# 

**EMPLOYEE PERFORMANCE APPRISAL AND RATING PORTAL**

**A REPORT**

**BY**

**Naga Sowmya .V(AP18110020145)**

**Manikanta .Y (AP18110020147)**

**Rajya Talatam (AP18110020151)**

**Pavan Kumar .M (AP18110020155)**

**Sravani .G (AP18110020182)**

Submittedin partial fulfillment of the requirements for

UROP Project

In **ELECTRIONIC COMMUNICATION & ENGINEERING**

Under the guidance of

**Mr. SUJITH KALLURI**



**Department of Electronic Communication & Engineering**

**SRM UNIVERSITY – AP, AMARAVATHI**

**SRM UNIVERSITY AP**

**Title**

**EMPLOYEE PERFORMANCE APPRISAL AND RATING PORTAL**

**BY**

**Naga Sowmya .V(AP18110020145)**

**Manikanta .Y (AP18110020147)**

**Rajya Talatam (AP18110020151)**

**Pavan Kumar .M (AP18110020155)**

**Sravani .G (AP18110020182)**

**The UROP Committee certifies that this report complies with the regulations and meets the standard required for completion of**

**UROP Project**

**Report Advisor: Mr. SUJITH KALLURI**

**Committee Member: Mr.Anirban Ghosh**

**Head of the Department:**

**Date: -05-2021 Mr.SivaSankarYellampalli**

**Table of Contents:**

ABSTRACT……….……....…………………………………………………………………….3

PROBLEM DEFINITION……..….………………………………………………………….....4

SYSTEM ANALYSIS…….…….………..…………………………………………….……..4-5

EXISTING SYSTEM…………..……………………………………………………………..4-5

PROPOSED SYSTEM…………….…………...……………………………………………….5

PROJECT OVERVIEW…………………………………………………………………………6

PROJECT MODULES………………………………………………………………………..6-9

DEFINITION ACRONYMS ABBREVIATIONS………………………………………………9

SYSTEM REQUIREMENTS…………………..……………………………………………9-10

TECHNOLOGIES…………………………………………………………………..……...10-22

SYSTEM DESIGN………………………………………………………………………....22-25

SEQUENCE DIAGRAM…….…………………………………………………………………26

ACTIVITY DIAGRAMS………………………………………………………………...…27-29

SCREEN LAYOUT…………………………………………………………………………30-34

DATA BASE DESIGN……………………………………………………………………..35-37

TEST REPORT AND TEST PLAN………………………………………………………..38-39

SYSTEM INTEGRATION TESTING……………………………………………………..40-41

CONCLUSION……………………………………………………………………………..41-42

BIBLIOGRAPHY…….………………………………………………………………………..42

**ACKNOWLEDGEMENTS:**

We wish to take this opportunity to express our deep gratitude to all those who helped, encouraged, motivated, and have extended their cooperation in various ways during our project work. It is our pleasure to acknowledge the help of all those individuals who was responsible for foreseeing the successful completion of our project.

We would like to thank **Mr. SUJITH KALLURI**  and express our gratitude with great admiration and respect. For the valuable advice and help throughout the development of this project by providing us with required information without those guidance, cooperation and encouragement, this project could not have been materialized.

Last but not the least; we would like to thank the entire respondents for extending their help in all circumstances.

**ABSTRACT:**

The main goal of this project is to determine what kind of system would be appropriate for a given problem and how to create it. The system's specifications can be specified by looking at the current system and its issues. They talk about the new system that will be installed and what they expect from it.

**Software requirements:**

Windows 7 is the operating system.

Java/J2EE is the technology used (Servlets, JSP, JDBC)

HTML, JavaScript, and CSS are examples of web technologies.

Tomcat 7.0 is the web server.

Oracle 10g Express Edition database

JDK 1.6 is the software.

**Hardware requirements:**

          Hardware                     : Pentium based systems with a minimum of P4

           RAM                           : 1 GB (minimum)

# PROBLEM DEFINITION:

When appraising employees, it's important never to match their abilities and using it to form a judgment.

Each employee is presented in their unique way and thus has different strengths and weaknesses. once you try and compare or contrast their abilities, it means you may not get a good review because high performers will definitely make relatively low performers for particular tasks to appear below average, which on some occasions is rarely the case.

Of essence is to make sure that you simply appraise every worker by their performance against established standards and criteria, individually.

# SYSTEM ANALYSIS:

## EXISTING SYSTEM:

The existing Employee Performance vascular system consists of excel sheets accumulated from each employee. Each employee submits excel sheets to their superior who consolidates them into one sheet. The superior in question will need to confirm each sheet is error – free so perform tedious calculations to seek out ratings for workers supported completed duties and work targets reached. The manager will must use excel macros to convert the information into palatable graphs for review by upper level management.

In the existing system,

1. it's recommended that managers spend about an hour per employee writing performance appraisals.

2. The performance appraisal process has to be one among encouragement, positive reinforcement, and a celebration of a year’s worth of accomplishments..

3. it's difficult to stay biases out of the performance appraisal process.

4. within the existing system there are not any Avocation facilities.

**PROPOSED SYSTEM:**

within the proposed system, the aim of the project is to motivate improvement to the organization’s contribution. the worker Performance Portal aims to require over the duties of the present system, within the process it'll reduce the time required for performance appraisals, reduce user entry errors, enable global sharing of the reports and supply performance information in an appeasing graphical format. it'll also assist in other day – to – day HR requirements like hiring and maintenance of personnel information.

**Advantages of the Proposed System:**

1. A performance management process forces managers to debate performance issues with employees.

2. Performance appraisals are a time to celebrate all the wonderful things an employee does over the course of a year and will encourage staff.

3. Ensures data access authentication.

4. Dynamically updating the main points of the worker and manager and assistant manager.

5. we are able to include feedback of the worker.

