system limits record modifications to authorized administrators only, which helps reduce paperwork, save time, and decrease paper usage.

## Objectives

* To simplify the creation and management in surveys by eliminating outdated manual and paper-based method.
* To provide a user-friendly platform that allows users to easily access and complete surveys.
* To secure all survey data by implementing strict access controls.
* To offer immediate results and feedback to users upon completion of surveys.

# Chapter 2: Implementation

## Tools used

**Visual Studio Code:** The Survey System project is developed using the Java programming language as its primary tool for implementation. Java was chosen for versatility and strong community support.

In addition, the project utilizes Visual Studio Code (VS Code) as the integrated development environment (IDE) for efficient code writing, debugging, and management. Further, MySQL was also used for storing and managing the data efficiently.

## Implementation detail of module

The Survey System developed in Java allows users to create, participate in, and manage surveys efficiently. Below is a concise overview of key classes and methods involved in the system:

a) **createTable Function**: This method is responsible for creating tables in the database if they do not exist. It takes a SQL query string as an argument, prepares a statement, and executes the SQL command to ensure that the necessary table structure is in place.

b) **registerUser Function**: This function handles the registration of new users in the system. It collects user input for username, password, role, and address, then inserts these details into the `Users` table in the database. It uses prepared statements to protect against SQL injection.

c) **loginUser Function**: This method facilitates user login. It prompts the user to enter their username and password, checks these credentials against the `Users` table, and returns a User object if the credentials are correct. This function ensures secure login using prepared statements.

d) **createSurvey Function**: This function manages the creation of new surveys. It captures the survey title and description from user input, inserts these into the Surveys table, and handles the creation of associated questions. It iterates over user-provided questions, adding each to the `Questions` table linked to the survey's ID.

e) **participateInSurvey Function**: This method enables users to participate in an existing survey. It displays questions from the specified survey one by one, collects answers from the user, and records these answers in the `Responses` table, linking each response with the user ID and question ID.

f) **viewSurveyResults Function**: This method is used to view the results of a specific survey. It queries the database to retrieve and display all responses associated with each question of the survey, providing insights into how participants have responded.

These functions and methods form the backbone of the Survey System, allowing it to function as a comprehensive tool for survey management, participation, and analysis.

# Chapter 3: Source Code

import java.sql.\*;

import java.util.Scanner;

public class DatabaseMa {

public static void main(String[] args) {

System.out.print("\033[H\033[2J");

System.out.flush();

Connection connection = null;

Scanner scanner = new Scanner(System.in);

try {

String url = "jdbc:mysql://localhost:3306/my\_db";

String userName = "root";

String password = "root";

connection = DriverManager.getConnection(url, userName, password);

createTable(connection, "CREATE TABLE IF NOT EXISTS Users (ID INT AUTO\_INCREMENT PRIMARY KEY, Username VARCHAR(255), Password VARCHAR(255), Role ENUM('admin', 'user') NOT NULL, Address VARCHAR(255))");

createTable(connection, "CREATE TABLE IF NOT EXISTS Surveys (SurveyID INT AUTO\_INCREMENT PRIMARY KEY, Title VARCHAR(255), Description TEXT)");

createTable(connection, "CREATE TABLE IF NOT EXISTS Questions (QuestionID INT AUTO\_INCREMENT PRIMARY KEY, SurveyID INT, QuestionText TEXT, FOREIGN KEY (SurveyID) REFERENCES Surveys(SurveyID))");

createTable(connection, "CREATE TABLE IF NOT EXISTS Responses (ResponseID INT AUTO\_INCREMENT PRIMARY KEY, QuestionID INT, UserID INT, Answer TEXT, FOREIGN KEY (QuestionID) REFERENCES Questions(QuestionID), FOREIGN KEY (UserID) REFERENCES Users(ID))");

System.out.println("1. Login\n2. Register");

System.out.print("Choose an option: ");

int initChoice = Integer.parseInt(scanner.nextLine());

