**A MINOR PROJECT SYNOPSIS**

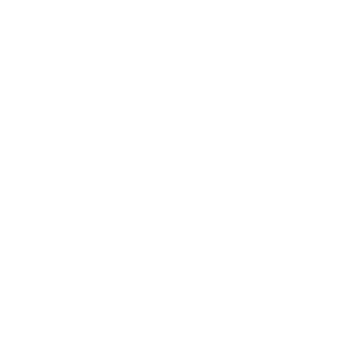
**ON**

Hotel Management System

**BACHELOR OF COMPUTER APPLICATIONS**

Degree

Session 2020-21





**DEPARTMENT OF BCA**

**BEACON INSTITUTE OF TECHNOLOGY**

**MEERUT**



**SUBMITTED TO** **SUBMITTED BY**

Lucky Berwal Ajay Kumar (180968106011)

Chetan Gupta (180968106025)

Manish Sharma (180968106039)

Priya Rana(180968106051)

**TO WHOM IT MAY CONCERN**

This is to certify that Ajay Kumar, Chetan Gupta, Manish Sharma, Priya Rana from Department of BCA, Beacon Institute of Technology, and Meerut has carried out the project at our organization from 25-Dec-2020 to 10-Jan-2021.

Under training period, they have worked on the project entitled Hotel Management Project under the guidance of Mrs. Lucky Berwal. During this project, he/she has been an active member of the team. Ajay Kumar’s specific role in the project was Front-end, Chetan Gupta’s specific role in the project was Back-end, Manish Sharma and Priya Rana’s specific role in the project was Requirement Gathering.

The work carried by him/her is satisfactory and wish his /her all the best for future assignments.

Authorized Signatory

Name………………….

Designation ……………….

|  |
| --- |
|  |

**TABLE OF CONTENTS**

(i) Acknowledgement

(ii) Declaration/ Certificate

(iii) Abstract

1. Introduction

1.1Existing system and its limitations/ Problem Statement

1.2 Proposed system – Scope and Objective

1.3 Organization of the project report in accordance with SDLC – chapter overview

2. Process Model

2.1 Model Description

2.2 Architecture

2.3 Development Methodology

3. Requirement Analysis

3.1 Requirement Elicitation

3.1.1 Use Case Scenarios

3.1.2 Use Case Diagrams

3.2 Feasibility Study

3.2.1 Technical Feasibility (based on hardware & software availability)

3.2.2 Operational Feasibility (based on ease of understanding and use)

3.2.3 Economical Feasibility (based on cost estimation)

3.3 Requirement Analysis of the Proposed System

3.3.1 Context Level DFD

3.3.2 Level-1 DFDs

3.3.3 Level-2 DFDs

3.3.4 Data Dictionary

3.3.5 E-R Diagrams

3.4 Software Requirement Specifications (SRS)

3.5 Hardware and Other Requirements

4. System Design

4.1 Design Approach – Top Down/ Bottom Up (Whichever applicable)

4.2 Design Methodology – Function Oriented/ Object Oriented

4.3 Module and Sub-Module Description

4.4 User Interface Design - Snapshots

5. Coding

5. Hard Code

6. Testing

6.1 Methodology of Testing used in the project

6.2 Test Cases

6.3 Steps of Testing of Module

7. Implementation/Maintenance

7.1 Installation Procedure

7.2 Do and Don’ts

8. Conclusion

9. Limitations and Future Works

10. Bibliography and References

Page No.

6

6

6

7

8

8

8

8

9

9

9

10

11

11

11

11

12

12

12

13

14

16

17

17

18

18

18

18

20

22

22

38

38

38

39

40

40

51

52

53

54

**ACKNOWLEDGEMENT**

Praise and thanks to Almighty God for the success of this project work. It is essential at this point to acknowledge with thanks those who have made useful contributions to facilitate the successful accomplishment of this academic task.

We wish to express or reserved gratitude to our supervisor, Mrs. Lucky Berwal for his comments and suggestions for expanding of knowledge on this our research work.

Our deepest appreciation goes to our friends and course mates and everyone who helped us throughout the course of writing this project.

Finally, while we acknowledge the contribution of all, we claim total responsibility for whatever shortcomings the project may have or contain. Perfection belongs only to Almighty God.

**Ajay Kumar**

**Chetan Gupta**

**Manish Sharma**

**Priya Rana**

**ABSTRACT**

Hotel Management System is a software system where the management of entire hotel is computerized. The hotel management system is designed using JAVA Swing as the rich GUI for front end and MySQL Server as the secured backend database.

In this project the details are maintained like customer details, employee details, booking details and billing details. Booking the rooms, vacating the rooms, the restaurant management, billing process, staff management etc all is computerized and the management is done without any difficulty.

The reports can be viewed completely and the head of the management daily or weekly or monthly can review it. For company auditing it will be more useful. This Proposed System will be interactive, faster and user-friendly for the end users. Using the hotel management system, the following activities can be performed.

* + Room Service
  + Check In
  + Check Out
  + Staff Master
  + Restaurant
  + Login

**Introduction**

**1.1Existing system and its limitations/ Problem Statement**

The phase of system analysis process deals with problems that are affecting the current manual system. The problems are those, which are affecting the hotel in its daily routine work. As the growing trend in most business in InfoTech World of Computers, need of accuracy, perfectness, speed and high memory data storage is a must. Each and every problem must be solved with least amount of time and energy.

The problems faced by the existing system and hope to be solved by the Hotel Management System are described below:

* Difficulty in maintenance of records
* Time consuming
* Editing of data becomes a tedious job
* No security of data
* Lack of efficiency
* Data redundancy
* Data inconsistency
* Incidence of Fraud

**1.2 Proposed system – Scope and Objective**

The scope of the proposed system is to provide solutions to the problems stated above and help the user to manage the hotel effectively and efficiently through:

* Adequate Record Keeping :

To eliminate manual record keeping and install an electronic record keeping thereby ensuring adequate record of transactions are kept. This ensures a centralized system where all necessary data and information can easily be accessed, Tracked, and monitored

* Reduced Incidence of Fraud :

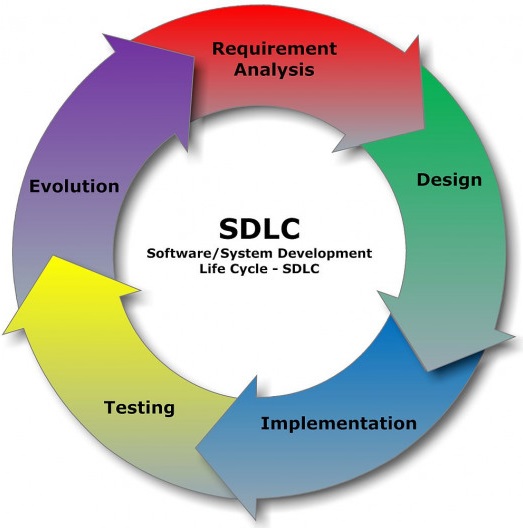
The program is envisaged to reduce the incidence of fraud both by staff and outsiders through proper record keeping, tracking and monitoring of transaction operations in the organization.

* Maximum Accountability: To Instill accountability in the process of management in the hotel organization by not only reducing incidence of fraud, but also eliminate wastages.
* Provide Data Security: The study will install security measures by providing different access levels to various staff.
* Effective Resource Management: The Human Resource module (HR) and Finance & Account (F&A) module will enable effective utilization of financial and human resources by comparing the accounts receivable with the account payable and complete record of personnel through the nominal roll module will enhance staff deployment and productivity.
* Increased Profit Line for The Organization: when there is reduced incidence of fraud this and proper management of resources this will enable increased profit line for the organization.
* Reduced Time Consumption: A good search algorithm will be implemented on the web application to enhance the search facility whereby users of the system can search for all kinds of data using various criteria.

The system can be handy to the user in the following ways:

* To automate the work such as gathering information, gathering Hotel Staff information, Workers’ roster, food ordering and Hotel administration in general.
* To atomize different types of reports.
* Removal of Data Redundancy.
* To create a centralized system where all necessary data and information can be accessed easily.
* Data Consistency.

**1.3 Organization of the project report in accordance with SDLC – chapter overview**



**Process Model**

**2.1 Model Description**

The Hotel Management system is developed to overcome the most of the problems occurring in the manual system by computerizing the existing system. The features of the newly proposed computerized system are described in brief below:

* After computerizing the system, the Hotel Administrator can finish their work in the least amount of time possible and with very little effort. The computerized system has many gains and efforts which the manual system cannot give in any type of situation.
* In any manual system if we take the main problem arising is to maintain the number of records and finding a particular record, computerized systems are most helpful in dealing with areas where database comes into existence.
* A computer can hold large amount of data in storage devices and it can operate at a very high speed. The user can input all types of information into the computer and can be able to perform any type of task which when done manually is tedious and time consuming.