6. Avocation is implemented within the proposed system.

Since the project is intended with ASP.NET with C# as face and SQL Server 2000 as face, it's easy to put in all told the systems wherever needed. it's more efficient, easy and user-friendly to grasp by almost everyone. Huge amount of information will be handled efficiently using SQL Server as face. Hence this project has good technical feasibility.

Since this project is developed using ASP.NET with C# and SQL Server which is more commonly available and even the value involved within the installation process isn't high.

Similarly it's easy to recruit persons for operating the software since the majority the people are responsive to ASP.NET with C# and SQL Server. whether or not we wish to coach the persons in these area the value involved in training is additionally very less. Hence this project has good economic feasibility.

The system once developed must be used efficiently. Otherwise there's no meaning for developing the system. For this a careful study of the prevailing system and its drawbacks are needed.

**PROJECT OVERVIEW:**

The main goal is to work out what style of system is acceptable for a given problem and the way to construct it. By staring at this system and its issues, the specifications for this technique may be determined. They speak about the new system that may be installed and what they hope to urge out of it.

The Employee Performance Portal in itself will comprise of modules which will facilitate interaction between employees and managers. the workers are able to enter the requisite details like the on – job duties completed and performance measures that are worked upon and other such information. Further this data will be utilized by the varied managers to accurately analyse and rate employee performance. The portal will contain information like completed duties and work targets reached on an hourly, daily, weekly, fortnightly, monthly or annual basis. there'll even be a provision for managers to produce valuable feedback or areas of improvement to the worker. All users of the portal are going to be informed about employees who achieve stellar ratings and this culture of praise for labor will motivate more employees to strive for positive ratings.

**PROJECT MODULES:**

1. Admin Module

2. Appraise Module

3. Appraiser Module

4. Reviewer Module

**ADMIN:**

This will enable all the executive tasks of the portal like management of records and look at total statistics. the protection functions of the portal like new logins also will be managed from here.

⮚ **ID**

**⮚ Password**

**⮚ Email-id**

⮚ **Phone number**

**⮚ Gender**

**⮚ experience**

**⮚ salary**

**APPRISE:**

This module treats every employee as an appraisee and allows them to submits appraisals to his manager which contains task details like Task ID, Task Desc, Task Status, Weightage of the task, self rating for the all the task he has finished during this half-yearly period. He can view the appraisals what he has submitted. He can view the rating given by the appraiser for the task in his appraisals.

⮚ **employee ID**

**⮚ Password**

**⮚ Task ID**

**⮚ Feedback ID**

**APPRISER:**

Appraiser is nothing but Appraisee’s reporting manager. Main function is to judge appraisal and rate appraisee’s performance. He can evaluate the rating given by the appraisee and he can give the consolidate rank to the appraisee. After that, he can submit it to the reviewer. He can view the reviewers requests forwarded from reviewers and re-rate the suitable employee appraisal. He can view reports like My Projects, View CR, print appraisal and look at My C Rank.

⮚ **Manager ID**

**⮚ password**

**⮚ name**

**⮚ phone number**

**⮚ email id**

**⮚ experience**

**⮚ salary**

**REVIEWER:**

Reviewer is nothing but Appraiser’s reporting manager to review the appraisal. He can view the review requests from different appraisees and forward those requests to appraisers. He can collect the report from the appraiser’s supported that report he can give the rating to the worker.

**Register Managers:**

Administrator can ready to register new Manager for every department to process the citizen or guest requests. The manager can send status information within the half administrator.

Employee:

In an employee-initiated review system, employees are informed that they'll provoke a review from their manager. this kind of on-demand appraisal isn't meant to switch a traditional review process. Rather, it will be accustomed promote an attitude of self-management among workers. Adherents to the present form of review process contend that it promotes regular communication between staff and managers. Detractors, though, note that it's keen about the employees' initiative, making it a but ideal alternative for a few workers with quiet, retiring personalities or confidence issues.

**DEFINITION ACRONYMS ABBREVIATIONS:**

**ABBREVIATIONS :**

HTTP : Hypertext Transfer Protocol

HTML : Hyper Text Markup Language

URL : Uniform Resource Locator

SRS : Software Requirement Specification

WWW : World Wide Web

CTO : Central time officer

JDBC : Java database connectivity

**SYSTEM REQUIREMENTS**

## Hardware Requirements

* Hardware : Pentium based systems with a minimum of P4.
* RAM : 256MB (minimum)

## Software Requirements

* Operating System : Windows XP Service Pack2.
* Technology : Java/J2EE (Servlets, JSP, JDBC).
* Web technology : HTML, JavaScript, CSS.
* Web Server : Tomcat 6.0
* Database : Oracle 10g Express Edition.
* Software’s : JDK 1.6

# TECHNOLOGIES:

**Languages Used:**

We chose Java as the programming language for this project.

* **JAVA**

**HTML:**

The most popular markup language for web pages is HTML, which stands for Hypertext Markup Language. It enables the development of structured documents by indicating structural semantics for text elements such as headings, paragraphs, lists, links, quotes, and other elements. It allows for the embedding of images and objects which can be used to construct interactive forms. It's composed in HTML elements, which are made up of

**HTML TAGS:**

**Basic HTML Tags:**

<!----> it specifies comments

<A>.......</A>Creates hypertext link

<B>......</B>Formats text as bold

<BIG>.....</BIG>Formats text in large font

<BODY>.......</BODY>Contains all the tags and text in the HTML documents

<CENTER>.....</CENTER>Aligns text center

<DD>....</DD>Creates definition of a tem

<DL>.....</DL>Creates definition list

<FONT>.....</FONT>Formats text with a particular font

<FORM>.....</FORM>encloses a fill-out form

<FRAME>...........</FRAME>Defines a particular frame in a set of frames

<H#>............</H#>Creates headings of different levels

<HEAD>...........</HEAD>Contains tags that specify information about a document

<HR>..........</HR> Creates a horizontal rule

<HTML>..........</HTML>Contains all other HTML tags

**Advantages:**

• Since HTML documents are short, they are simple to submit over the internet.