User user = null;

if (initChoice == 1) {

user = loginUser(scanner, connection);

} else if (initChoice == 2) {

registerUser(scanner, connection);

user = loginUser(scanner, connection);

} if (user == null) {

System.out.println("Authentication failed!");

return;

}

while (true) {

System.out.println("1. Create Survey");

System.out.println("2. Participate in Survey");

System.out.println("3. View Survey Results");

System.out.println("4. Exit");

System.out.print("Choose an option: ");

int choice = Integer.parseInt(scanner.nextLine());

switch (choice) {

case 1:

if ("admin".equals(user.getRole())) {

createSurvey(scanner, connection);

} else {

System.out.println("Only admin can create surveys.");

}

break;

case 2:

participateInSurvey(scanner, connection, user.getId());

break;

case 3:

viewSurveyResults(scanner, connection);

break;

case 4:

System.out.println("Exiting program.");

return;

default:

System.out.println("Invalid choice. Please choose again.");

}

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

if (connection != null) connection.close();

scanner.close();

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

private static void createTable(Connection connection, String sql) throws SQLException {

try (PreparedStatement preparedStatement = connection.prepareStatement(sql)) {

preparedStatement.executeUpdate();

}

}

private static User loginUser(Scanner scanner, Connection connection) throws SQLException {

System.out.println("Enter Username:");

String username = scanner.nextLine();

System.out.println("Enter Password:");

String password = scanner.nextLine();

String sql = "SELECT ID, Username, Role FROM Users WHERE Username = ? AND Password = ?";

try (PreparedStatement preparedStatement = connection.prepareStatement(sql)) {

preparedStatement.setString(1, username);

preparedStatement.setString(2, password);

try (ResultSet resultSet = preparedStatement.executeQuery()) {

if (resultSet.next()) {

return new User(resultSet.getInt("ID"), username, resultSet.getString("Role"));

} else {

System.out.println("Login failed, please check your username and password.");

}

}

}

return null;

}

private static void createSurvey(Scanner scanner, Connection connection) throws SQLException {

System.out.println("Enter Survey Title:");

String title = scanner.nextLine();

System.out.println("Enter Survey Description:");

String description = scanner.nextLine();

String sqlInsertSurvey = "INSERT INTO Surveys (Title, Description) VALUES (?, ?)";

try (PreparedStatement preparedStatement = connection.prepareStatement(sqlInsertSurvey, Statement.RETURN\_GENERATED\_KEYS)) {

preparedStatement.setString(1, title);

preparedStatement.setString(2, description);

preparedStatement.executeUpdate();

try (ResultSet generatedKeys = preparedStatement.getGeneratedKeys()) {

if (generatedKeys.next()) {

long surveyId = generatedKeys.getLong(1);

System.out.println("Enter number of questions:");

int numQuestions = Integer.parseInt(scanner.nextLine());

for (int i = 0; i < numQuestions; i++) {

System.out.println("Enter question " + (i + 1) + ":");

String questionText = scanner.nextLine();

addQuestionToSurvey(connection, surveyId, questionText);

}

}

}

}

System.out.println("Survey created successfully!");

}

private static void participateInSurvey(Scanner scanner, Connection connection, int userId) throws SQLException {

System.out.println("Enter Survey ID to participate:");

int surveyId = Integer.parseInt(scanner.nextLine());

String sqlSelectQuestions = "SELECT QuestionID, QuestionText FROM Questions WHERE SurveyID = ?";

try (PreparedStatement preparedStatement = connection.prepareStatement(sqlSelectQuestions)) {

preparedStatement.setInt(1, surveyId);

try (ResultSet resultSet = preparedStatement.executeQuery()) {

if (!resultSet.isBeforeFirst()) {

System.out.println("No questions found for this survey.");

return;

}

while (resultSet.next()) {

long questionId = resultSet.getLong("QuestionID");

String questionText = resultSet.getString("QuestionText");