**2.2 Architecture**

A system architecture using only one single integrated model for the description of multiple views such as planning, requirement analysis, design, implementation, deployment, structure, behaviour, input data and output data views is a kind of system model.

**2.3 Development Methodology**

This section involves data gathering. Area of study, the data quality controls provided for the study. It also illustrates how the data was analyzed and presented.

The swing refers to the method to be used to create the graphical user interface with Java Swing application. Rather than writing numerous lines of code describe the appearance and location of interface elements, you simply drag and drop pre-built objects into a place on screen.

Swing revolves around ready-made objects and it is event-driven that is all the activities in a program are triggered by one event or another. Each object has its own properties, determining its size, color, appearance and nature of its text and much more. Each object also has its own event-Handling procedures.

Swing also handles images, menus, dialog boxes, drive and directory list and much more. The application will be desktop-based using the java platform. MySQL is used as the backend where Customer Record is stored. It serves as the Database.

**Requirement Analysis**

**3.1 Requirement Elicitation**

In Order for the goals of the automated system to be achieved the design of the HMIS takes the following into consideration:

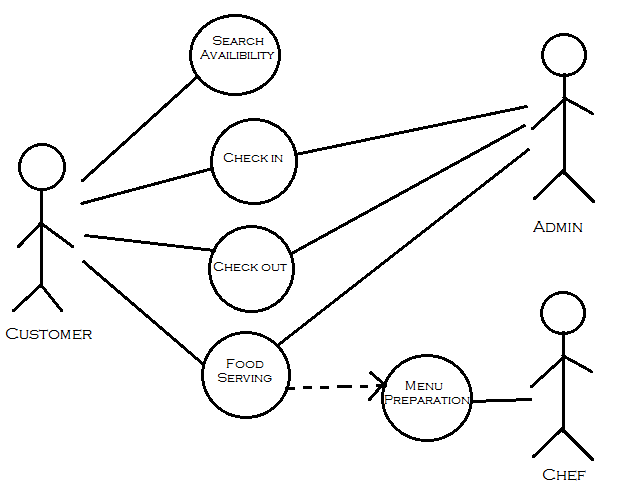
* The system must make the hotel services fully known to the customer such as the room details and pricing.
* The system must be able to search databases or records to provide quick result based on users query.
* The system should ensure data consistency and no duplication of data no matter how small.
* The system must be accessed only by authorized persons and should indicate the user at any point in time.
* The design (Graphical) must be comprehendible and not clumsy to the user; easy to use, and easy to understand.
* The system should be able to generate reports and print out information on users demand.
* The system must have access levels based on user roles such as Manager-Administrator-Accountant-Other staff.
* The system should generate primary keys in most cases unless specified by user, and the unique keys should be visible.
* The system must carry out only actions specified by the user (browse, modify, delete, add).

**3.1.1 Use Case Scenarios**

The Project work will ensure booking of hotel rooms, staff management, and resource management. A “Use Case” scenario is the room search for room booking. Users may face difficulties searching between available and booked rooms, but the automated system would search more efficiently with the proficient search algorithm. All details of the rooms are stored in the database servers and can be retrieved or modified with very little stress. Another “Use Case” is the accounts receivable and payable field of the F&A module. The accounts receivable simply captures all funds coming-in with their sources and dates while the accounts payable displays the money going-out of the organization with their destination. The business flow is quite simple; however, to accomplish all these tasks is burdensome for both the customer side and the hotel side without an efficient and integrated hotel management system.

**3.1.2 Use Case Diagrams**

This Use Case Diagram is a graphic depiction of the interactions among the elements of Hotel Management System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Hotel Management System. The main actors of Hotel Management System in this Use Case Diagram are: Super Admin, chef, customer, who perform the different type of use cases such as Manage Hotel, Manage Rooms, Manage Services, Manage Payments, Manage Booking, Manage Customers, Manage Users and Full Hotel Management System Operations. Major elements of the UML use case diagram of Hotel Management System are shown on the picture below.



**3.2 Feasibility Study**

**3.2.1 Technical Feasibility (based on hardware & software availability)**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows 7 platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible .The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

**3.2.2 Operational Feasibility (based on ease of understanding and use)**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, disposability, sustainability, affordability and others. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

**3.2.3 Economical Feasibility (based on cost estimation)**

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

**3.3 Requirement Analysis of the Proposed System**

**3.3.1 Context Level DFD**

Room

Category

Management

Employee

Management

Booking

Management

System

Admin

Management

Restaurant

Service

Management

**3.3.2 Level-1 DFDs**

Create account

Customer

Admin

Sign in

Employee

**3.3.3 Level-2 DFDs**

Admin Detail

And Encrypt password

Admin

SQL: Account

Employee

Sign

In

Customer

SQL: Room

SQL: Room

Availability

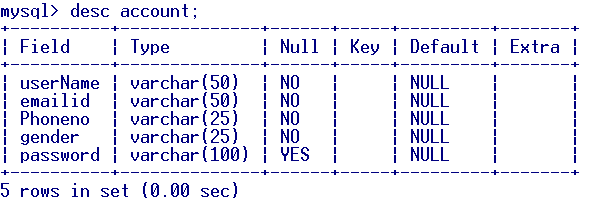
SQL: Check Account

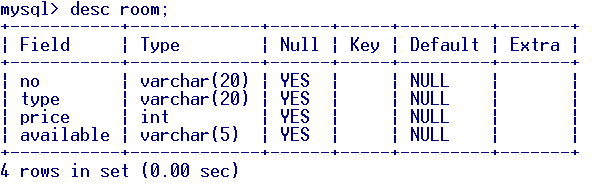
SQL: Customer

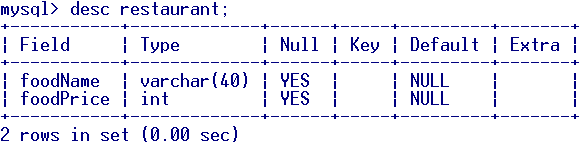
SQL: Restaurant

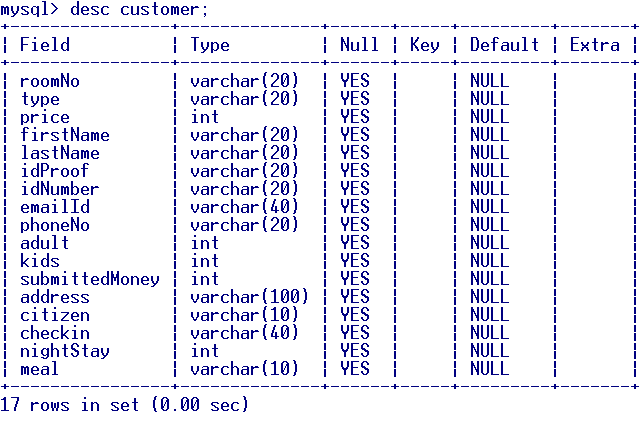
SQL: Employee

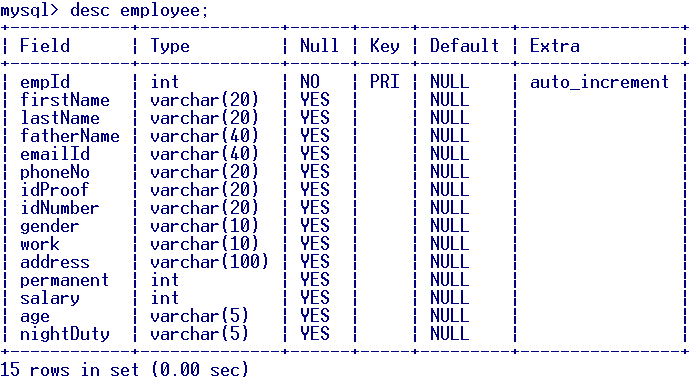
**3.3.4 Data Dictionary**

****

****

****

****

****

3.3.5 E-R Diagrams

M

1

Food in

M

1

1

M

Order

Booked

Work In

1

N

Restaurant

Customer

1

Are in

Room

M

Account

Employee

3.4 Software Requirement Specifications (SRS)

1. Operating System : Windows 7 and upwards operating system

2. JDK : Java SE Development Kit 11

3. Database : MySQL 8.0.17

3.5 Hardware and Other Requirements

1. Processor : Dual core Intel
2. Clock Speed : 2 GHz
3. Storage : 750 MB free disk space
4. RAM : 1GB or more
5. Keyboard : 101 Keys
6. Mouse : Optical Mouse

**4. System Design**

**4.1 Design Approach – Top Down/ Bottom Up (Whichever applicable)**

In the top-down approach, a complex algorithm is broken down into smaller fragments, better known as ‘modules.’ These modules are then further broken down into smaller fragments until they can no longer be fragmented. However, during the modularization process, you must always maintain the integrity and originality of the algorithm. By breaking a bigger problem into smaller fragments, the top-down approach minimizes the complications usually incurred while designing algorithms. Furthermore, in this approach, each function in a code is unique and works independently of other functions. The top-down approach is heavily used in the java programming language.