• It's compact because it's devoid of formatted details.

• HTML can run on different platforms**.**

**JDBC:**

JDBC is a Java programming language API that describes how a client can interact with a database.

It contains methods for querying and updating database data. JDBC is designed for use with relational databases.

JDBC is a database connectivity protocol.

The Java2 Platform, Standard Edition, version 1.1(J2SE), along with a reference implementation, were first implemented in the Java2 Platform, Standard Edition, version 1.1(J2SE).

**Functionality:**

Multiple implementations of JDBC will coexist and be used by the same programme. The API offers a framework for dynamically loading and registering the appropriate Java packages with the JDBC Driver Manager. The Driver Manager is used to create JDBC connections as a link factory.

JDBC connections allow you to write and execute statements. These may be update statements like CREATE, INSERT, UPDATE, and DELETE in SQL, or query statements like SELECT. Additionally, a JDBC link can be used to call stored procedures. JDBC uses one of the following classes to describe statements:

* **Statement** – the statement is sent to the database server each and every time.
* **Prepared Statement**- the statement is cached and then the execution path is pre determined on the database server allowing it to be executed multiple times in an efficient manner.
* **Callable Statement**-used for executing stored procedures on the database.

Change statements like INSERT, UPDATE, and DELETE produce an update count that shows how many rows in the database were affected. There is no other knowledge returned by these claims.

A JDBC row result set is returned by query statements. The result set is walked over using the row result set. Individual columns in a row can be retrieved by column number or by name. The result set may have any number of rows. The metadata in the row result set defines the names and styles of the columns.

**JDBC Drivers:**

JDBC drivers are client-side adapters (that is, they are installed on the client machine rather than the server) that convert Java programme requests into a protocol that the database management system understands.

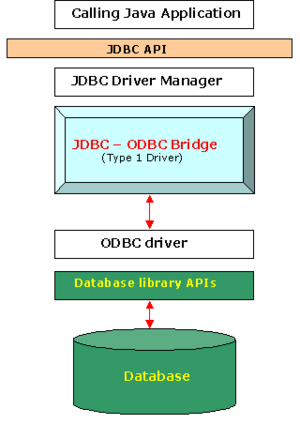
JDBC drivers fall under four categories:

* **TYPE-1**
* **TYPE-2**
* **TYPE-3**
* **TYPE-4**

**Type 1 driver- JDBC-ODBC Bridge**

The JDBC-ODBC bridge is a database driver that connects to the database using the ODBC driver. JDBC method calls are converted to ODBC function calls by the driver.

Other installation dependencies are triggered by this driver; for example, ODBC must be installed on the device where the driver is installed, and the database must accept an ODBC driver. If a pure-Java driver is available, it is recommended that you use it instead.

****

**Functions:**

* Translates a JDBC query into a corresponding ODBC query, which is then done by the ODBC driver.
* A JDBC-ODBC Bridge driver is available from Sun.
* Sun.jdbc.odbc.JdbcDriver is a Sun.jdbc.odbc.JdbcDriver.
* This driver is closed source and written in native code rather than Java.
* Database -> Client -> JDBC Driver -> ODBC Driver

There is some overhead associated with the conversion from JDBC to ODBC.

**Advantages:**

• Virtually every database that has an ODBC driver installed can be accessed.

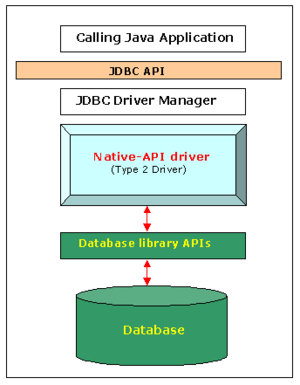
• Installing a type 1 driver is easy.

**Disadvantages:**

* The driver is platform-dependent since it relies on ODBC, which in turn relies on native libraries of the operating system on which the JVM is running.
* This technology isn't designed for high-volume transactions.
* Because of the binding between the driver and the platform, this driver is not portable.

**Type 2 Driver – Native – API Driver specification:**

The Native –API driver, also known as the JDBC form 2 driver, is a database driver that uses the database's client-side libraries. The driver translates JDBC method calls to database API native calls.

****

Since it interfaces with non-Java code that allows the final database calls, the type 2 driver is not completely written in Java. The driver is designed to work with a specific operating system. The Type 4 driver, which is a full-Java implementation, is favoured for platform interoperability over this driver.

The type 2 driver, on the other hand, has more features and better performance than the type 1 driver.

**Advantages:**

* Better performance than Type 1 Driver(JDBC-ODBC Bridge).
* Provides Fastest performance than all 3 drivers as it calls native APIs ( MySQL, Oracle etc.)

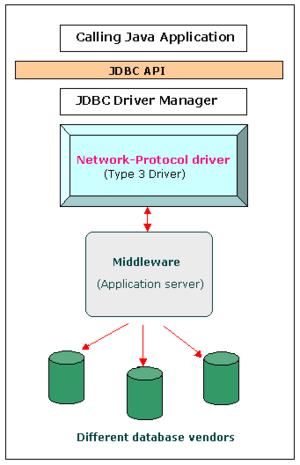
**Disadvantages:**

* The vendor client library has to be installed on the client machine.
* Cannot be used in web-based application due to the client side software needed.
* Not all databases have client side library.
* This server is platform dependent.This driver supports all java applications except Applets.

**Type 3 Driver – Network – Protocol Driver**

The Pure Java Driver for Database Middleware, also known as the JDBC type 3 driver, is a database driver implementation that uses a middle tier between the calling software and the database.

