**Index**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.no.** | **Chapter name** | | | **Page no.** |
| 1 | Chapter 1: Introduction | | | 2 |
| 1.1 | Projects Objectives | | | 3 |
| 1.2 | Existing system and need of project | | | 3 |
| 1.3 | Scope of work | | | 4 |
| 1.4 | Operating Environment-Hardware and Software | | | 5 |
| 1.5 | Technology Used | | | 5 |
| 1.6 | Module Specification | | | 5 |
| 2 | Chapter 2: Analysis & Design | | | 7 |
| 2.1 | Data Flow Diagram | | | 7 |
| 2.2 | Entity Relationship Diagram | | | 9 |
| 2.3 | Use Case Diagram | | | 10 |
| 2.4 | Activity Diagram | | | 11 |
| 2.5 | Sequence Diagram | | | 12 |
| 2.6 | Class Diagram | | | 14 |
| 2.7 | User Interface Screens | | | 15 |
| 2.8 | Table Structure | | | 21 |
| 3 | Drawbacks and Limitations | | | 23 |
| 4 | Proposed Enhancement | | | 25 |
| 5 | Conclusion | | | 26 |
| 6 | | Bibliography | 27 | | |

**Chapter 1: Introduction**

* In this fast growing and digitalizing world of technology, it becomes extremely difficult to store data with a proper manner in a short time period but a To Do Application or any online web portal is of great utility as it makes it very easy to keep and arrangement of our daily data. And tracking system of project and their related data. In there a track of the daily work performed by a worker or employee. In specific company or any organization.
* Its manage the timetable of workers as per their entry and exit time and also know updates to manager which specific work of any project for a client so, this performed by a worker or employee this information related to project is also knows to manager and HR of this company.
* This type of systems can be used in management at any workplace eg. Any company or any organization It helps to company to manage the work done or which task perform by employee for specific company. And manage the information about which type work as well as project completed by the employees and manager of this company or organization.
* Timesheet tracking system, web timesheet, time and expense software, time clock, time and attendance, the list goes on.  Sometimes the difference is just one of convention for a particular country, but other times it implies a distinct and important difference in the functionality or industry focus of the software application.
* Timesheet tracking system is used to record and analyse the time spent by individuals on tasks. Typically, tasks are linked with projects and clients, and staff assigned to work on them. Costs and/or revenue can usually be defined, and may vary depending on the person, task, project or client.

1.1Project Objectives:

* The main objective of timesheet portal is storage format to store data performed by an individual worker on a particular day either on their own assigned or some other project of their choice were analyzed carefully.
* Plans of fulling the needs were made for instance, a worker can access his storage daily using his username and password and make changes according to the work performed. Dates would also be recorded.
* Suppose in our general daily life if you have your own a company or manage a team of employees and also manage the project ordered by a client you understand the challenges of motivating staff and establishing a culture of accountability and an understanding of objectives and goals.
* The best managers and leaders know that you cannot intimidate an employee into good performance, nor can you force them to care enough about delivering excellence and work productively. If you manage a group or team without using a time accounting system you are operating blindly or, at best, using an honour system.
* That may work well for a certain percentage of your employees but it will certainly cost you money in the long run. Even the most honest employee who wants to do a good job can find themselves spending time on things that are not top priority.

1.2Existing system and need of project:

* Timesheet tracking system is a practice of tracking time spent by employees. This is a set of data that helps to analyze how much time is spent on the project or a task they are assigned with.
* This data can be also utilized for the payroll calculation by the accounting and human resource department. This software helps to reduce employee time theft.
* Now a days with the help of advanced technology, we now have Al-powered software that helps to give accurate data and store all the important information on the cloud for you. To get the most efficiency we need tools that are easy to operate, low maintenance and are accessible remotely with ease.
* This project timesheet tracker log your time with or without tagging a project or task. Fully customize your timesheet. Give access to client managers for approval.
* Timesheet tracker software speeds up your pay and bill cycle and gives you the tools to track your profitability and manage projects and contracts in real time, online.
* Timesheet tracker is an adaptable web-based application, and will work for you whether you're project based, recruitment agency, service or manufacturer. Timesheet is a concept that rarely gets a lot of good press among employees. And it's no different even among the press.
* Admittedly, timesheets often live up to their bad reputation: nobody likes to sign a ton of papers just to "prove" they had worked several hours on a given task.
* This project is about Timesheet management its a practice of monitoring and analysing your personal and team timesheets as records of working hours across all your projects.

1.3Scope of work:

* This project can be implemented at any field of work to store data regarding the workers and their work.
* On an account of which calculating salaries and knowing the efficiency of the worker becomes easy.
* Large amounts of data can be stored and accessed efficiently and systematically. Privacy regarding the data is maintained, only authorized users can access the data.
* Timesheets are filled out regularly and truthfully.
* The quantitative timesheet data is supported with additional information and regular communication that provides a more descriptive account of the work.
* The working time used to be summarized on a sheet of paper, hence the name of the timesheet. Nowadays, online timesheets are much more popular, although there are several options here as well: from simple spreadsheets to AI-powered software tools.

1.4 Operating environment hardware and software:

* Hardware requirement:

1. RAM : 4GB
2. Disk Space : 16 GB
3. Processor: 2 266MHz processor
4. Mouse,Keyboard

* Software requirement:

1. Operating system
2. Java IDE
3. Java JDK
4. SQL

1.5Technology used:

* Front End:-

Java Swing

* Back End:-

MySQL

1.6Modules specification:

1.Employee login:

with the help of employee login module registered employee can be able to login into their account and add information about their work performed on the day or task completed by this for this day.

2.Administration login:

In administration login there is manager is able to see the work perform by the worker and when the work is completed.

3.Employee Dashboard management:

In employee dashboard management employee is see and update their information as per date as task given by the manager as like a task description and task completion time.

4.Manager dashboard login:

In manager dashboard manager is see which project or work for a project is perform by the employee under her. And arrange it update them. In this its also add project with project id, project name and other project details for show the project he also able to delete this project

5.Project management:

With the help of project management we able update project details with thes project name, project id, project description, client name etc… information.

6.personal details:

In personal details employee in this company are add their personal details and edit them as per their requirement and save it. Personal details as like a name, add, adhar no. mobile no. bank account no. etc….

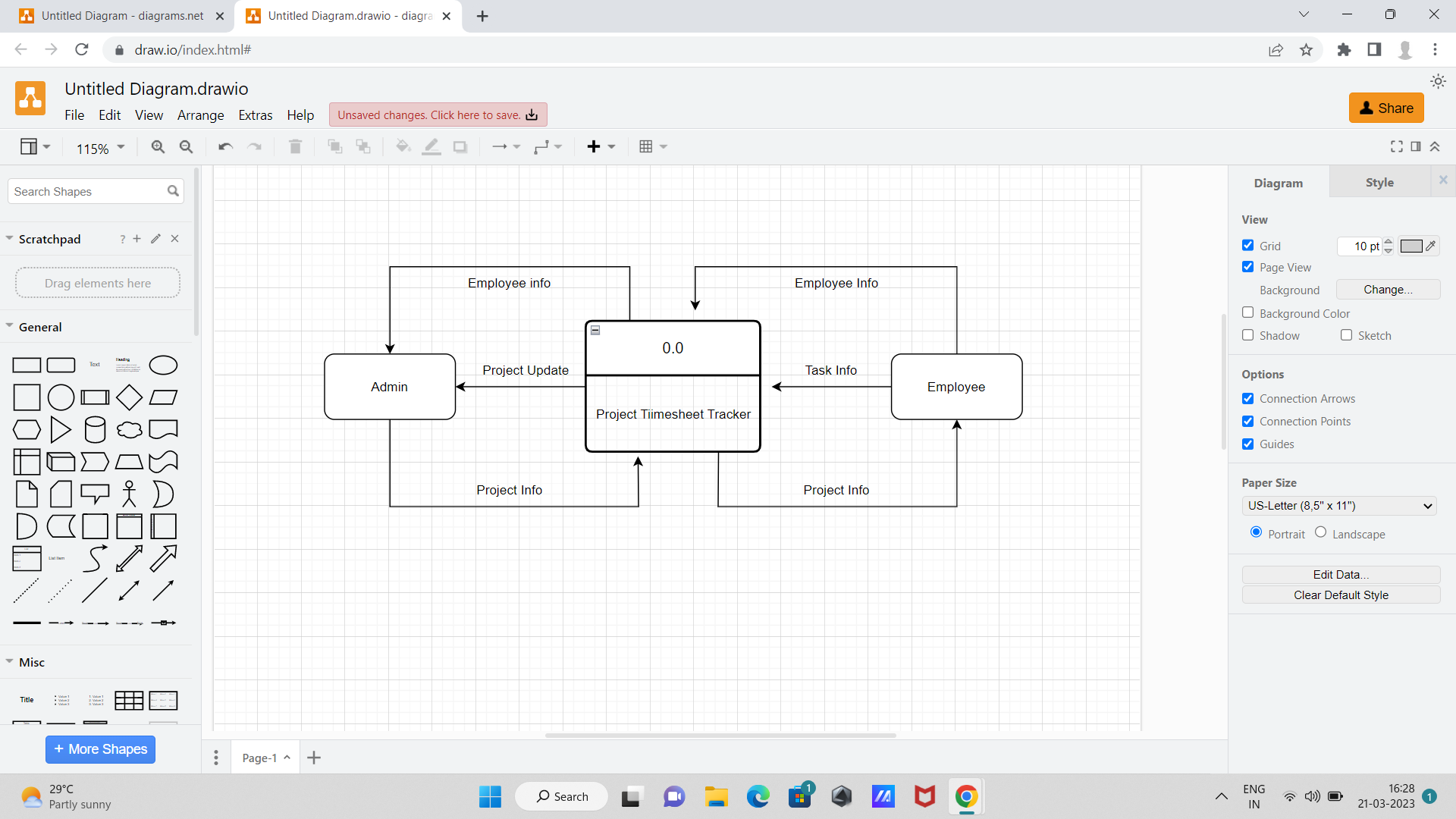
7.Task management:

Task management in this project is used for add task, view task, edit this task as with the help of date Peaker.