Contrary to the top-down approach, the bottom-up approach focuses on designing an algorithm by beginning at the very basic level and building up as it goes. In this approach, the modules are designed individually and are then integrated together to form a complete algorithmic design.

So, in this method, each and every module is built and tested at an individual level prior to integrating them to build a concrete solution. The unit testing is performed by leveraging specific low-level functions

**4.2 Design Methodology – Function Oriented/ Object Oriented**

**4.3 Module and Sub-Module Description**

**1. Check-in and Check-out module:**

First greet the customer and we take the details of the customer so that we can verify the identity of that customer. Then, we will talk about the room i.e. either you want room for smoking area or not and we will also talk for spend the days. Then we provide the identity card number to you. When you will arrive we will ask your name and the date at which you have registered. We will give you the registration form so we can check you in and show the identity card and then how will you pay the money either by credit card, debit card, cash, etc. then we will provide you the room key.

When there is a time to check out then first we will see the room condition and if it is perfect then you can go and get the receipt from the counter. If you have any complain about our hotel then you can say without hesitation we will do our best at next time.

**2. Restaurant:**

This is the major facility which we provide to our customers. We have a proper kitchen with adequate staff and we also maintain a menu according to the demands of the customers. We also take further steps for proper hygiene in the kitchen. We also provide telephone facility so that the customers will order the food form their room.

**3. Staff management:**

We have created a proper staff as per need in the hotel. And we are selective in hiring our staff. We give them work according to their skills and we communicate with them properly so that they work properly. We also monitor who and when has changed the order status. Our staff uses their time efficiently as modifications of new orders go directly to the right department and everybody knows what to do next. We are in touch with them for their help.

**4. Admin Login:**

This aims to create hotel booking online. Customer can check the available rooms and the customer data. The user register and log into system. The administrator knows the details of room.

Software is being designed in such a way as they provide all the information to the customers including hotel rooms, specifications, and services provided. By this, it is the advantage for the customer to remove the existing flaws in manual system booking hotel rooms.

**4.4 User Interface Design - Snapshots**

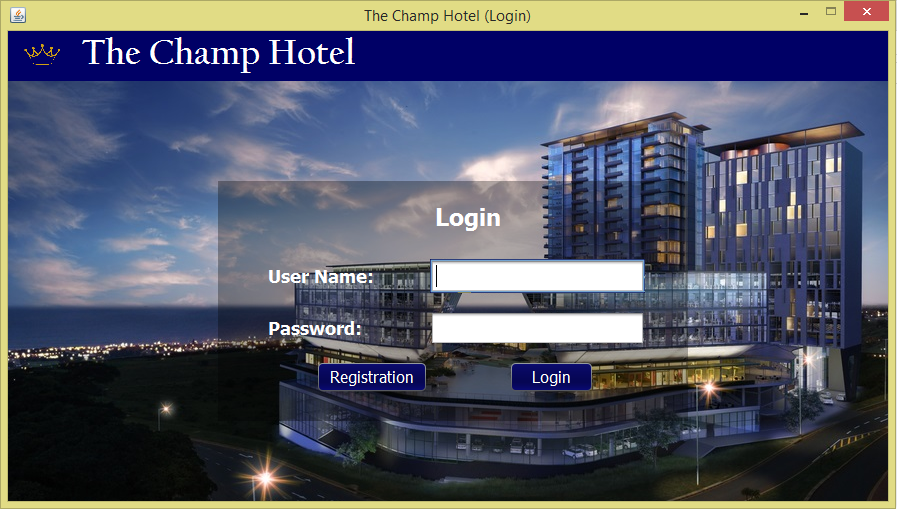
****

Fig 4.4.1

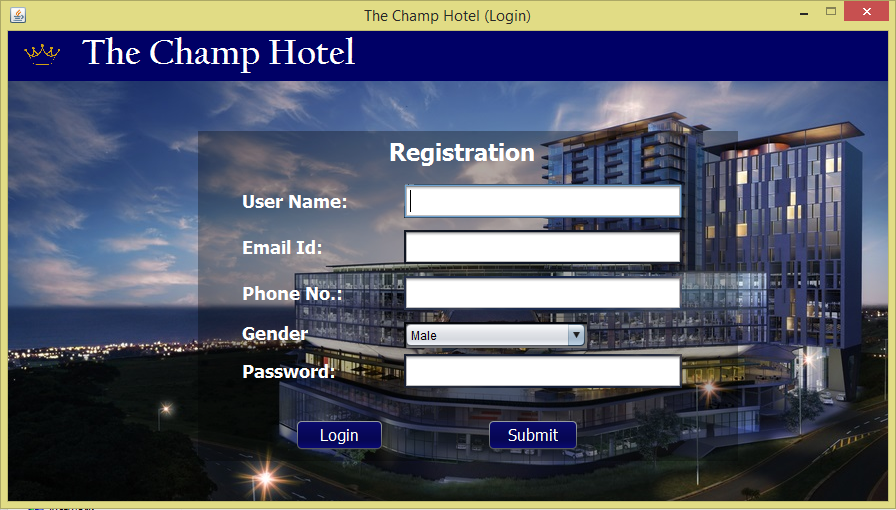
****

Fig 4.4.2

****

Fig 4.4.3



Fig 4.4.4

****

Fig 4.4.5

**5. Coding**

**5.1 Hard Code**

**File name –** MainGUI.java

Admin login and registration module

import java.security.NoSuchAlgorithmException;

import java.sql.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

public class MainGUI extends javax.swing.JFrame {

public MainGUI() {

initComponents();

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

login\_page = new javax.swing.JPanel();

login\_popupl = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

jLabel6 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jPasswordField1 = new javax.swing.JPasswordField();

jLabel2 = new javax.swing.JLabel();

registration\_page = new javax.swing.JPanel();

registration\_pop = new javax.swing.JPanel();

jButton3 = new javax.swing.JButton();

jLabel8 = new javax.swing.JLabel();

jTextField2 = new javax.swing.JTextField();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

jButton4 = new javax.swing.JButton();

jPasswordField2 = new javax.swing.JPasswordField();

jLabel11 = new javax.swing.JLabel();

jTextField3 = new javax.swing.JTextField();

jTextField4 = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();

jLabel13 = new javax.swing.JLabel();

jComboBox1 = new javax.swing.JComboBox<>();

jLabel4 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("The Champ Hotel (Login)");

setResizable(false);

getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel1.setBackground(new java.awt.Color(0, 0, 102));

jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Champ/logo\_60.png"))); // NOI18N

jLabel3.setBackground(new java.awt.Color(255, 255, 255));

jLabel3.setFont(new java.awt.Font("Californian FB", 1, 36)); // NOI18N

jLabel3.setForeground(new java.awt.Color(255, 255, 255));

jLabel3.setText("The Champ Hotel");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jLabel3)

.addGap(0, 529, Short.MAX\_VALUE))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel3)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 49, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE))

);

getContentPane().add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 880, 50));

jPanel2.setLayout(new java.awt.CardLayout());

login\_page.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

login\_popupl.setBackground(new java.awt.Color(0, 0, 0,80));

jButton1.setBackground(new java.awt.Color(0, 0, 51));

jButton1.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N

jButton1.setForeground(new java.awt.Color(255, 255, 255));

jButton1.setText("Registration");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel6.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel6.setForeground(new java.awt.Color(255, 255, 255));

jLabel6.setText("User Name:");

jTextField1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel5.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jLabel5.setForeground(new java.awt.Color(255, 255, 255));

jLabel5.setText("Login");

jLabel7.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel7.setForeground(new java.awt.Color(255, 255, 255));

jLabel7.setText("Password:");

jButton2.setBackground(new java.awt.Color(0, 0, 51));

jButton2.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Login");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jPasswordField1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

javax.swing.GroupLayout login\_popuplLayout = new javax.swing.GroupLayout(login\_popupl);

login\_popupl.setLayout(login\_popuplLayout);

login\_popuplLayout.setHorizontalGroup(

login\_popuplLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(login\_popuplLayout.createSequentialGroup()

.addGap(98, 98, 98)

.addComponent(jButton1)

.addGap(81, 81, 81)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 85, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(login\_popuplLayout.createSequentialGroup()

.addGap(217, 217, 217)

.addComponent(jLabel5)

.addGap(0, 0, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, login\_popuplLayout.createSequentialGroup()

.addGap(50, 50, 50)

.addGroup(login\_popuplLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel6)

.addComponent(jLabel7))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 56, Short.MAX\_VALUE)

.addGroup(login\_popuplLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jPasswordField1)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 215, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(43, 43, 43))

);

login\_popuplLayout.setVerticalGroup(

login\_popuplLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(login\_popuplLayout.createSequentialGroup()

.addGap(21, 21, 21)

.addComponent(jLabel5)

.addGroup(login\_popuplLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(login\_popuplLayout.createSequentialGroup()

.addGap(28, 28, 28)