JDBC calls are converted directly or indirectly into the vendor-specific database protocol by the middle-tier (application server).

****

The protocol conversion logic is located in the middle-tier, not at the client, as it is with the type 4 driver. The type 3 driver is written entirely in Java, much like the type 4 driver. Multiple databases can be served by the same engine. It is determined by the number of databases that the middleware will accommodate. Because of the platform-related differences, the type 3 driver is platform-independent.

**Functions:**

• Can communicate with several databases – not vendor specific. • Follows a three-tier communication strategy.

• The Java-based JDBC Client driver interacts with a middleware-net-server using a database-independent protocol, and the net server then converts the request into database commands specific to that database.

• As a result, the correspondence between the client driver and the middleware is database independent.

**Advantages:**

* Since the communication between client and the middleware server is database independent, need not be changed for a new database.
* The Middleware Server (which can be a full fledged J2EE Application server) can provide typical middleware services like caching(connections, query results, and so on), load balancing, logging, auditing etc. Eg. For the above include jdbc driver features in Weblogic.
* At client side a single driver can handle any database. (It works provided the middleware supports that database)
* Can be used in internet since there is no client side software needed.

**Disadvantages:**

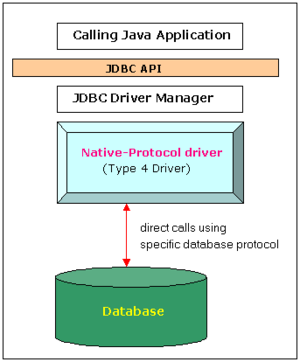
• Database-specific coding must be performed in the middle tier. A time-bottleneck could occur if an extra layer is added. However, this is usually resolved by delivering effective middleware services.

**Type 4 Driver - Native-Protocol Driver:**

The Direct to Database Pure Java Driver, also known as the JDBC form 4 drivers, is a database driver that translates JDBC calls directly into the vendor-specific database protocol.

Since the type 4 driver is written entirely in Java, it is platform agnostic. It is built in the client's Java Virtual Machine. It outperforms type 1 and 2 drivers in terms of efficiency.

Since the database protocol is vendor-specific, different drivers, which are normally provided by the vendor, must be used to link to different databases.

****

**Functions:**

* Type 4 drivers are wholly written in Java and communicate with a vendor's database directly, typically via socket connections.
* There is no need for translation or middleware layers, which improves performance.
* The driver transforms JDBC calls to the vendor-specific database protocol, allowing client applications to interface with the database server directly.
* To achieve platform freedom, the program was written entirely in Java.
* For example, oracle.jdbc.driver is a widely used Oracle thin driver.
* OracleDriver uses the jdbc:oracle:thin URL format to connect to Oracle.

**Advantages:**

These drivers do not need a middleware layer to service requests, nor do they translate requests into an intermediary format (such as ODBC).

●

As a result, the efficiency could be significantly improved.

The JVM will handle any element of the application-to-database relation, making debugging much easier.

**Disadvantages:**

Each database needs its own driver on the client side.

**STAGES IN A JDBC PROGRAM:**

**REGISTERING THE DRIVER:**

A database driver is software containing the classes and interfaces written in line with the JDBC API. Since there are several drivers available within the market, we must always first declare the driving force which goes to be used for communication with the database server during a java program.

**CONNECTING TO A DATABASE:**

during this stage we establish the reference to a particular database through the driving force which is already registered within the previous step.

**PREPARING SQL STATEMENT:**

we must always create SQL statements in our java program using anyone of the interfaces like Statement, Prepared Statement, Callable Statement which are available in java.sql package.

**EXECUTING THE SQL STATEMENTS ON THE DATABASE:**

For this purpose, we are able to use executeUpdate(), executeQuery(), methods of statement Interface.

**RETRIEVING THE RESULTS:**

The Results obtained by executing the SQL statements can be stored in an object with the help of interfaces like Result Set.

**CLOSING THE CONNECTION:**

We should close the connection between the Java program and the database using close() method of connection Interface.

**DBMS Terminology:**

**Database**: A database is a collection of tables, with related data.

**Table**: A table is a matrix with data. A table in a database looks like a simple spreadsheet.

**Column:** One column (data element) contains data of one and the same kind, for example the column postcode.

**Row**: A row (tuple, entry or record) is a group of related data, for example the data of one subscriptionn.

**Primary Key**: A primary key is unique. A key value cannot occur twice in one table. With a key you can find at most one row.

**Foreign Key**: A foreign key is the linking pin between two tables.

**Composite Key**: A composite key is a key that consist of multiple columns,

because one column is not sufficiently unique.