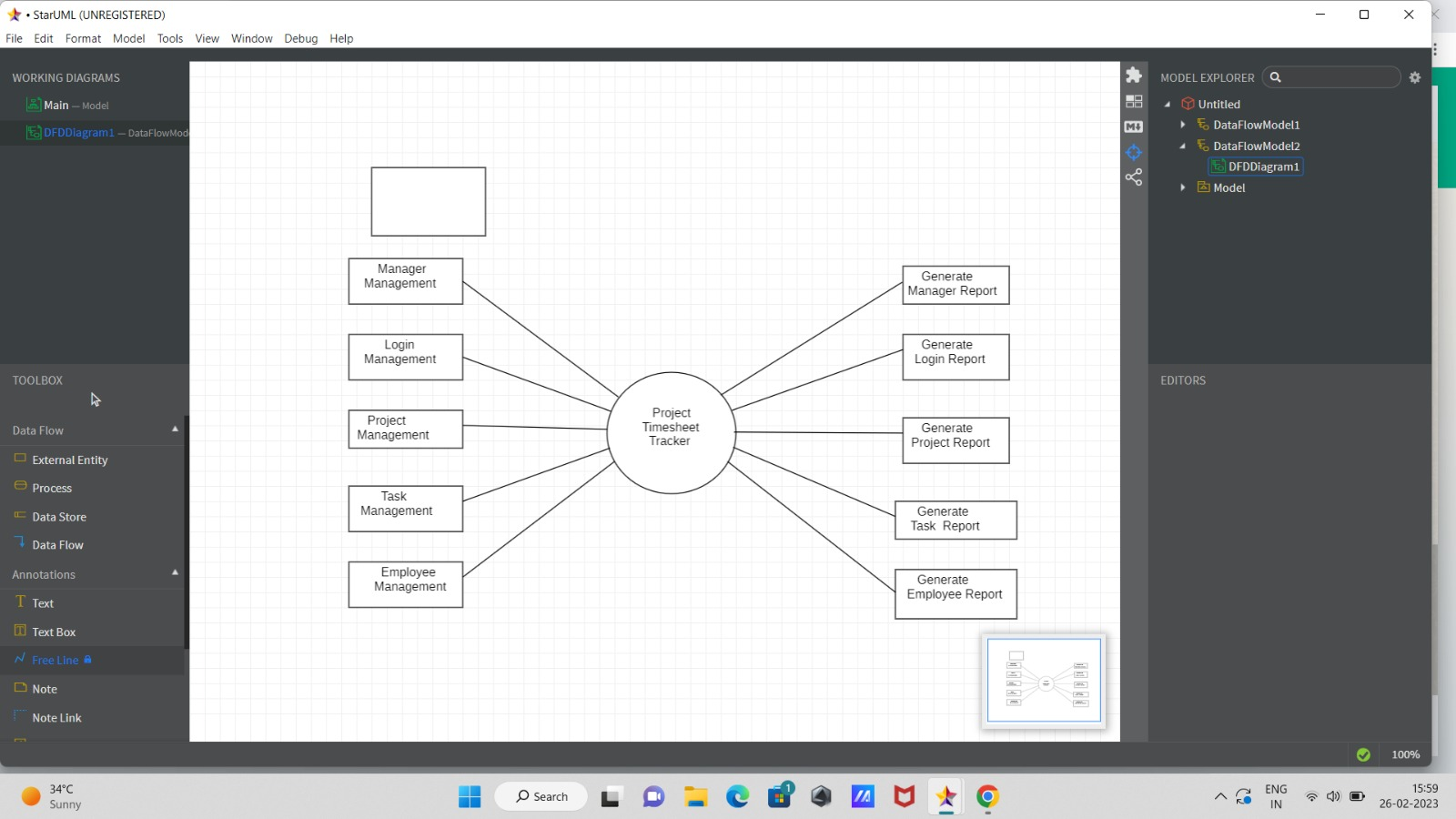
**Chapter 2: Analysis and design**

2.1Data flow Diagram:

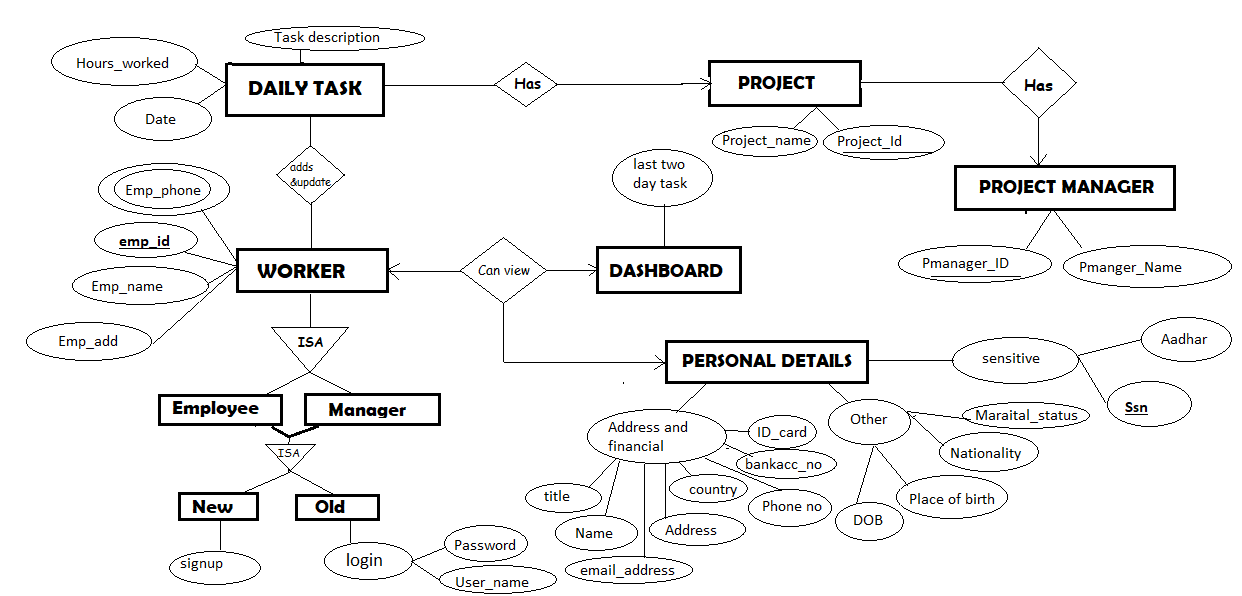
1.zero level DFD:



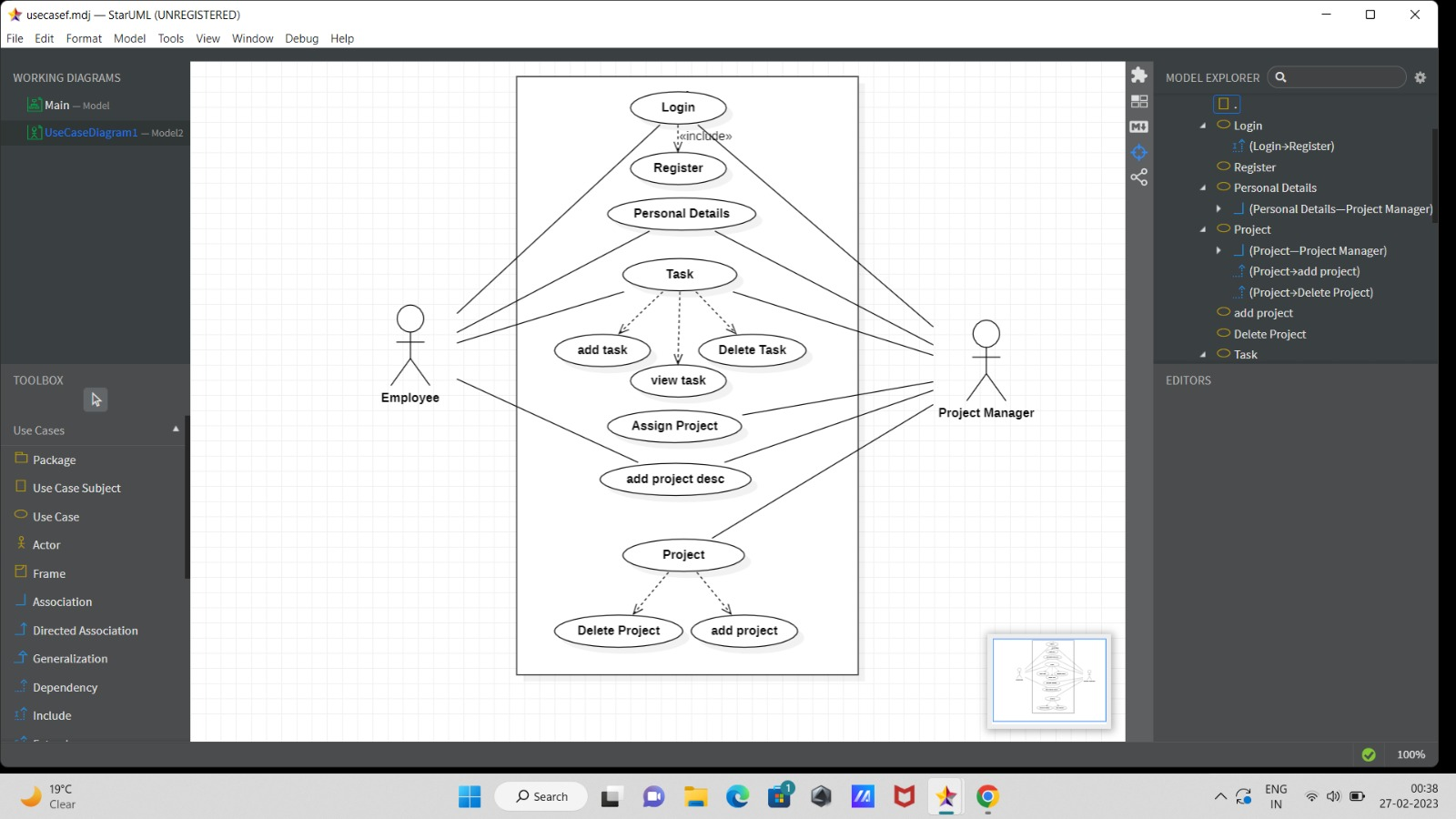
2.level 1st DFD:



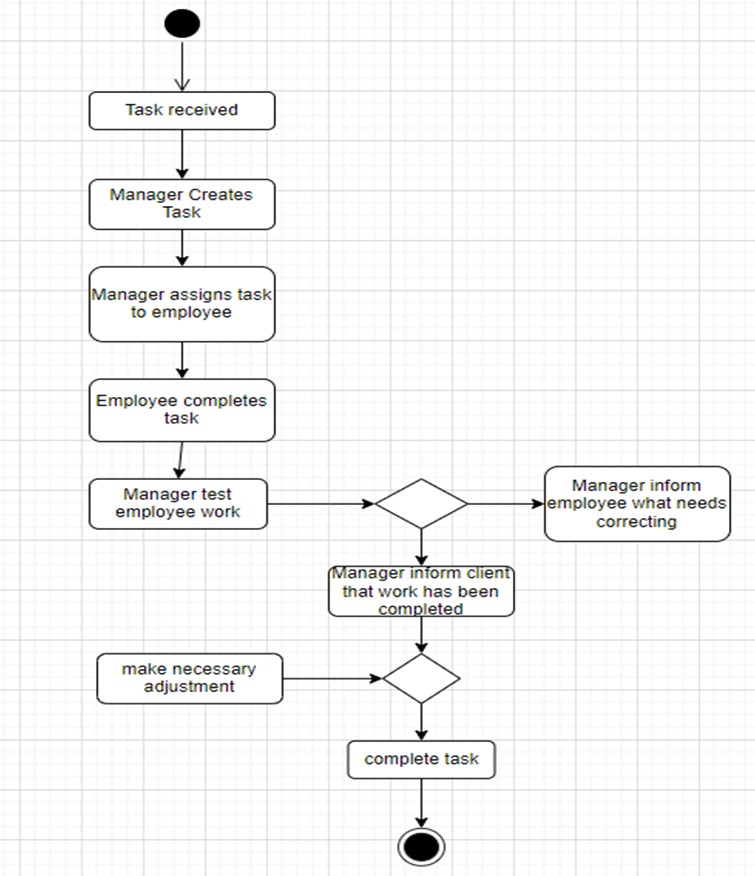
2.2Entity Relationship Diagram:



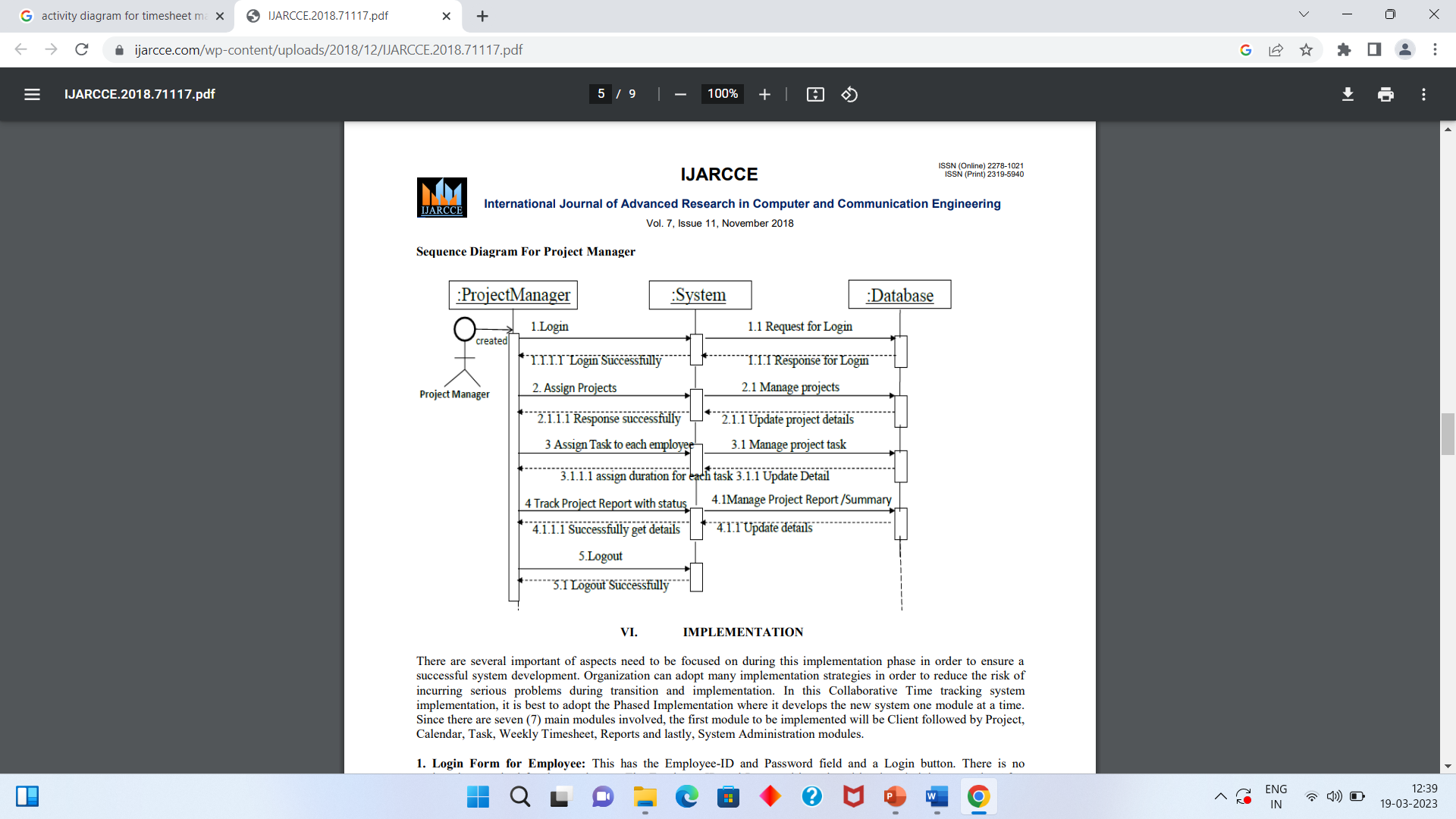
2.3Use case Diagram:



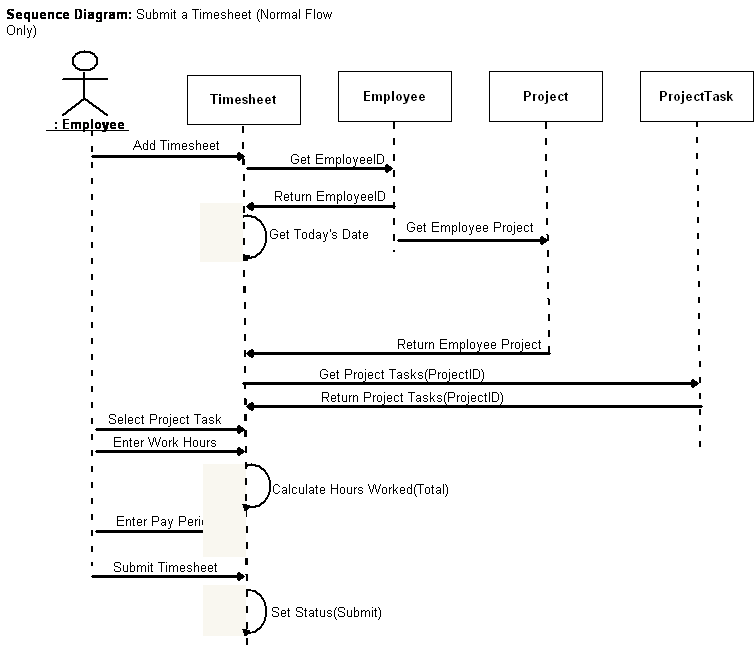
2.4Activity Diagram:



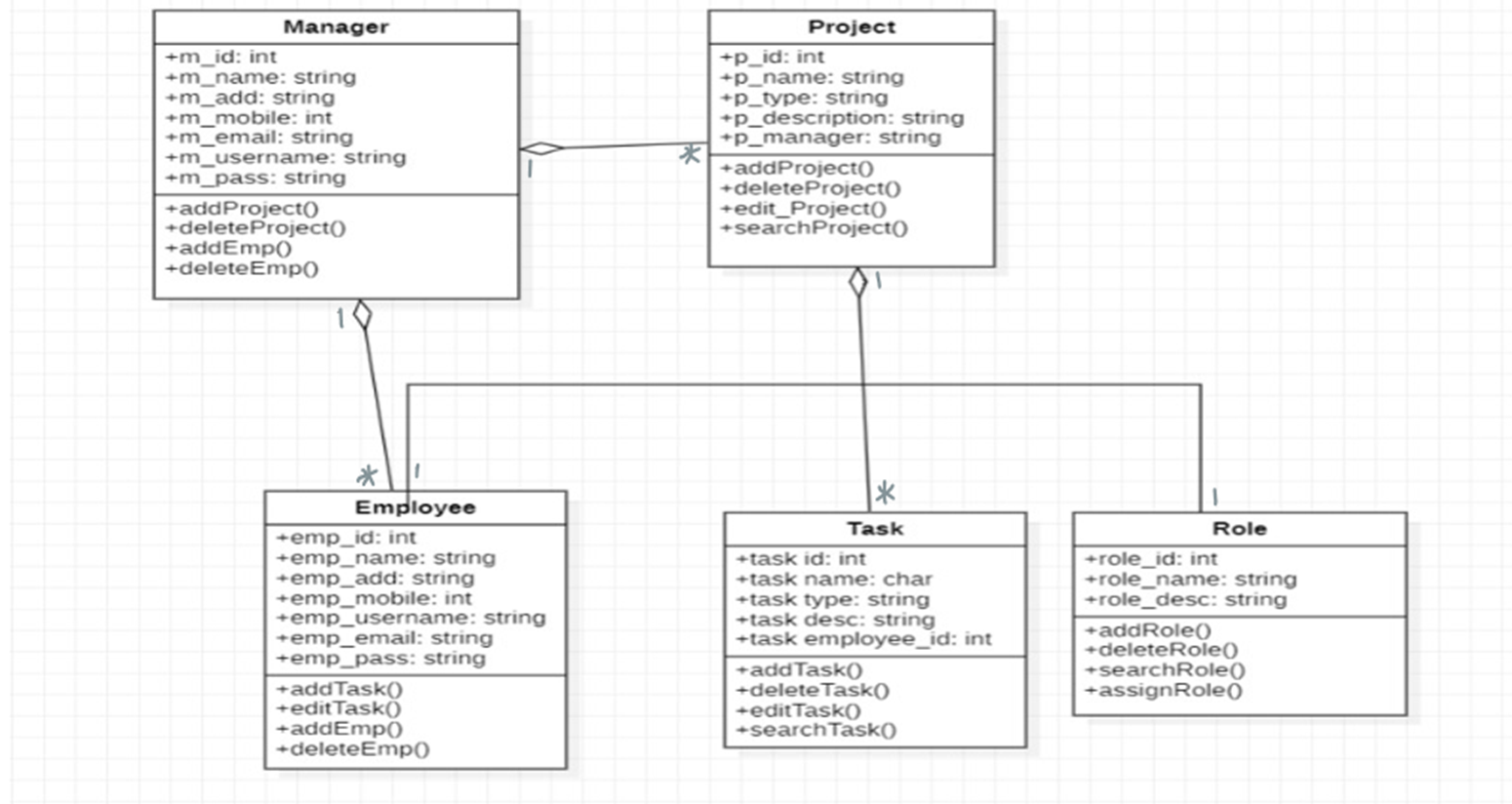
2.5Sequence diagram:

Project manager:

Sequence Diagram for employee:

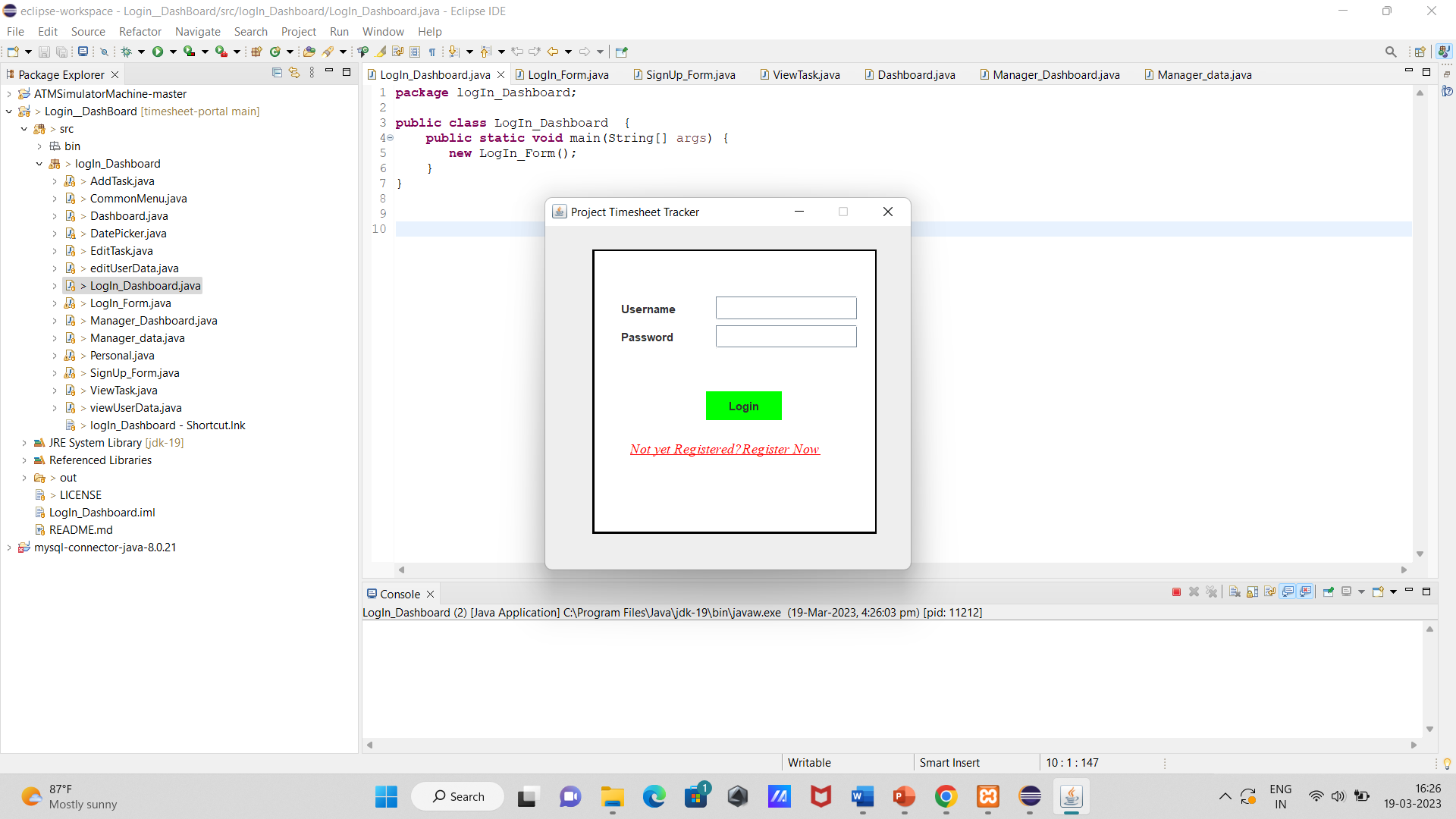


2.6Class Diagram:

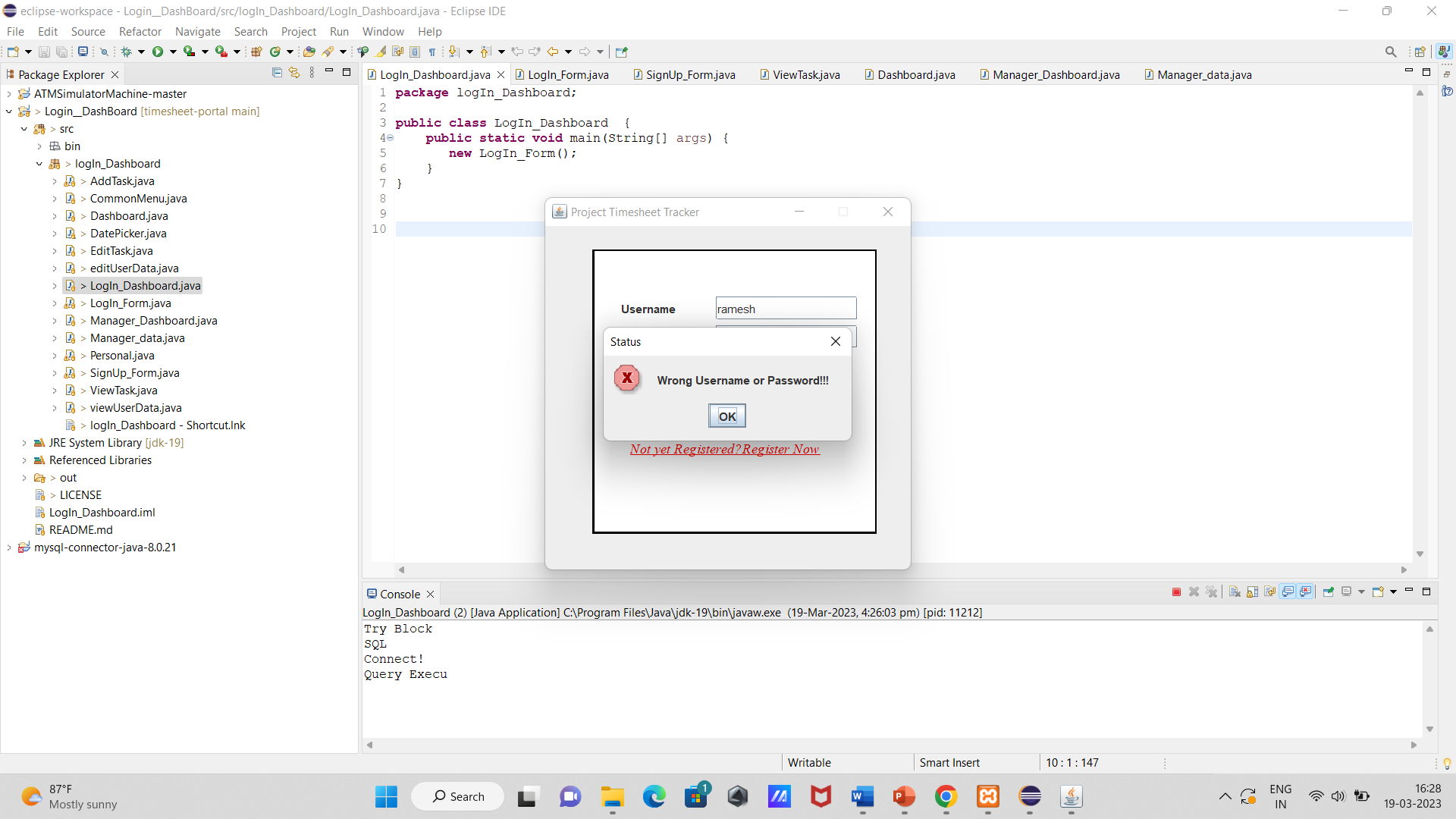


2.7User interface Screen:

1.login page:



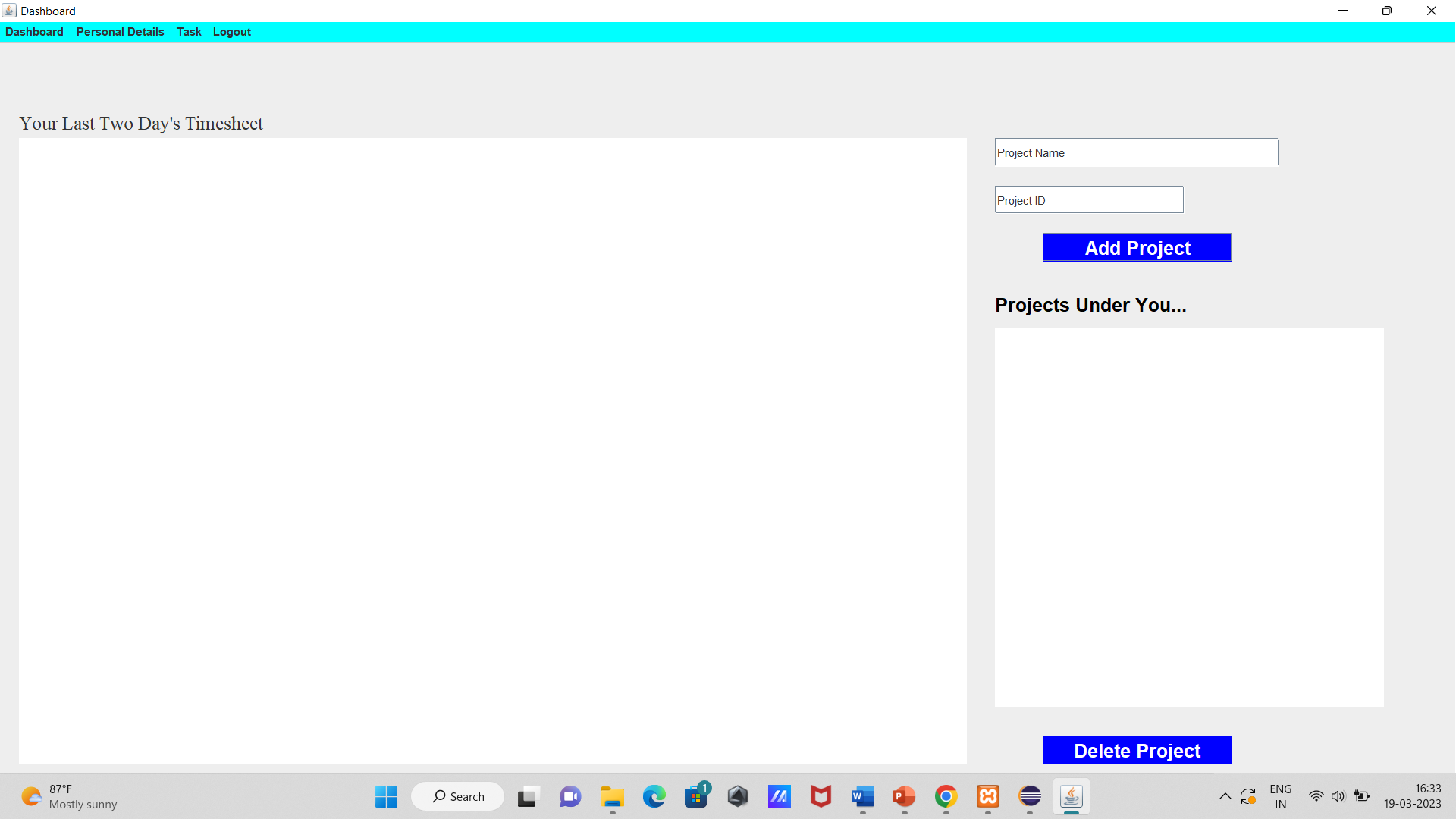
2.Login Validation Screen:



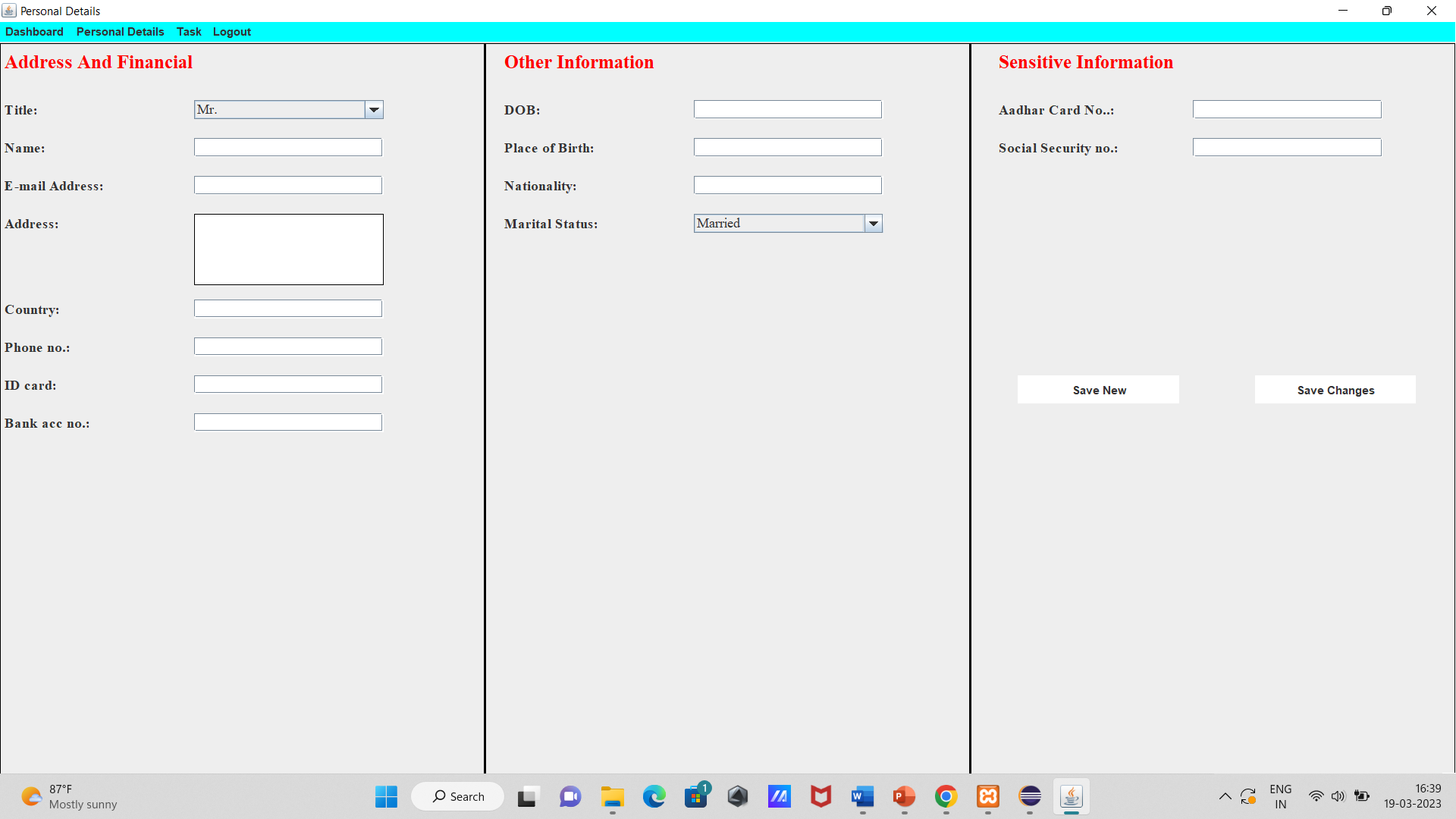
3.Registration page:



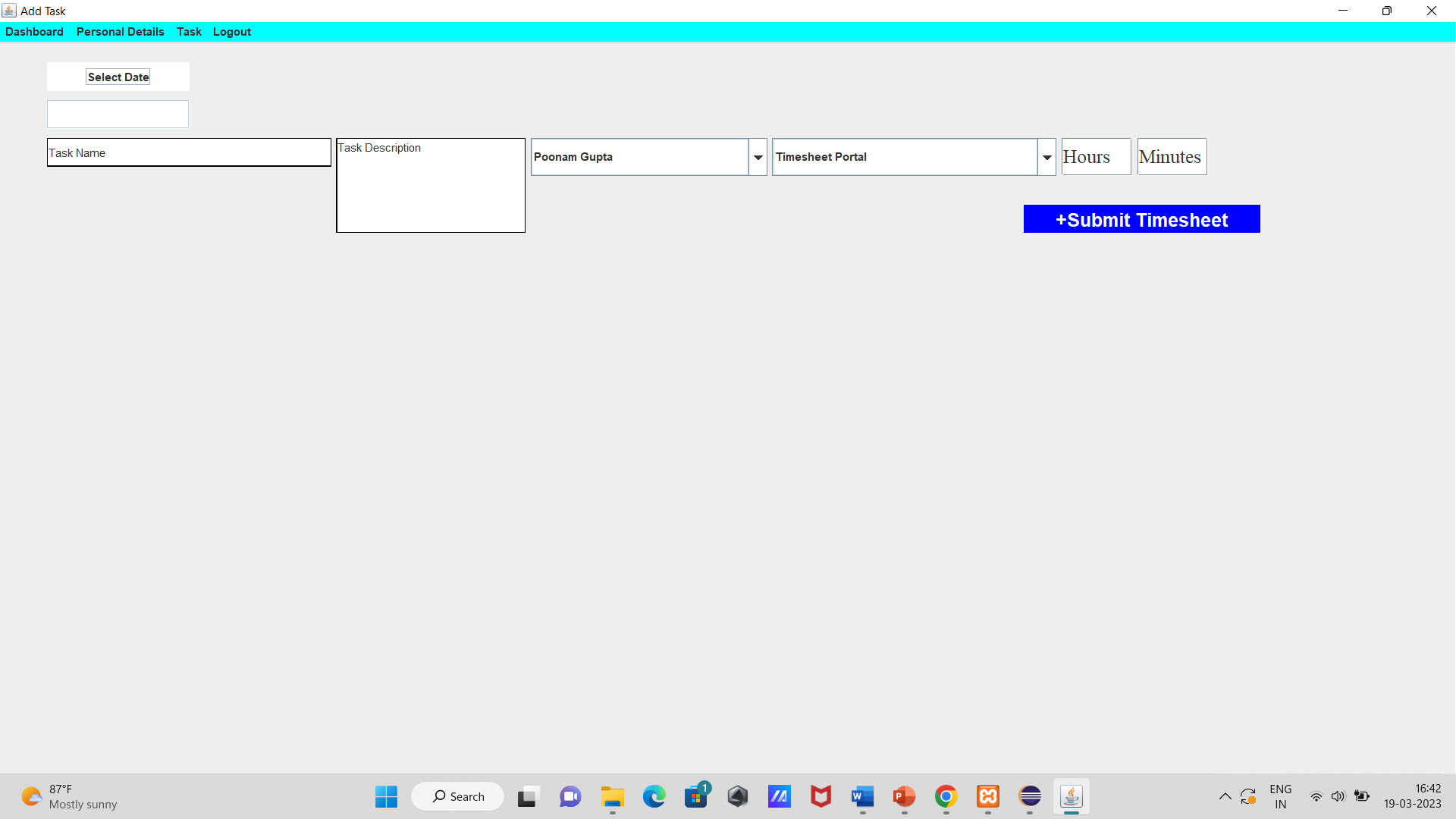
4.Add Project Screen:



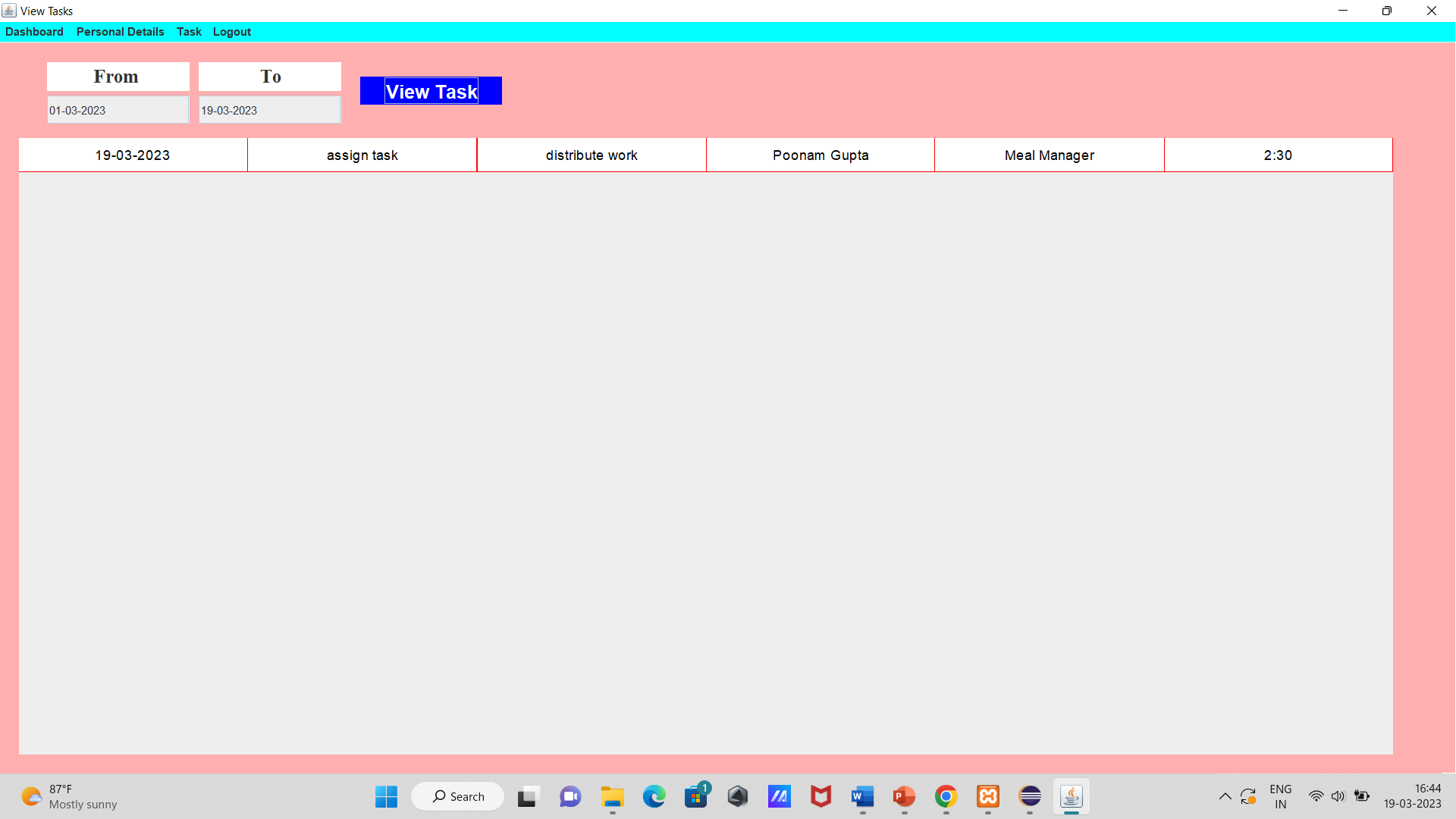
5.Personal Details:



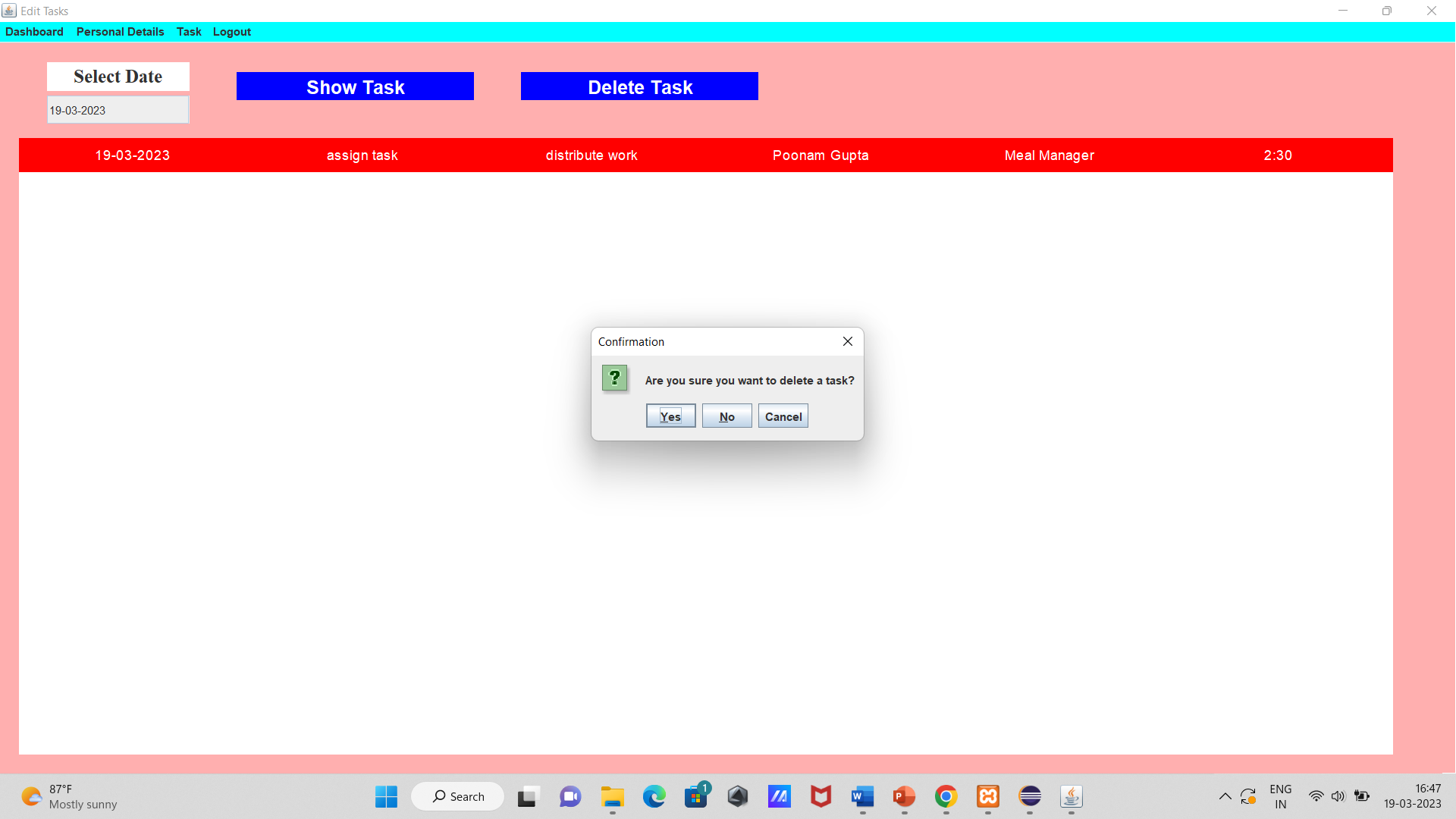
6.Edit task:



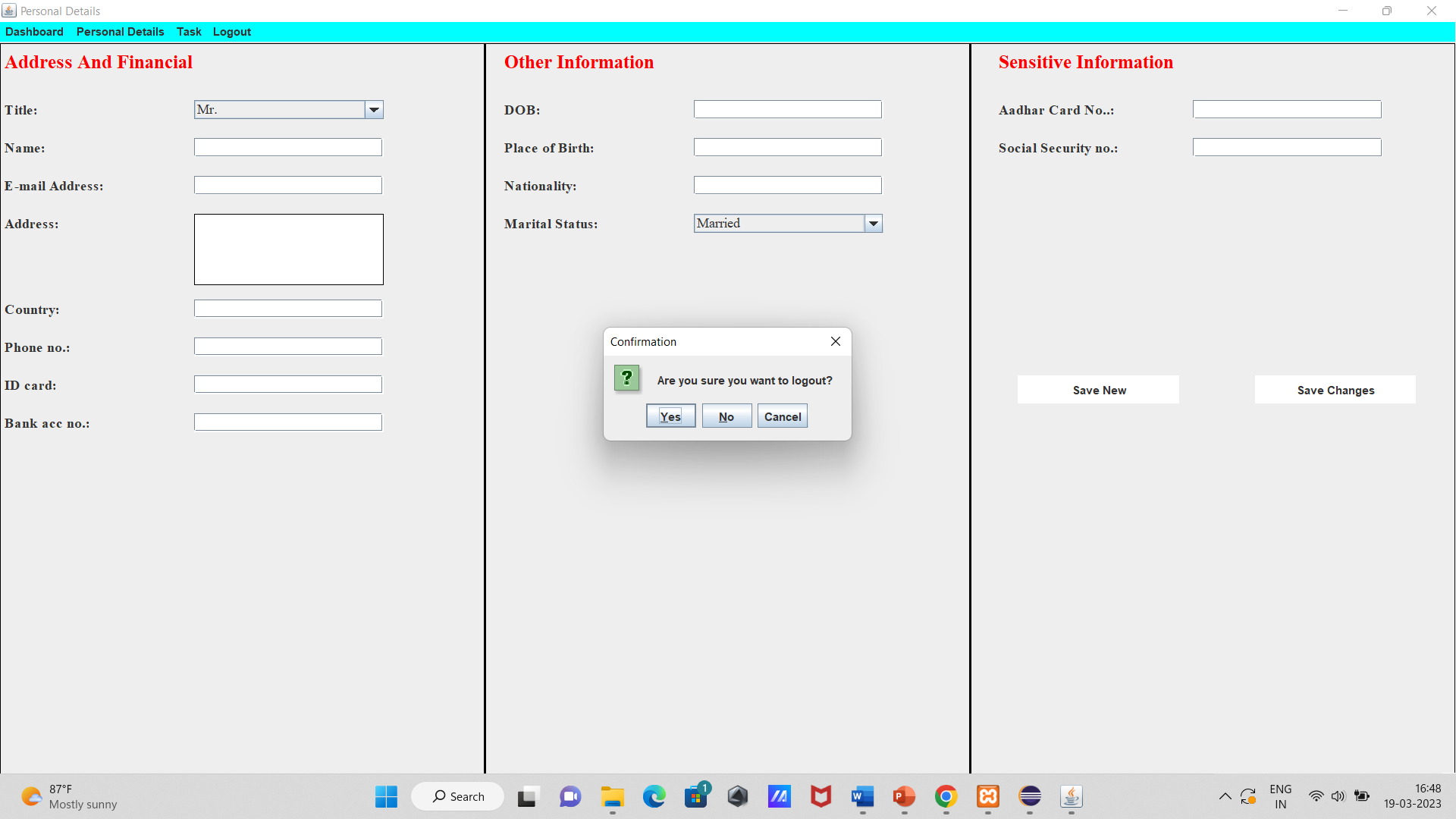
7.View Task:



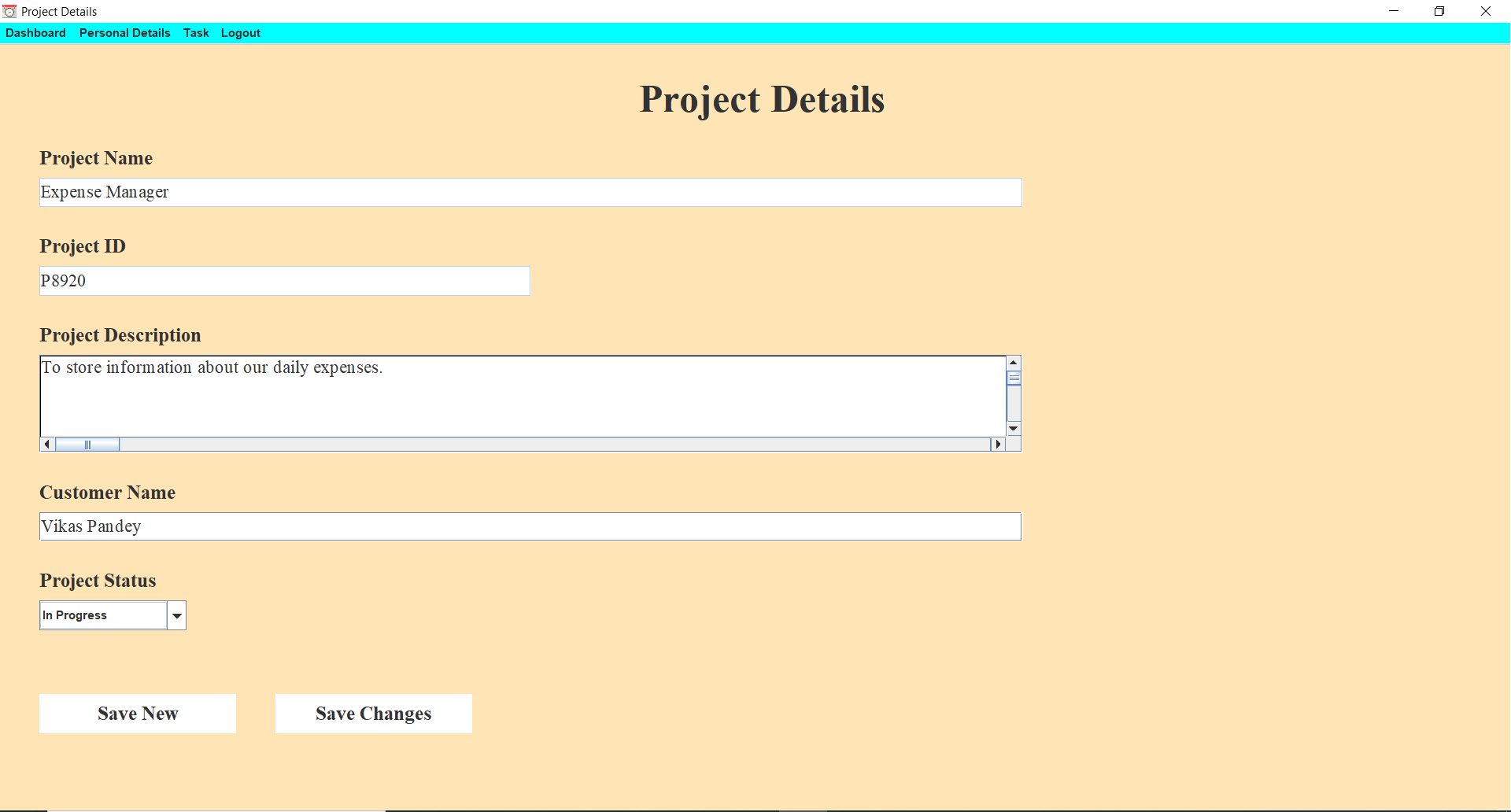
8.Delete task:



9.Logout:

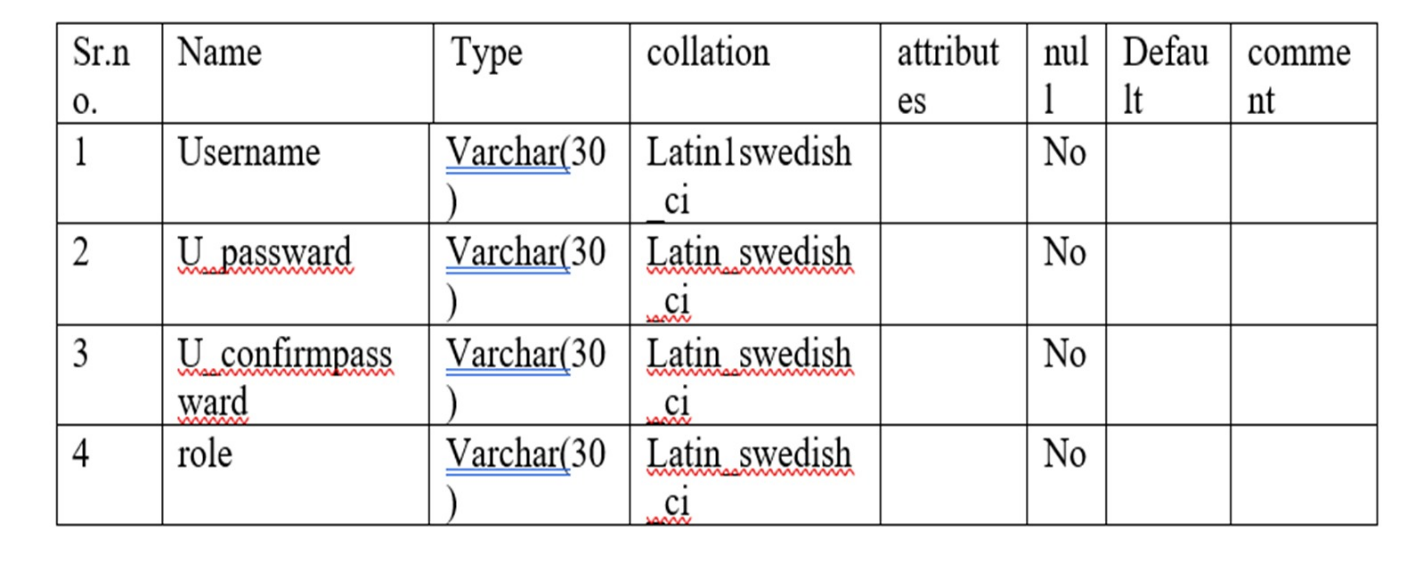


10.Project details:

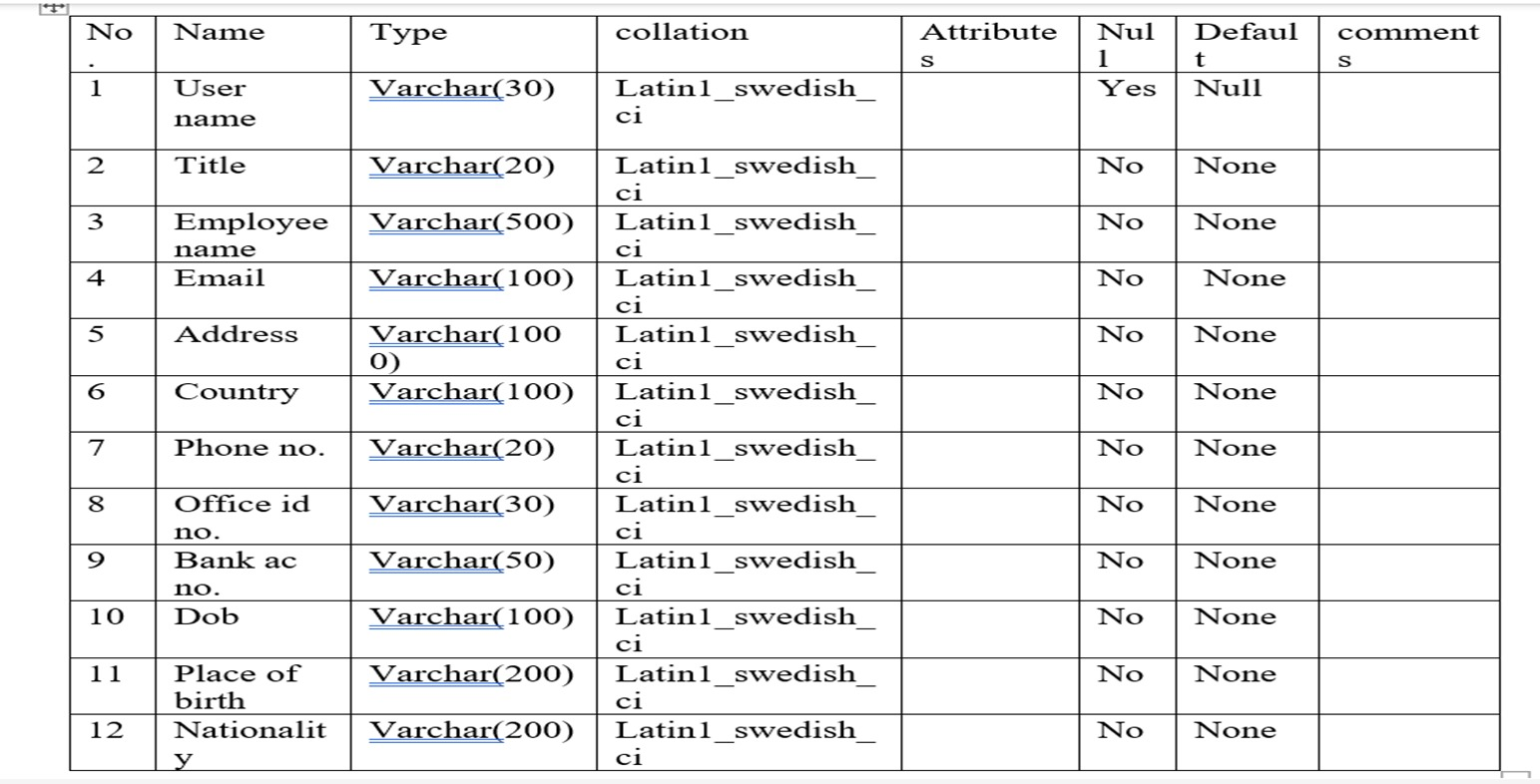


2.8 Table Structure:

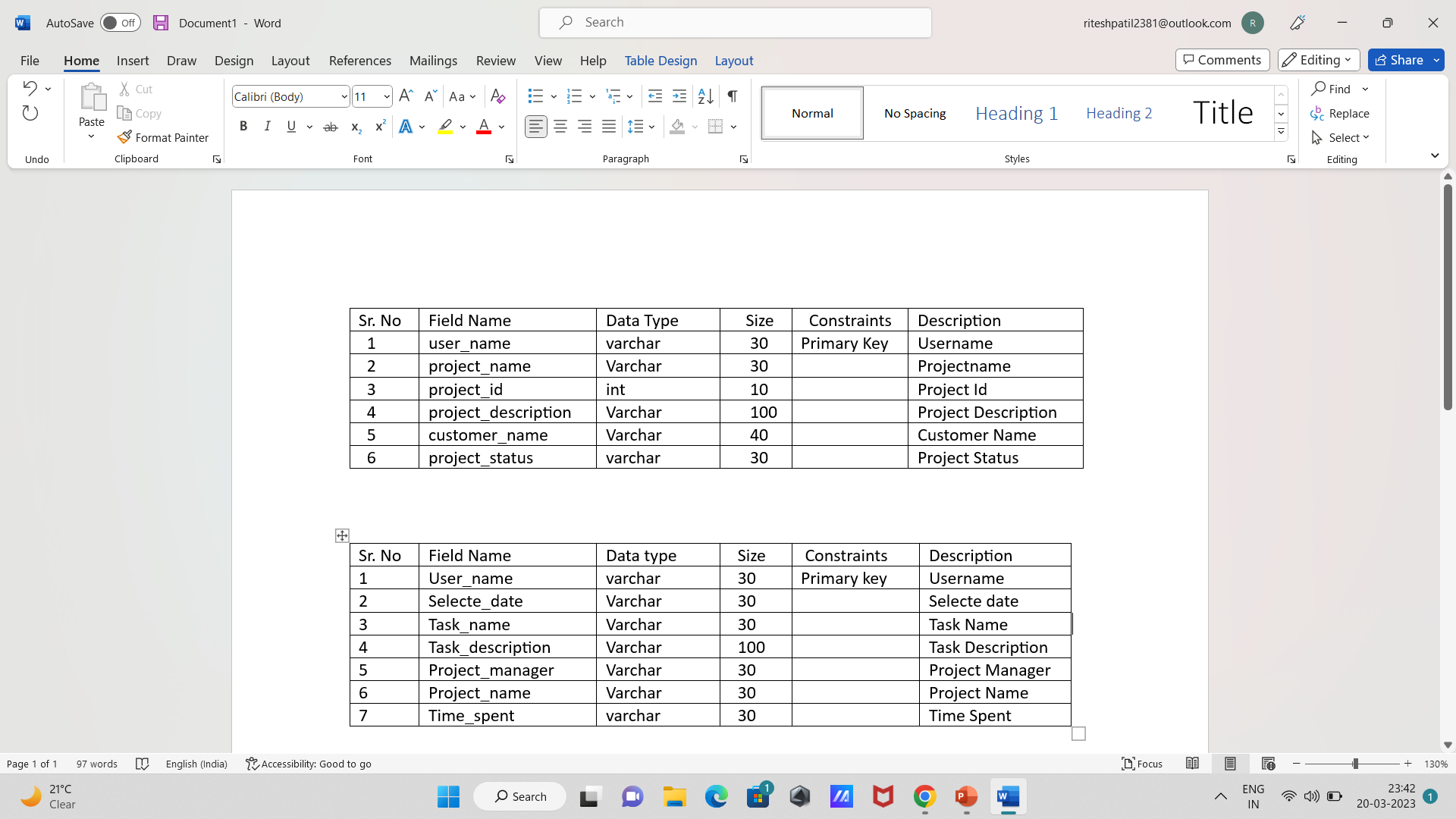
1.Login Data:



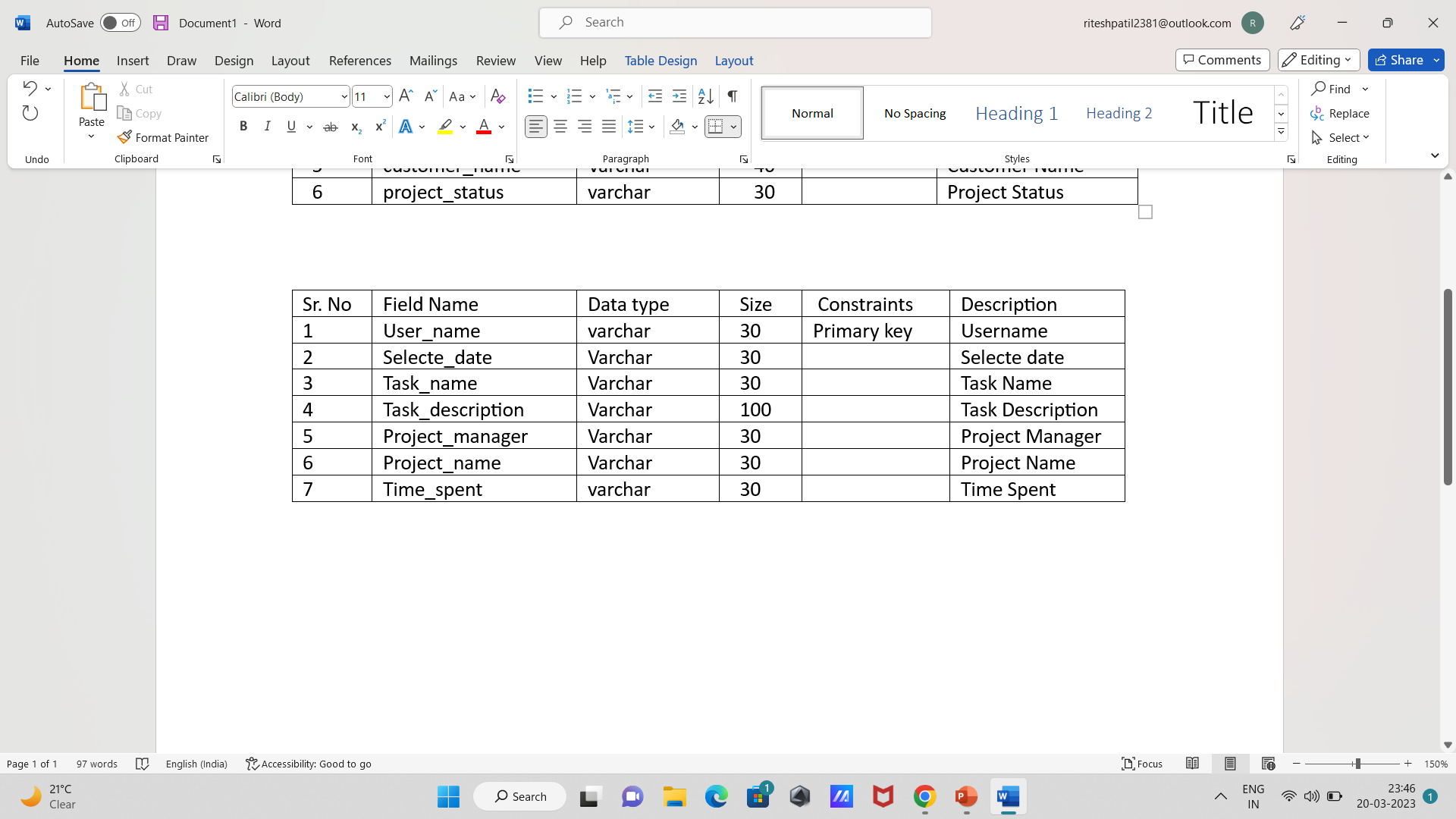
2.Personal Details:



3.Project Details:



4.Add Task:



**Chapter3: Drawbacks and limitations**

1. Decreased employee morale

Timesheet tracker tracking employees’ time carries the risk of having them feel like they are being spied on.

This is one of the most common fears that employees have when a time tracking system is implemented. No one likes to feel like they’re always being watched.

### 2. Complex systems can be expensive

The cost of an employee time tracking system is a legitimate concern.

Complex systems that allow automatic tracking and payroll or provide activity metrics and in-depth information can be expensive. But you don’t need to implement time tracking for the whole company right away.

Scalable solutions allow you to start with just a few of your employees. When you begin seeing positive ROI on your investment within a few weeks’ time, you can consider adopting the system more widely.

### 3. Tracking intangible benefits is hard

Not everything can be tracked and measured.

Brainstorming is an example of something that’s hard to analyse in that way. Sometimes it takes [hours of toiling to reach a breakthrough](https://bocoup.com/blog/developer-weeks) and resolve an issue in a matter of minutes.

From a time tracking perspective, all of this may look like wasted time. That’s the worry, anyway.

But time tracking doesn’t just aim to measure basic productivity metrics, like keys pushed or mouse clicks. What it provides is [greater transparency about how efforts are distributed](http://mattragland.com/my-30-day-experiment-in-time-tracking) throughout the day. That lets managers see how effort is turning into results.

And a good employee time tracking system allows employees to add notes to their activities to provide context. Screenshots might make it look like an employee wasn’t doing anything, but a note that they were brainstorming or researching clears up these issues quickly.

### 4.  Potential for micromanagement

Employees often fear they will be micromanaged non-stop once they start tracking their time.

And while certain managers may be tempted to do this, the point of tracking time is to **spend less time on managing** employees. These systems let managers supervise more efficiently without picking or observing every detail of the work process.

Time tracking allows managers to get a clear picture of an employee’s performance at a glance.

And based on that, managers can dig into the details if they feel an employee is lagging behind, experiencing difficulties, or simply slacking.

**Chapter4: Proposed Enhancement**

* Where the job title is required for the user at the time they logged time. For example, a user may have had a different job title in the past, or may have worked in the capacity of a different job title for a specific project
* The **Timesheets Expected From** date indicates the date from which timesheet completion should be checked and the **Expected Timesheets** setting enables you to choose the frequency (Daily, Weekly or Monthly) that timesheets are expected to be submitted by the user.
* The system checks the user’s capacity (based on their Clarizen calendar) from the “Timesheets Expected From” date using the time frames indicated by “Expected Timesheet” and compares that with the sum of the submitted and approved timesheets for the same time period.
* If the hours worked meet or exceed the capacity then the “Timesheets Complete Until” date is populated with the date of the first working day in the next “Expected Timesheet” period. This then indicates the starting date of the next incomplete timesheet.

**Chapter 5: Conclusion**

This project can be implemented at any field of work to store data regarding the workers and their work. On an account of which calculating salaries and knowing the efficiency of the worker becomes easy. Large amounts of data can be stored and accessed efficiently and systematically. Privacy regarding the data is maintained, only authorized users can access the data.

A good timesheet software paves the way for a productive workforce and allows them to be efficient in their work. On blink provides you with a one-stop solution for an effective timesheet and leave management system that helps you improve your [employee-employer relationship](https://www.onblick.com/blogs/timesheets-for-a-valid-employer-employee-relationship-and-dol-compliance) with minimal effort. Sign up for a quick [demo](https://www.onblick.com/schedule-your-demo) to engage with a timesheet management tool that can coherently improve the way you work.