.addGroup(login\_popuplLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel6)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(login\_popuplLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel7))

.addGap(41, 88, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, login\_popuplLayout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(login\_popuplLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1)

.addComponent(jButton2))

.addGap(28, 28, 28))))

);

login\_page.add(login\_popupl, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 100, 470, 240));

jLabel2.setIcon(new javax.swing.ImageIcon("D:\\project\\project with java\\Hotel Management System\\src\\Champ\\hotel\_pic.jpg")); // NOI18N

login\_page.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, -1, -1));

jPanel2.add(login\_page, "card3");

registration\_page.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

registration\_pop.setBackground(new java.awt.Color(0, 0, 0,80));

jButton3.setBackground(new java.awt.Color(0, 0, 51));

jButton3.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N

jButton3.setForeground(new java.awt.Color(255, 255, 255));

jButton3.setText("Login");

jButton3.setToolTipText("");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jLabel8.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel8.setForeground(new java.awt.Color(255, 255, 255));

jLabel8.setText("User Name:");

jTextField2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel9.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jLabel9.setForeground(new java.awt.Color(255, 255, 255));

jLabel9.setText("Registration");

jLabel10.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel10.setForeground(new java.awt.Color(255, 255, 255));

jLabel10.setText("Password:");

jButton4.setBackground(new java.awt.Color(0, 0, 51));

jButton4.setFont(new java.awt.Font("Tahoma", 0, 16)); // NOI18N

jButton4.setForeground(new java.awt.Color(255, 255, 255));

jButton4.setText("Submit");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

jPasswordField2.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel11.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel11.setForeground(new java.awt.Color(255, 255, 255));

jLabel11.setText("Email Id:");

jTextField3.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jTextField4.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

jLabel12.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel12.setForeground(new java.awt.Color(255, 255, 255));

jLabel12.setText("Phone No.:");

jLabel13.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N

jLabel13.setForeground(new java.awt.Color(255, 255, 255));

jLabel13.setText("Gender");

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Male", "Female", "Other" }));

javax.swing.GroupLayout registration\_popLayout = new javax.swing.GroupLayout(registration\_pop);

registration\_pop.setLayout(registration\_popLayout);

registration\_popLayout.setHorizontalGroup(

registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(registration\_popLayout.createSequentialGroup()

.addGap(44, 44, 44)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel8)

.addComponent(jLabel11)

.addComponent(jLabel12)

.addComponent(jLabel13)

.addComponent(jLabel10))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jTextField4, javax.swing.GroupLayout.DEFAULT\_SIZE, 278, Short.MAX\_VALUE)

.addComponent(jTextField3)

.addComponent(jPasswordField2)

.addComponent(jTextField2))

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, 183, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(registration\_popLayout.createSequentialGroup()

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(registration\_popLayout.createSequentialGroup()

.addGap(97, 97, 97)

.addComponent(jButton3, javax.swing.GroupLayout.PREFERRED\_SIZE, 89, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(103, 103, 103)

.addComponent(jButton4, javax.swing.GroupLayout.PREFERRED\_SIZE, 92, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(registration\_popLayout.createSequentialGroup()

.addGap(191, 191, 191)

.addComponent(jLabel9)))

.addContainerGap(159, Short.MAX\_VALUE))

);

registration\_popLayout.setVerticalGroup(

registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, registration\_popLayout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel9)

.addGap(18, 18, 18)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel8))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel11))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel12))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel13)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 20, Short.MAX\_VALUE)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jPasswordField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel10))

.addGap(31, 31, 31)

.addGroup(registration\_popLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton3)

.addComponent(jButton4))

.addGap(23, 23, 23))

);

registration\_page.add(registration\_pop, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 50, 540, 330));

jLabel4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Champ/Hotel\_pic.jpg"))); // NOI18N

registration\_page.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 880, -1));

jPanel2.add(registration\_page, "card3");

getContentPane().add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 50, 880, 420));

pack();

}// </editor-fold>//GEN-END:initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

login\_page.setVisible(false);

registration\_page.setVisible(true);

}//GEN-LAST:event\_jButton1ActionPerformed

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton3ActionPerformed

registration\_page.setVisible(false);

login\_page.setVisible(true);

}//GEN-LAST:event\_jButton3ActionPerformed

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton4ActionPerformed

String registration\_password= new String(jPasswordField2.getPassword());

if (!"".equals(jTextField2.getText().trim()) || !"".equals(jTextField3.getText().trim()) || !"".equals(jTextField4.getText().trim()) || !"".equals(registration\_password.trim())) {

try{

String securePassword = EncryptedSecurity.hashPassword(registration\_password);

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/hotel\_management\_system","root","Champ123@");

String sql1;

sql1 = "SELECT \* from account WHERE username = '"+jTextField2.getText()+"'";

Statement ptm1 = con.createStatement();

ResultSet rs = ptm1.executeQuery(sql1);

if (rs.next()) {

javax.swing.JOptionPane.showMessageDialog(null,"Already exist username");

}

else {

String sql = "INSERT INTO account VALUES (?,?,?,?,?)";

PreparedStatement ptm = con.prepareStatement(sql);

ptm.setString(1, jTextField2.getText().trim());

ptm.setString(2, jTextField3.getText().trim());

ptm.setString(3, jTextField4.getText().trim());

ptm.setString(4, jComboBox1.getSelectedItem().toString());

ptm.setString(5, securePassword);

ptm.executeUpdate();

javax.swing.JOptionPane.showMessageDialog(null,"Successfully registered");

con.close();

jTextField2.setText("");

jTextField3.setText("");

jTextField4.setText("");

jPasswordField2.setText("");

}

}

catch(ClassNotFoundException | SQLException | NoSuchAlgorithmException ex) {

Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

}//GEN-LAST:event\_jButton4ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

String login\_password= new String(jPasswordField1.getPassword());

if (!"".equals(jTextField1.getText().trim())) {

try{

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/hotel\_management\_system","root","Champ123@");

String sql1;

sql1 = "SELECT \* from account WHERE username = '"+jTextField1.getText().trim()+"' and password ='"+EncryptedSecurity.hashPassword(login\_password)+"'";

Statement ptm1 = con.createStatement();

ResultSet rs = ptm1.executeQuery(sql1);

if (rs.next()) {

MainGUI2.jLabel75.setText(jTextField1.getText().trim());

jTextField1.setText("");

jPasswordField1.setText("");

MainGUI2.main2.setVisible(true);

MainGUI.main.setVisible(false);

UpdateTable.update\_table1();

UpdateTable.update\_table2();

UpdateTable.update\_table3();

UpdateTable.update\_table4();

UpdateTable.update\_table5();

UpdateTable.update\_dashBoard();

}

else {

javax.swing.JOptionPane.showMessageDialog(null,"Username or password not matched");

}

}

catch(ClassNotFoundException | SQLException | NoSuchAlgorithmException ex) {

Logger.getLogger(MainGUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

}//GEN-LAST:event\_jButton2ActionPerformed

public static javax.swing.JFrame main;

public static void main(String args[]) {

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainGUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

main= new MainGUI();

main.setVisible(true);

MainGUI2.main2 = new MainGUI2();

MainGUI2.main2.setVisible(false);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JComboBox<String> jComboBox1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel13;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPasswordField jPasswordField1;

private javax.swing.JPasswordField jPasswordField2;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

private javax.swing.JPanel login\_page;

private javax.swing.JPanel login\_popupl;

private javax.swing.JPanel registration\_page;

private javax.swing.JPanel registration\_pop;

// End of variables declaration//GEN-END:variables

}

**File Name -** EncryptedSecurity.java

Encrypt password module

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

class EncryptedSecurity{

public static String hashPassword(String password) throws NoSuchAlgorithmException{

MessageDigest md = MessageDigest.getInstance("SHA");

md.update(password.getBytes());

byte[] b = md.digest();

StringBuilder sb = new StringBuilder();

for (byte b1 : b){

sb.append(Integer.toHexString(b1 & 0xff));

}

return sb.toString();

}

}

**6. Testing**

**6.1 Methodology of Testing used in the project**

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables us to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by java. The various controls are tested to ensure that each performs its actions as required.

**6.2 Test Cases**

A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

**Black-Box Testing**

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

**White-Box Testing**

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

**6.3 Steps of Testing of Module**

* For Module Testing, designing a Test Case is an important segment. While designing test cases for a module test, a tester has to take two things into consideration.
  + Specification for the module
  + The module's source code
* Analyze the module's logic by using one or more of the white box methods, and then supplement these test cases by applying black box methods to the modules specification.
* Once the test case is designed, the next step is to combine the module for testing. For this, the method used is either an **Incremental or non-Incremental method**.
* Non-incremental method- all modules are tested independently. First, it combines all modules and then test the whole program.
* Incremental method- each module is tested first and then gradually incremented to the tested collection. It does a step wise retesting
* Incremental Testing, there are two approaches – Top down and Bottom Up testing
* To execute the module with the selected data, it requires a **driver**for supplying the test data, monitoring the execution and capturing the results

**7. Implementation/Maintenance**

**7.1 Installation Procedure**

### Step 1 - Download JDK

Open the browser and search for Download JDK 11 or click the [link](https://www.oracle.com/java/technologies/javase-jdk11-downloads.html) to download from Oracle website.