**Unique Key**: A unique key is a key that allows null values in tables. Each table can have multiple unique keys.

# SYSTEM DESIGN

## Use Case Diagrams:

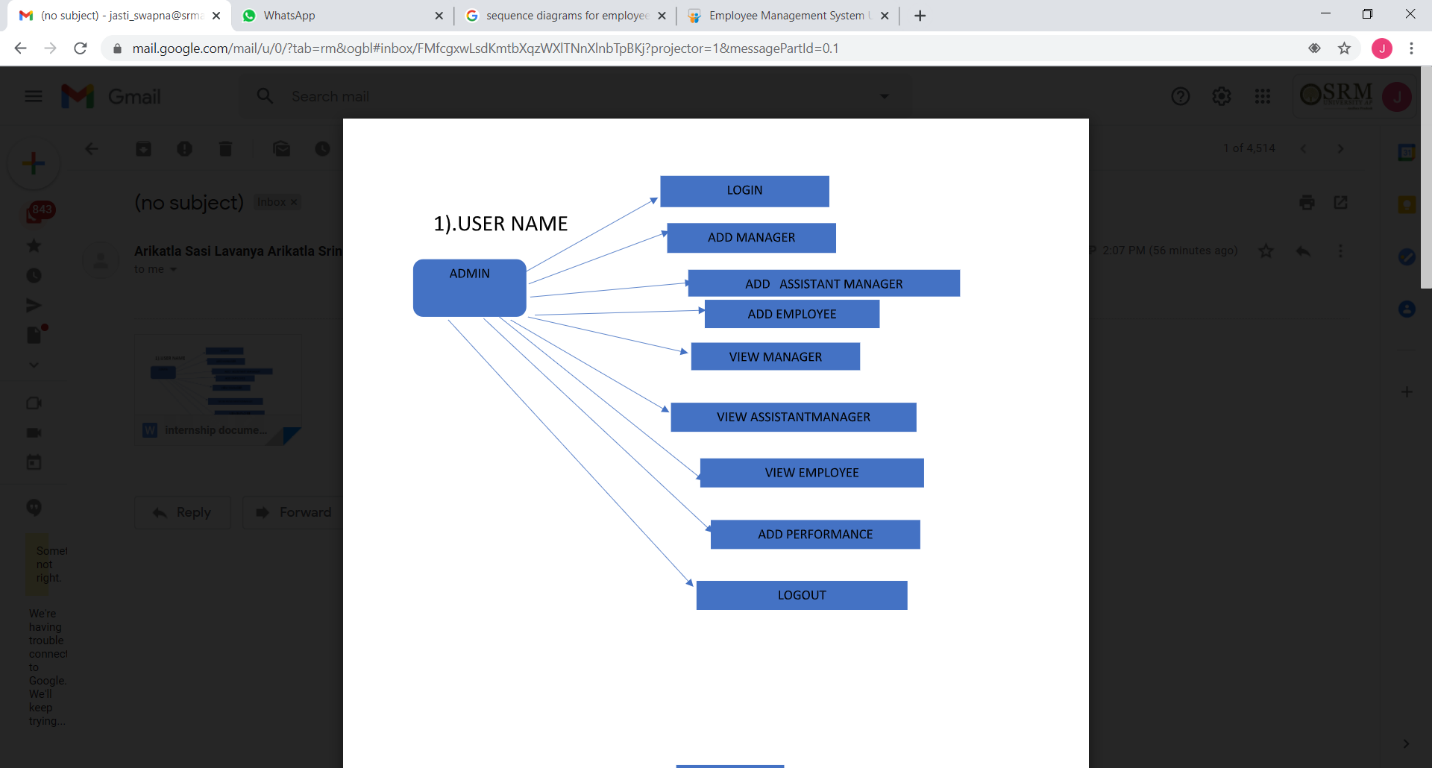
|  |  |
| --- | --- |
| http://www.visualcase.com/kbase/usecaseobject.jpg | Each **use case** on the diagram represents a single task that the system needs to carry out.  *Buy a Product*, *Add Client*, *Make Purchase* and *Validate Order Information* are all examples of use cases.  Some use cases may include or extend a task represented by another use case.  For example, in order to make a purchase, the order information will need to be validated. |

Actor

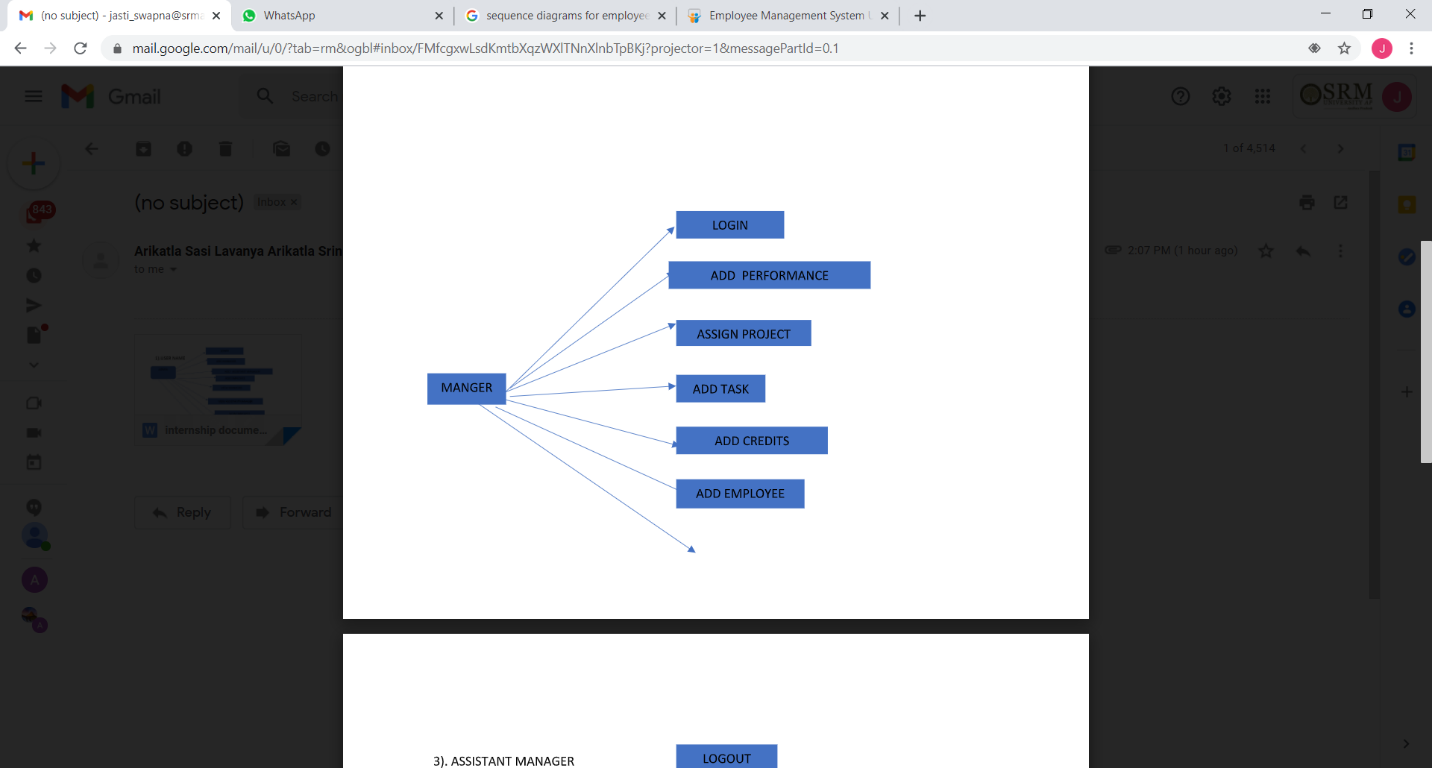
|  |  |
| --- | --- |
|  | An **actor** is anything outside the system that interacts with the system to complete a task.  It could be a user or another system.  The actor "uses" the use case to complete a task.  *System Administrator*, *Credit Authentication System, Accounting System* and *Web Client* are all examples of actors.  Often, it is useful to look at the set of use cases that an actor has access to -- this defines the actor's overall role in the system. |