System.out.println(questionText);

System.out.println("Enter your answer:");

String answer = scanner.nextLine();

recordResponse(connection, questionId, userId, answer);

}

}

private static void viewSurveyResults(Scanner scanner, Connection connection) throws SQLException {

System.out.println("Enter Survey ID to view results:");

int surveyId = Integer.parseInt(scanner.nextLine());

String sqlSelectResponses = "SELECT Q.QuestionText, R.Answer FROM Responses R INNER JOIN Questions Q ON R.QuestionID = Q.QuestionID WHERE Q.SurveyID = ?";

try (PreparedStatement preparedStatement = connection.prepareStatement(sqlSelectResponses)) {

preparedStatement.setInt(1, surveyId);

try (ResultSet resultSet = preparedStatement.executeQuery()) {

if (!resultSet.isBeforeFirst()) {

System.out.println("No responses found for this survey.");

return;

}

while (resultSet.next()) {

String questionText = resultSet.getString("QuestionText");

String answer = resultSet.getString("Answer");

System.out.println(questionText + ": " + answer);

}

}

}

}

}

class User {

private int id;

private String username;

private String role;

public User(int id, String username, String role) {

this.id = id;

this.username = username;

this.role = role;

}

public int getId() {

return id;

}

public String getUsername() {

return username;

}

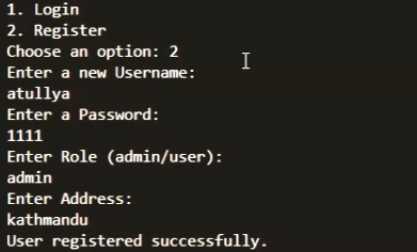
public String getRole() {

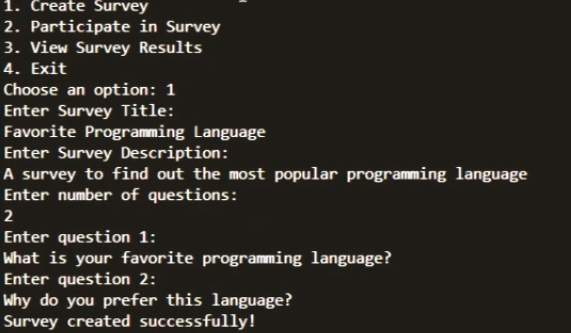
return role;

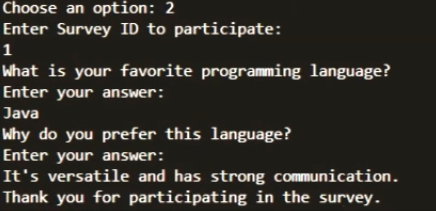
}

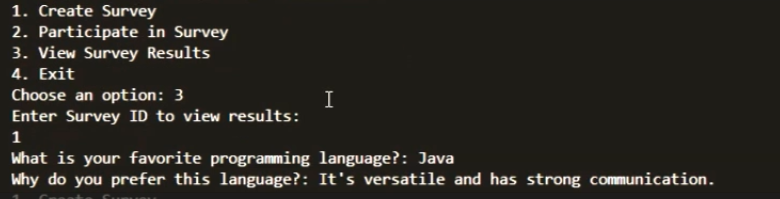
}

# Chapter 4: Expected Output









# Chapter 5: Conclusion

In conclusion, the Survey System project has successfully met its goals by providing a straightforward and effective platform for creating, managing, and participating in surveys. This system, built using Java and SQL, has made the survey process much easier and more secure. Users can easily register, create surveys, and see survey responses in real time. The successful implementation of these features shows the importance of careful planning, effective system design, and thorough testing in software development. Overall, the Survey System has proven to be a valuable tool for efficiently gathering information and improving decision-making processes.

# References

* https://copyassignment.com/survey-management-system-in-java/
* https://github.com/topics/online-survey-system
* https://www.scribd.com/document/559928617/Online-Survey-System