Accept the License Agreement and click on the link to download the installer as highlighted in Fig 7.1.1. It will start downloading JDK 11 installer for Windows.

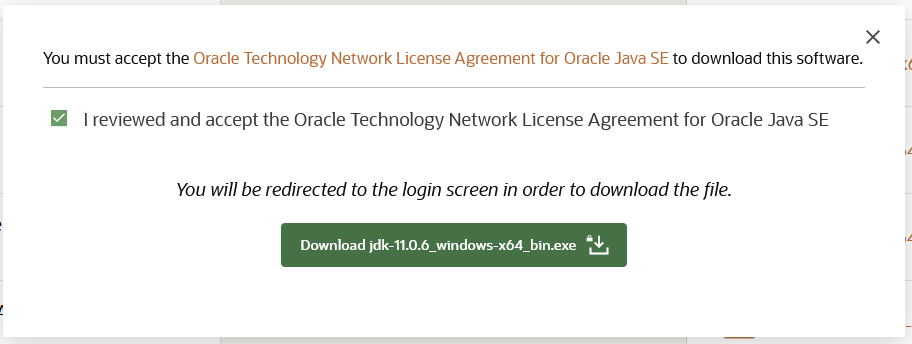


Fig 7.1.1

### Step 2 - Install JDK

Now execute the downloaded JDK installer by double-clicking it. It might ask system permission before starting the installation. Click on yes to allow the installer to execute itself. It shows the installer welcome screen as displayed in Fig 7.1.2

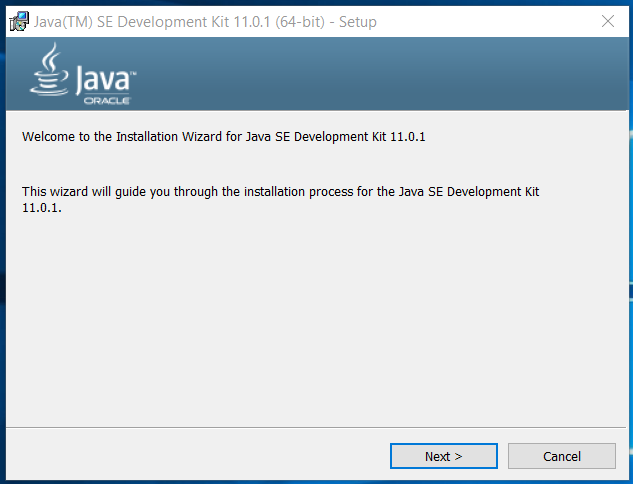


Fig 7.1.2

Click on Next to initiate the installation process. The next screen shows options to select optional features to be installed together. Leave the default options without making any change. We can also change the installation location on this screen if required as displayed in Fig 7.1.3.

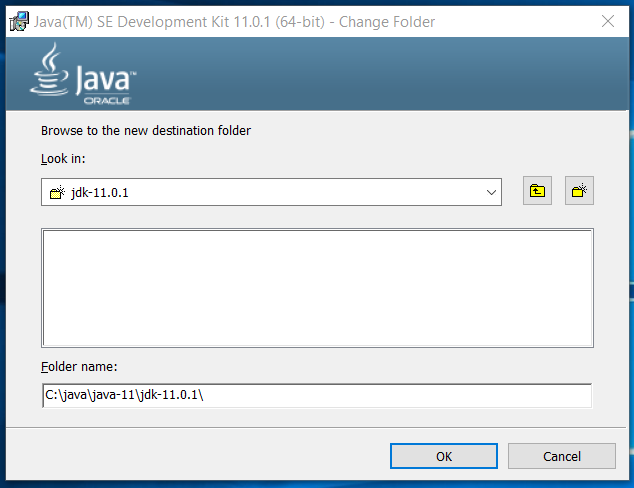


Fig 7.1.3

Now click on Next Button to start the installation. It will show the progress as displayed in Fig 7.1.4.

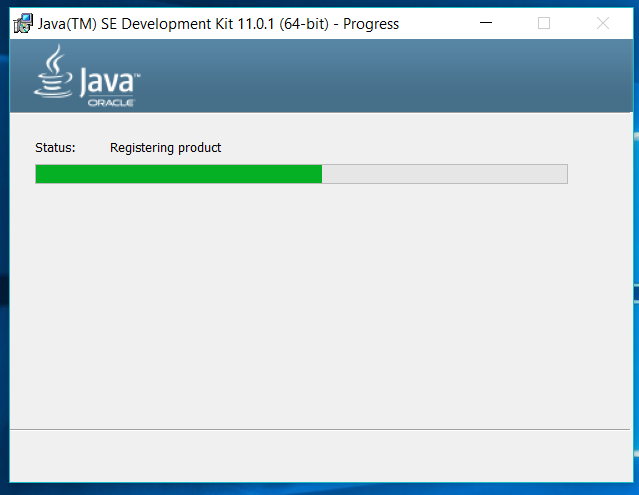


Fig 7.1.4

It shows the success screen after completing the installation as mentioned in Fig 7.1.5.

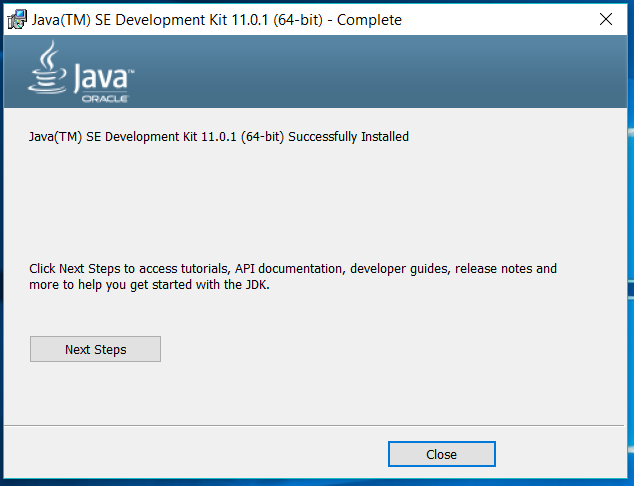


Fig 7.1.5

## Step 3 - Download MySQL Installer for Windows

The MySQL Installer for Windows helps you control the installation process by providing a user-friendly interface. It also guides you through the steps needed to configure MySQL.

Access your [Windows server](https://phoenixnap.com/servers/dedicated) and download the MySQL Installer. A free Community edition MySQL Installer is available from the official page: <https://dev.mysql.com/downloads/installer/>

You are given the option to download either the Web Community version or the Full MySQL package.

Select and download your preferred version. In this example, we selected the **Full MySQL Package** (B).

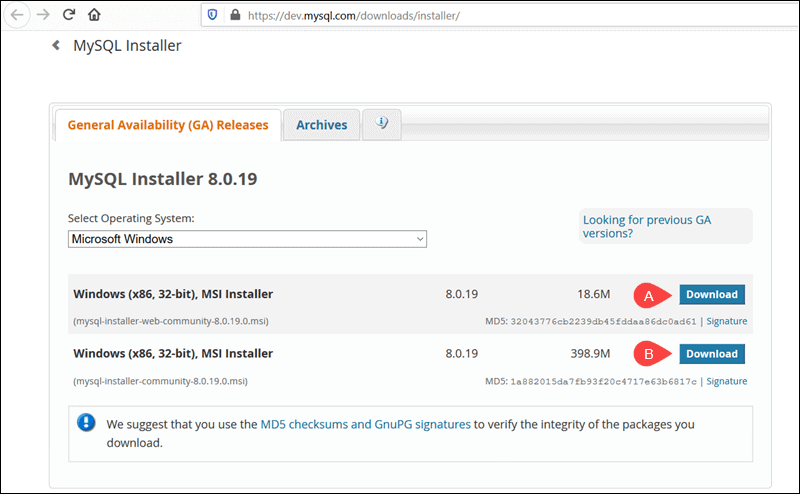


Fig 7.1.6

After selecting a version, you are provided with the option of signing up for a MySQL Community account. If you are not interested, select the **No thanks, just start my download** option at the bottom of the page.

By selecting this option, the download process starts immediately. Once the download is complete, you can execute the MySQL Installer file from the download folder.

It can take a few moments while Windows configures the MySQL Installer and prepares the installation and configuration process.

## Step 4 - Set Up MySQL Installer for Windows

After accepting the Oracle license agreement terms, the first screen you encounter allows you to define which MySQL products are going to be installed. You can choose between several predefined options or create your custom setup type.

After accepting the Oracle license agreement terms, the first screen you encounter allows you to define which MySQL products are going to be installed. You can choose between several predefined options or create your custom setup type.