**Over all Use Cases**

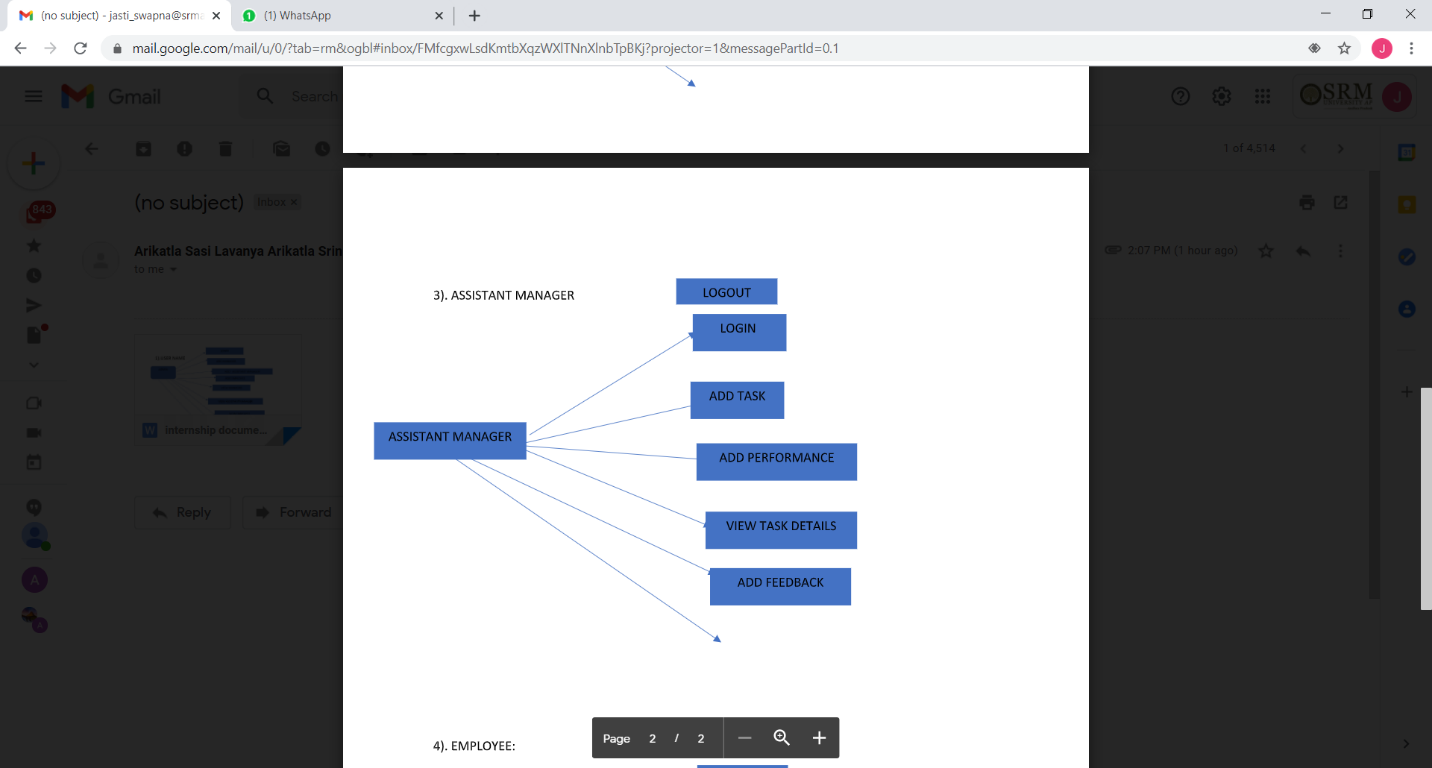
**ADMIN:**



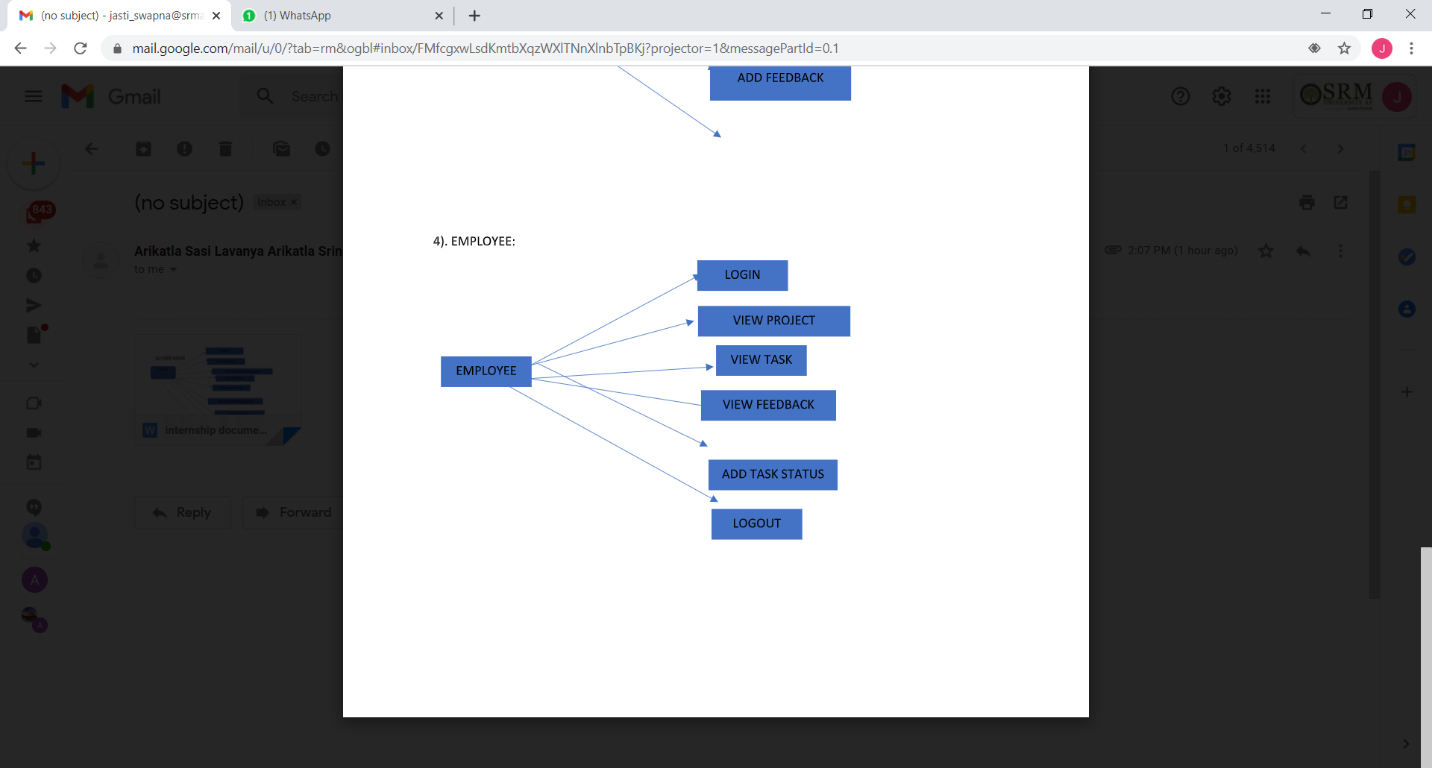
MANAGER:



ASSISTANT MANAGER:



EMPLOYEE:



**Use Case Diagrams**

The actors who have been identified in the system are as follows:

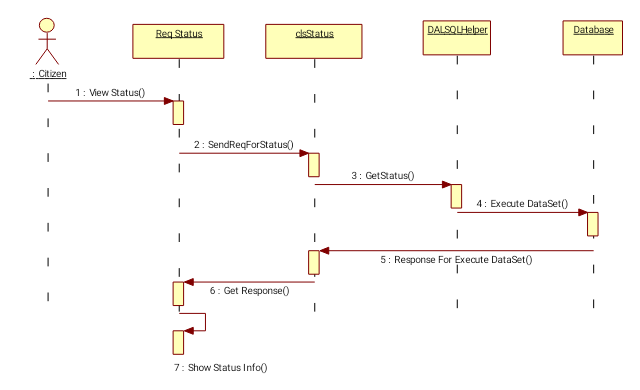
1. **Investigating officer**
2. **Administrator**
3. **Writer**

**Investigating officer:** He is the actor who can practically work upon the existing data in the police station only for view purpose.

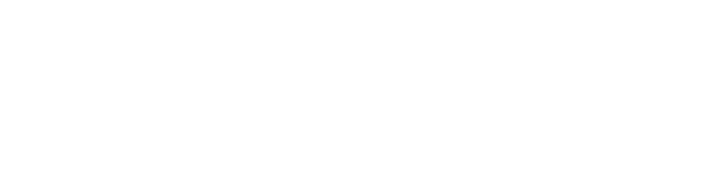
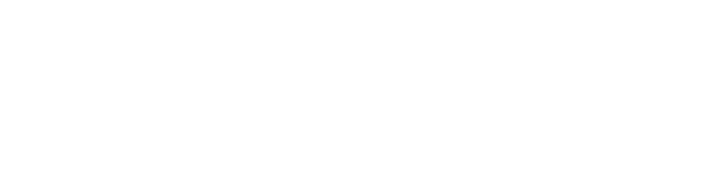
**Administrator:** He is the actor who has the full-length potentiality and privilege to carry out transactions upon the system. He is authorized to maintain consistency within the information.