**Chapter 6: Bibliography**

References used:

* <http://en.wikipedia.org/wiki/>
* [www.w3schools.com](http://www.w3schools.com)
* <http://stackoverflow.com/>
* <https://github.com>
* <https://codepen.io/tag/html-template>
* [www.javatpoint.com](http://www.javatpoint.com)

**Sample Code:**

Login.java

**package** logIn\_Dashboard;

**public** **class** LogIn\_Dashboard {

**public** **static** **void** main(String[] args) {

**new** LogIn\_Form();

}

}

Manager\_dashboard.java

package logIn\_Dashboard;

import javax.imageio.ImageIO;

import javax.swing.\*;

import javax.swing.table.DefaultTableCellRenderer;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Calendar;

import java.util.Date;

public class Manager\_Dashboard {

JFrame frame;

JPanel panel,tablepanel,tablepanel2;

Font font = new Font("",Font.BOLD,20);

JLabel welcome,label,projects\_under\_u;

JTextField Project,Project\_ID;

JButton addproject,deleteproject,showdetails;

JTable table,table2;

Object[] row,row2;

DefaultTableModel model,model2;

String prev1,prev2;

String name;

Manager\_Dashboard(){

frame = new JFrame("Dashboard");

panel = new JPanel();

frame.setJMenuBar(CommonMenu.displayMenu(frame));

frame.setLayout(null);

panel.setLayout(null);

panel.setBackground(Color.decode("#FFE4B5"));

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

frame.setSize(screenSize.width,screenSize.height);

frame.setBounds(0,0,screenSize.width, screenSize.height);

panel.setSize(screenSize.width,screenSize.height);

try {

BufferedImage i = ImageIO.read(new File("C:\\Users\\DELL\\Desktop\\Projects\\Login\_\_DashBoard\\src\\logIn\_Dashboard\\Images\\Timesheet.png"));

frame.setIconImage(i);

} catch (IOException e) {

e.printStackTrace();

}

Date date = new Date();

SimpleDateFormat formatter = new SimpleDateFormat("dd-MM-yyyy");

formatter.format(date);

Calendar cal = Calendar.getInstance();

cal.add(Calendar.DATE, -1);

prev1=formatter.format( cal.getTime())

Calendar cal1 = Calendar.getInstance();

cal1.add(Calendar.DATE, -2);

prev2=formatter.format( cal1.getTime());

try {

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

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3307/testdb","root",""); String query = "select \* from personal\_deatils where user\_name=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, LogIn\_Form.userText.getText());

ResultSet rs = ps.executeQuery();

if(rs.next()) {

name=rs.getString("employee\_name");

welcome = new JLabel("Hello " +name+" !!!");

welcome.setBounds(20,30,1000,35);

welcome.setFont(new Font("Times New Roman",Font.BOLD,30));

panel.add(welcome);

}

}catch(Exception e1) {

System.out.println(e1);

}

label = new JLabel("Your Last Two Day's Timesheet");

label.setBounds(20,60,300,50);

label.setFont(new Font("Times New Roman",Font.PLAIN,20));

panel.add(label);

table = new JTable();

row = new Object[6];

tablepanel = new JPanel();

tablepanel.setLayout(new BorderLayout());

tablepanel.setBounds(20,100,1000,660);

DefaultTableCellRenderer tableRenderer = new DefaultTableCellRenderer();

tableRenderer.setHorizontalAlignment(JLabel.CENTER);

table.setDefaultRenderer(Object.class, tableRenderer);

table.setBounds(20,100,1000,660);

model = new DefaultTableModel();

table.setModel(model);

model.addColumn("Date");

model.addColumn("Task Name");

model.addColumn("Task Description");

model.addColumn("Project Manager");

model.addColumn("Project Name");

model.addColumn("Time");

table.setBackground(Color.white);

table.setForeground(Color.BLACK);

table.setSelectionBackground(Color.RED);

table.setGridColor(Color.red);

table.setSelectionForeground(Color.white);

table.setFont(new Font("", Font.PLAIN,15));

table.setRowHeight(table.getRowHeight()+20);

JScrollPane pane = new JScrollPane(table);

pane.setVisible(true);

tablepanel.add(pane);

panel.add(tablepanel);

Project = new JTextField("Project Name");

Project .setBounds(1050,100,300,30);

Project.addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

if(Project.getText().equals("Project Name")) {

Project.setText("");

}

}

@Override

public void focusLost(FocusEvent e) {

if(Project.getText().equals("")) {

Project.setText("Project Name");

}

}

});

panel.add(Project);

Project\_ID = new JTextField("Project ID");

Project\_ID.setBounds(1050,150,200,30);

Project\_ID.addFocusListener(new FocusListener() {

@Override

public void focusGained(FocusEvent e) {

if(Project\_ID.getText().equals("Project ID")) {

Project\_ID.setText("");

}

}

@Override

public void focusLost(FocusEvent e) {

if(Project\_ID.getText().equals("")) {

Project\_ID.setText("Project ID");

}}});

panel.add(Project\_ID);

addproject= new JButton("Add Project");

addproject.setBounds(1100,200,200,30);

addproject.setForeground(Color.white);

addproject.setFont(font);

addproject.setBackground(Color.blue);

addproject.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

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

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb","root","");

if(Project.getText().equals("Project Name") || Project.getText().equals("")) {

JOptionPane.showMessageDialog(null, "Enter Project Name...");

}

else if(Project\_ID.getText().equals("Project ID") || Project\_ID.getText().equals("")) {

JOptionPane.showMessageDialog(null, "Enter Project ID...");

}

else {

String query4 = "Select \* from project\_data where project\_id=?;";

PreparedStatement ps = con.prepareStatement(query4);

ps.setString(1, Project\_ID.getText());

ResultSet rs = ps.executeQuery();

if(rs.next()) {

JOptionPane.showMessageDialog(null, "ID Already Exists...."+"\n"+"Enter Correct One...","Warning",JOptionPane.WARNING\_MESSAGE);

}

String query1 = "insert into project\_data values(?,?,?,?);";

PreparedStatement ps1 = con.prepareStatement(query1);

ps1.setString(1, LogIn\_Form.userText.getText());

String query2 = "select \* from personal\_deatils where user\_name=?;";

PreparedStatement ps2 = con.prepareStatement(query2);

ps2.setString(1, LogIn\_Form.userText.getText());

ResultSet rs2 = ps2.executeQuery();

if(rs2.next()) {

name=rs2.getString("employee\_name");

}

ps1.setString(2, name);

ps1.setString(3, Project.getText());

ps1.setString(4, Project\_ID.getText());

int rs1 = ps1.executeUpdate();

if(rs1>0) {

JOptionPane.showMessageDialog(null, "Project Added Successfully...");

}

}

}catch(Exception e1) {

System.out.println(e1);

}

}

});

panel.add(addproject);

projects\_under\_u = new JLabel("Projects Under You...");

projects\_under\_u.setBounds(1050, 260, 400, 30);

projects\_under\_u.setForeground(Color.black);

projects\_under\_u.setFont(font);

panel.add(projects\_under\_u);

table2 = new JTable();

row2 = new Object[2];

tablepanel2 = new JPanel();

tablepanel2.setLayout(new BorderLayout());

tablepanel2.setBounds(1050,300,410,400);

DefaultTableCellRenderer tableRenderer2 = new DefaultTableCellRenderer();

tableRenderer2.setHorizontalAlignment(JLabel.CENTER);

table2.setDefaultRenderer(Object.class, tableRenderer2);

table2.setBounds(1050,300,410,400);

model2 = new DefaultTableModel();

table2.setModel(model2);

model2.addColumn("Project Name");

model2.addColumn("Project ID");

table2.setBackground(Color.white);

table2.setForeground(Color.BLACK);

table2.setSelectionBackground(Color.red);

table2.setGridColor(Color.red);

table2.setSelectionForeground(Color.white);

table2.setFont(new Font("", Font.PLAIN,15));

table2.setRowHeight(table.getRowHeight()+5);

JScrollPane pane2 = new JScrollPane(table2);

pane2.setVisible(true);

tablepanel2.add(pane2);

panel.add(tablepanel2);

deleteproject = new JButton("Delete Project");

deleteproject.setForeground(Color.WHITE);

deleteproject.setBorderPainted(false);

deleteproject.setBackground(Color.BLUE);

deleteproject.setFont(font);

deleteproject.setBounds(1050,730,200,30);

deleteproject.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int result = JOptionPane.showConfirmDialog(null, "Are you sure you want to delete a project?","Confirmation",JOptionPane.YES\_NO\_CANCEL\_OPTION,JOptionPane.QUESTION\_MESSAGE);

if(result == JOptionPane.YES\_OPTION){

int row = table2.getSelectedRow();

String col1 = table2.getModel().getValueAt(row, 0).toString();

String col2 = table2.getModel().getValueAt(row, 1).toString();

model2.removeRow(row);

try {

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

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb","root","");

String query ="delete from project\_data where user\_name=? and project\_name=? and project\_id=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, LogIn\_Form.userText.getText());

ps.setString(2, col1);

ps.setString(3, col2);

ps.executeUpdate();

}catch(Exception e1) {

System.out.println(e1);

}

}

}

});

panel.add(deleteproject);

showdetails = new JButton("Show Details");

showdetails.setForeground(Color.WHITE);

showdetails.setBorderPainted(false);

showdetails.setBackground(Color.BLUE);

showdetails.setFont(font);

showdetails.setBounds(1300, 730, 200, 30);

showdetails.addActionListener(new ActionListener(){

@Override

public void actionPerformed(ActionEvent e) {

int row = table2.getSelectedRow();

String col1 = table2.getModel().getValueAt(row, 0).toString();

String col2 = table2.getModel().getValueAt(row, 1).toString();

new Project\_details(col1,col2);

frame.dispose();

}

});

panel.add(showdetails);

frame.add(panel);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

showTask();

showProject();

}

public ArrayList<viewUserData> taskList(){

ArrayList<viewUserData> tasksList = new ArrayList<>();

try {

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

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb","root","");

viewUserData user;

String query = "select \* from add\_task where user\_name=? and selected\_date>=? and selected\_date<=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, LogIn\_Form.userText.getText());

ps.setString(2, prev2);

ps.setString(3, prev1);

ResultSet rs = ps.executeQuery();

while(rs.next()) {

user = new viewUserData(rs.getString("user\_name"),rs.getString("selected\_date"),rs.getString("task\_name"),rs.getString("task\_description")

,rs.getString("project\_manager"),rs.getString("project\_name"),rs.getString("time\_spent"));

tasksList.add(user);

}

}catch(Exception e1) {

System.out.println(e1);

}

return tasksList;

}

public void showTask() {

ArrayList<viewUserData> list = taskList();

for(int i=0;i<list.size();i++) {

row[0]=list.get(i).getselected\_date();

row[1]=list.get(i).gettask\_name();

row[2]=list.get(i).gettask\_description();

row[3]=list.get(i).getproject\_manager();

row[4]=list.get(i).getproject\_name();

row[5]=list.get(i).gettime\_spent();

model.addRow(row);

}

}

public ArrayList<Manager\_data> projectList(){

ArrayList<Manager\_data> projectssList = new ArrayList<>();

try {

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

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3307/testdb","root","");

Manager\_data manager;

String query = "select \* from project\_data where user\_name=?;";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, LogIn\_Form.userText.getText());

ResultSet rs = ps.executeQuery();

while(rs.next()) {

manager = new Manager\_data(rs.getString("project\_id"),rs.getString("project\_name"));

projectssList.add(manager);

}

}catch(Exception e1) {

System.out.println(e1);

}

return projectssList;

}

public void showProject() {

ArrayList<Manager\_data> list = projectList();

for(int i=0;i<list.size();i++) {

row2[0]=list.get(i).getproject\_id();

row2[1]=list.get(i).getproject\_name();

model2.addRow(row2);

}

}

}