* **Server Only** is used to install an instance of the MySQL Server and forgo other MySQL products.

In the example below, we select the **Server Only** option and click **Next**.

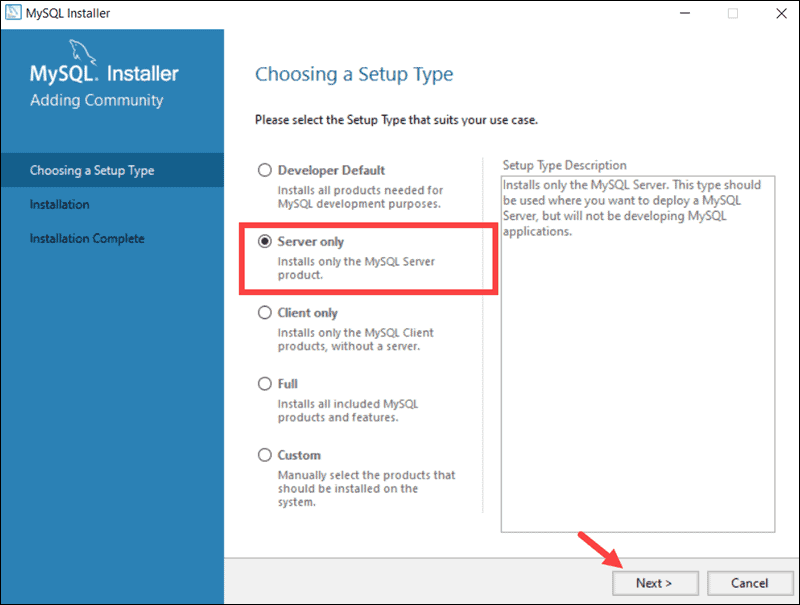


Fig 7.1.7

At this point, the system tries to resolve possible inconsistencies. It might inform you that additional packages need to be installed for the process to continue (e.g., Microsoft Visual C++ 2019 Redistributable Package). You can also run into Path installation inconsistencies if you have previous MySQL installations on your Windows Server.

Luckily the **MySQL Installer auto-resolves issues** and installs the latest binary compatible version of missing software. You are now ready to start the installation process in earnest. Click **Execute** to begin the installation process.

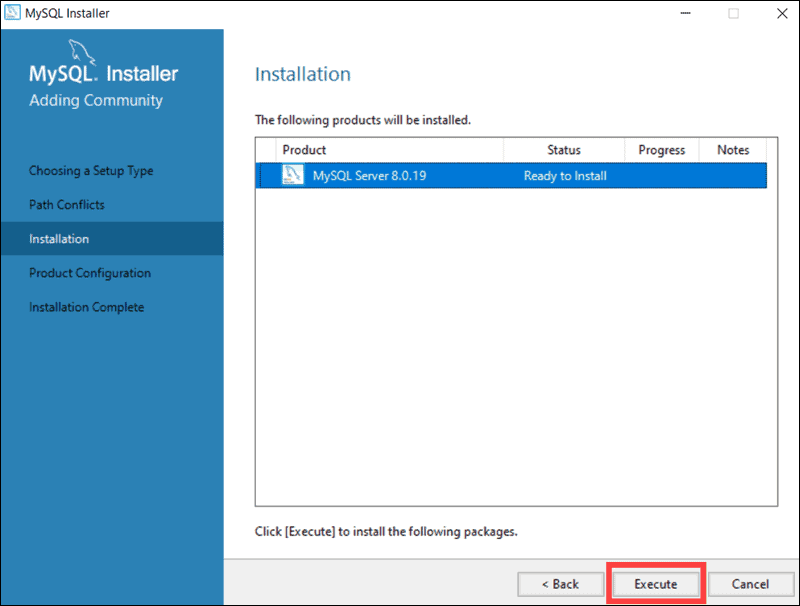


Fig 7.1.8

Once the status of the installation status is labelled as Complete, you are ready to configure the MySQL database.

The MySQL Server 8.0.19 is now ready to be configured. Initiate the process by clicking **Next**.

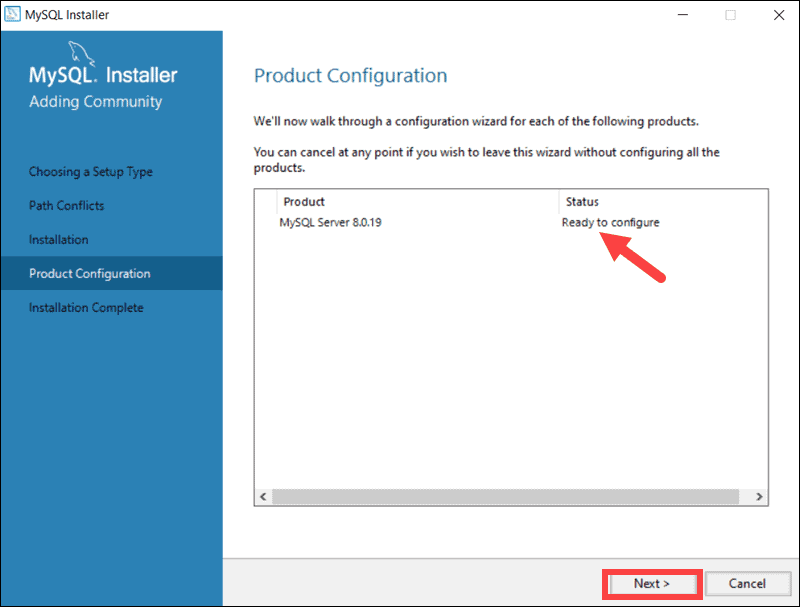


Fig 7.1.9

### High Availability

The first configuration option affects database availability. It allows you to decide if you want to set up a Standalone MySQL Server or an InnoDB server cluster to improve availability. In this instance, we selected the classic, single server option.

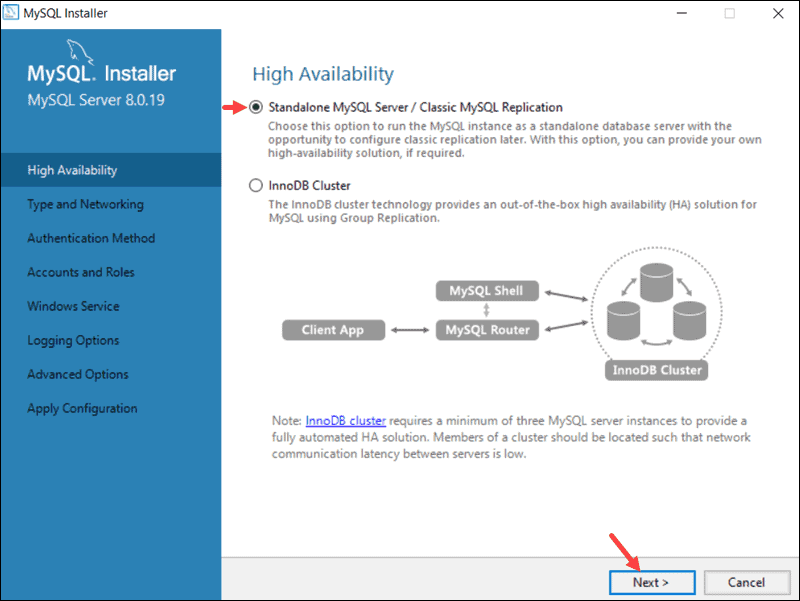


Fig 7.1.10

### Type and Networking

The **Type and Networking** section is used to define several essential features.

The Config Type option lets you choose between three server configuration types. Development Computer, Server Computer, and Dedicated Computer define whether the server is dedicated solely to running your MySQL database or is going to share the underlying system with other applications.

In this example, we decided to create a dedicated MySQL server.

The **Type and Networking** tab can also define the port the MySQL server is listening on. The default setting is port number 3306 and can be changed to suit your needs.

By checking the Show Advanced and Logging Option box, you can set additional logging options at a later stage.

Click **Next**once you’ve selected the options you feel meet your requirements.

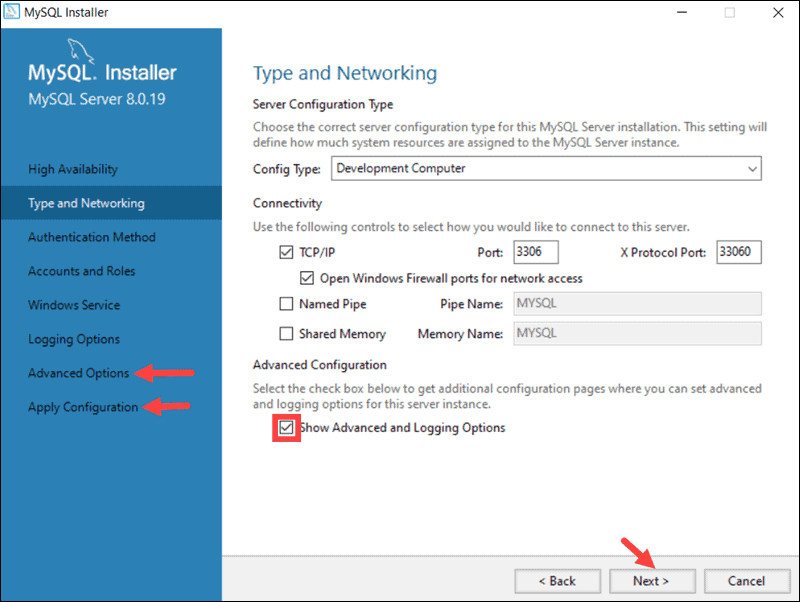


Fig 7.1.11

### Authentication Method

It is possible to choose between two authentication methods, the recommended Strong Password Encryption, and the Legacy Authentication Method. Select the recommended **Use Strong Password Authentication** option.

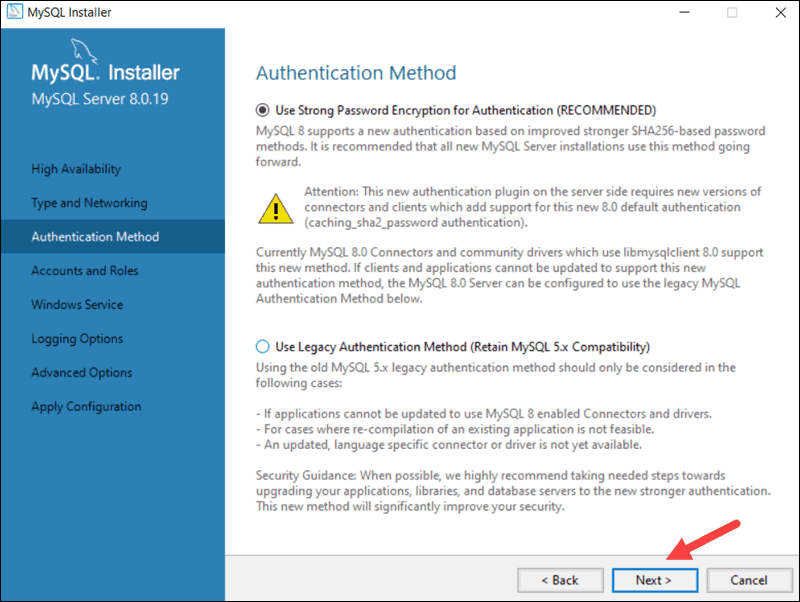


Fig 7.1.12

### Accounts and Roles

You are now prompted to enter a password for your MySQL root user. You can also create additional roles for various users and purposes.

This is only an initial setup, and credentials can be edited once the installation is complete.

Password = “**Champ123@**”

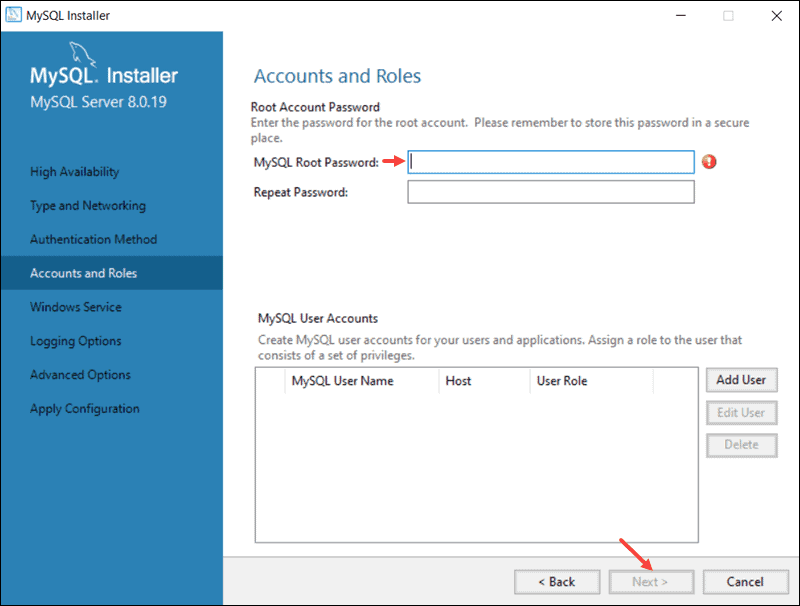


Fig 7.1.13

### Windows Service

By defining MySQL as a Windows Service, it can now start automatically whenever the Windows system boots.

If you decide to start MySQL as an executable application, you would need to configure it manually.

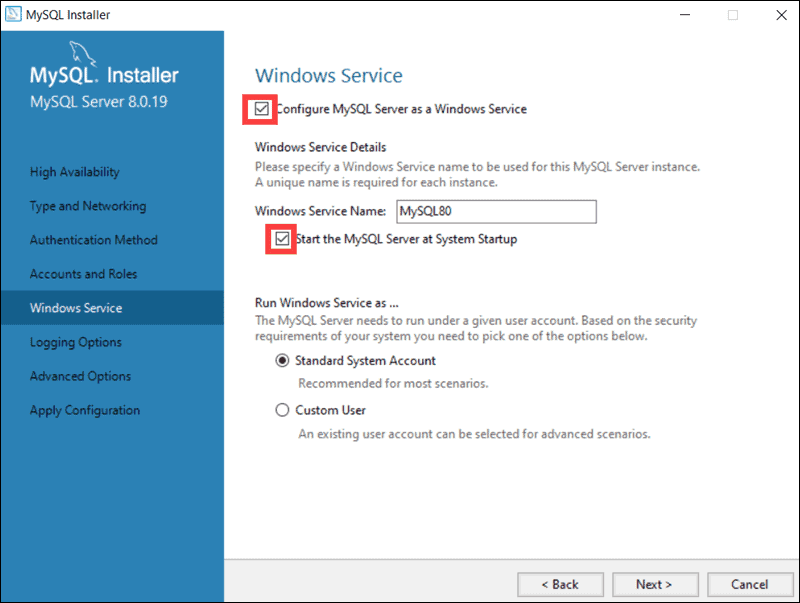


Fig 7.1.14

### Apply Configuration

You have successfully configured the MySQL server and need to confirm for the MySQL Installer to apply the configuration.

An overview of the configurations steps appears on the screen. Click **Execute** to apply the configuration.

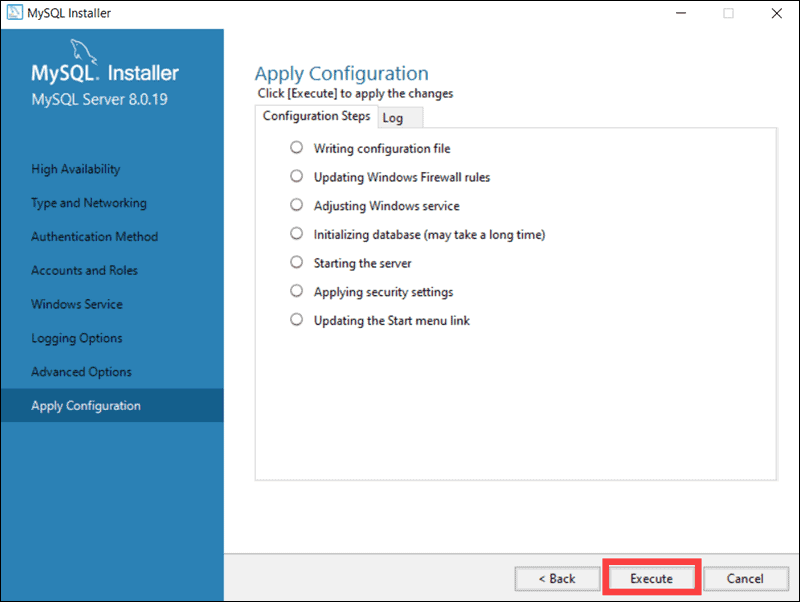


Fig 7.1.15

The system informs once the configuration process is completed. Select **Next** to continue the installation process.

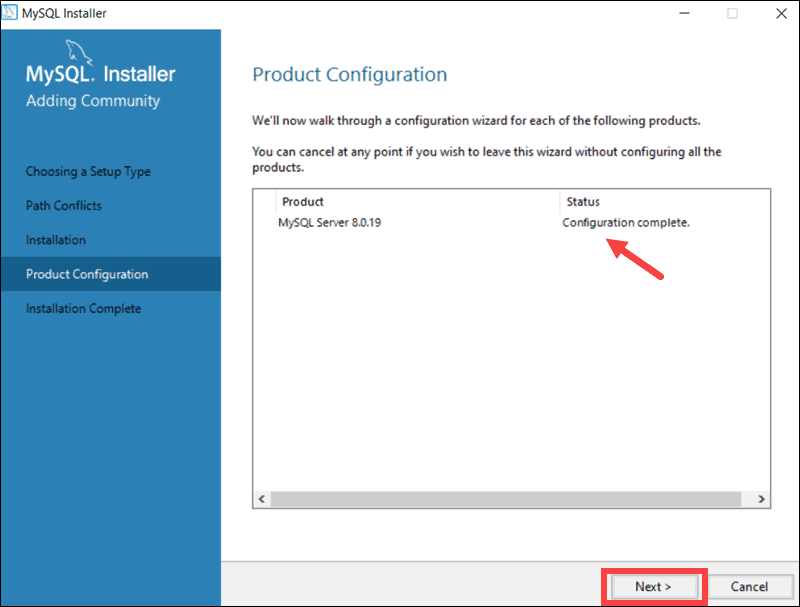


Fig 7.1.16

Click **Finish** to complete the MySQL server installation on Windows.

**Step 5 – Add to path**

We might need to set the environment variable in case the installed JDK and MySQL is not detected by the system. You can follow the below-mentioned steps to set the environment variable.

Right Click -> My Computer(This PC) -> Properties -> Advanced System Settings

The above steps will open the Windows settings panel as shown in Fig 7.1.17.

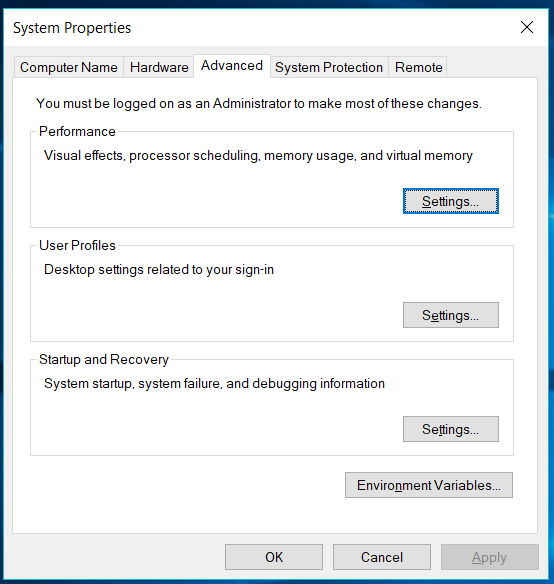


Fig 7.1.17

Now click on Environment Variables, select Path under System Variables section and click on Edit. We need to add the path of installed JDK to system Path.

Click on New Button and add the path to installed JDK and MySQL in which is **C:\java\java-11\jdk-11.0.1\bin; C:\Program Files\MySQL\MySQL Server 8.0\bin;** in our case. Press OK Button 3 times to close all the windows. This sets the JDK 11 on system environment variables to access the same from the console.

**Step 6 – Dump Database**

Now, Open start>cmd

And type **mysql –u root -p**

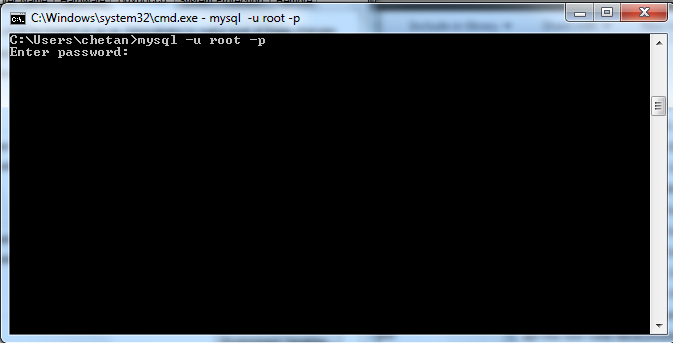


Fig 7.1.18

Write password of mysql server “**Champ123@**”

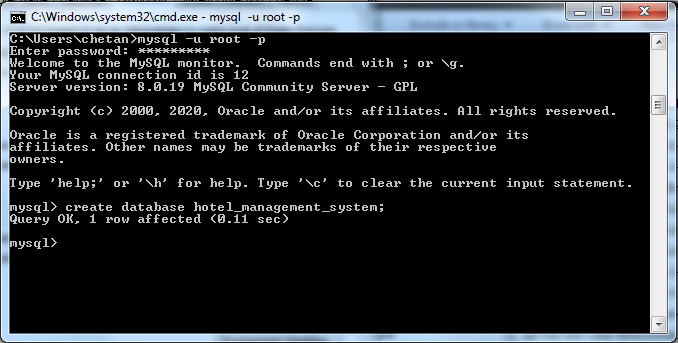


Fig 7.1.19

Now, Write “**create database hotel\_management\_system;**”

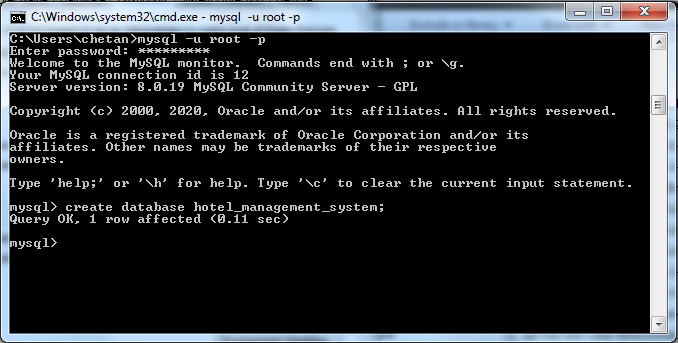


Fig 7.1.20

Open project folder and copy the path of **Database.sql**

And write “**mysql –u –p hotel\_management\_system <** copy\_path**\database.sql**”

And give password of MySQL (Champ123@)

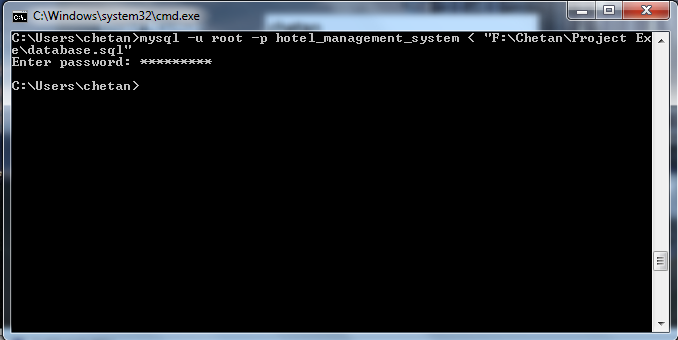
****

Fig 7.1.21

Now you can open project.exe file and work on it…

**7.2 Do and Don’ts**

* Install JDK 11 and Mysql 8.0.19
* Dump database.sql in MySQL
* Software don’t run without JDK and MySQL
* Software don’t work without dump database

**8. Conclusion**

The conclusion of this project is A Hotel management system is a computerized management system. This system keeps the records of hardware assets besides software of this organization. The proposed system will keep a track of Workers, Residents, Accounts and generation of report regarding the present status. This project has GUI based software that will help in storing, updating and retrieving the information through various user-friendly menu-driven modules. The project “Hotel Management System” is aimed to develop to maintain the day-to-day state of admission/Vacation of Residents, List of Workers etc. Main objective of this project is to provide solution for hotel to manage most there work using computerized process. This software application will help admin to handle customer information, room allocation details, payment details, billing information.etc. Detailed explanation about modules and design are provided in project documentation. The existing system is a manually maintained system. All the Hotel records are to be maintained for the details of each customer, Fee details, Room Allocation, Attendance etc. All these details are entered and retrieved manually, because of this there are many disadvantages like Time Consuming, updating process, inaccuracy of data. For avoiding this we introduced or proposed a new system in proposed system the computerized version of the existing system. Provides easy and quick access over the data.

**9. Limitations and Future Works**

As earlier mentioned the project study covers Reception, accommodation, finance and account, catering, food ordering, administrative, security, and general services transactions in the Hotel management.

However the project has limitations based on these facts

* The “finance and account” aspect of the HMIS will not capture the budget function; it only captures the accounts receivables and accounts payable.
* Another limitation of the system is that customer’s signature will not be captured. This process might make procedures cumbersome, which is what the study hopes to eliminate; however it captures full details of the customer.
* The system does not have an online payment option on the online room reservation menu.
* The system is not designed to run on-line.
* Due to time constraints certain fields were not included; the software was therefore reduced to covering critical aspect of hotel management.

These limitations were encountered in the course of the study, and appropriate techniques have been applied to ensure the system functions properly thereby eliminates the “stale mate”

**10. Bibliography and References**

[Java Documentation - Get Started (oracle.com)](https://docs.oracle.com/en/java/)

[Java Tutorial | Learn Java - javatpoint](https://www.javatpoint.com/java-tutorial)

[Java Tutorial (w3schools.com)](https://www.w3schools.com/java/)

[SQL Tutorial - Tutorialspoint](https://www.tutorialspoint.com/sql/)

[Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/)

[YouTube](https://www.youtube.com/)

Herbert Schildt “Java: The Complete Reference” Tata McGraw-Hill Publishing Company Limited, Seventh Edition