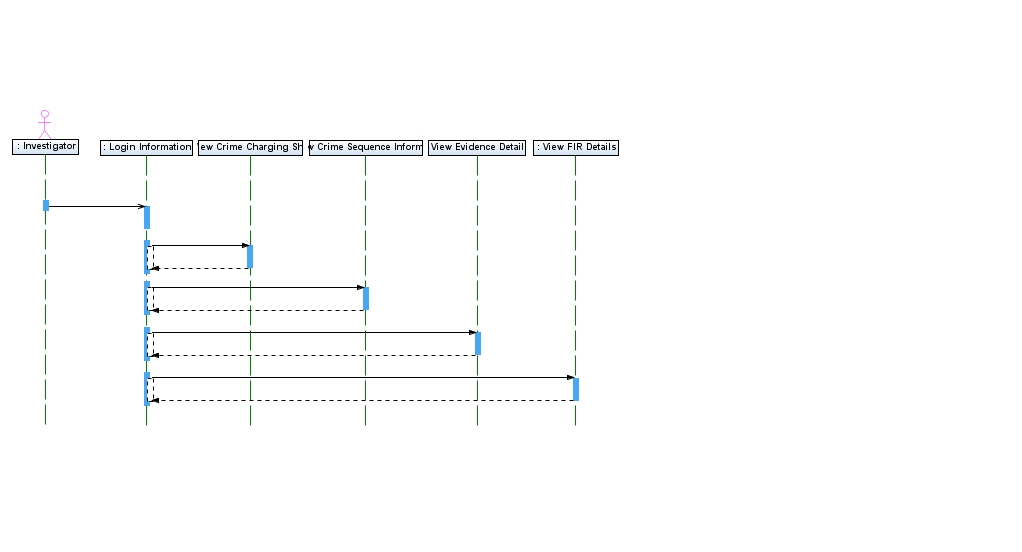
**Writer:** He is the actor who can enter all the details of the crime or evidence. Once entered cannot be edited. Only the administrator can edit or delete the record from the database.

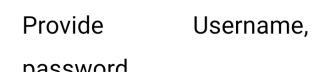
**Citizen Search for Request Status**

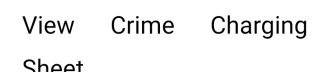


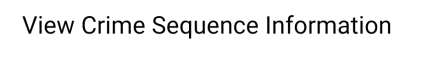
**sequence diagrams**

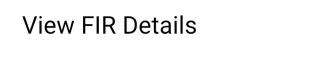
**Investigator:**



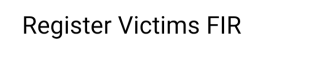
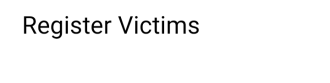
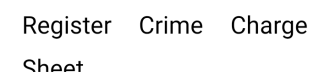
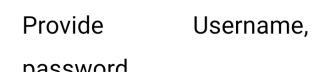
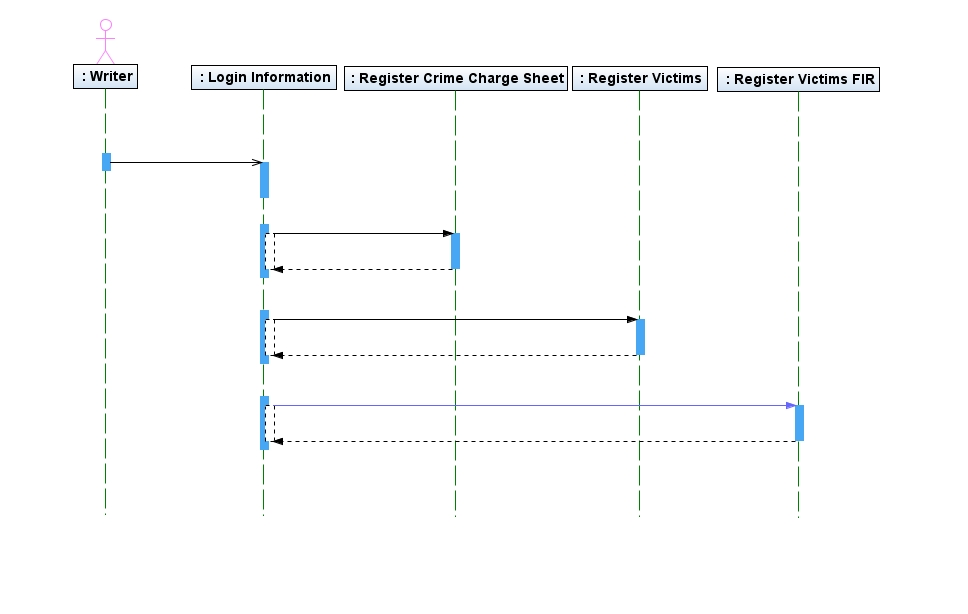








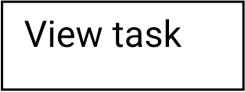
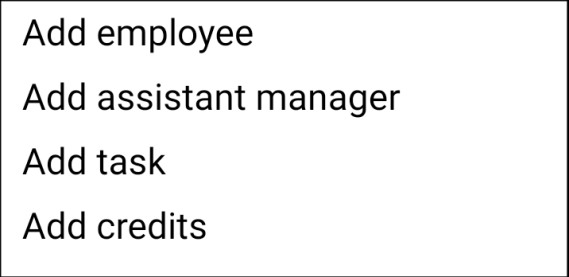
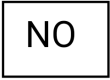
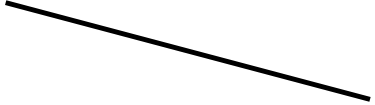
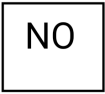
**Writer:**



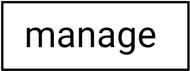
## Activity Diagrams

**admin:**

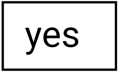
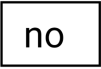
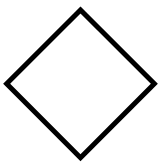




MANAGER:



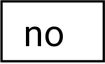


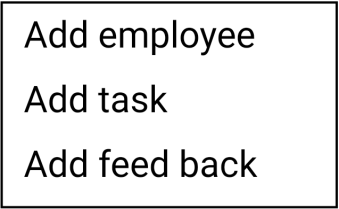


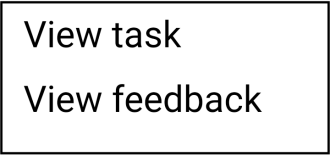






















ASSISTANTMANAGER:

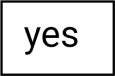








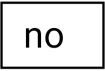
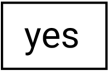




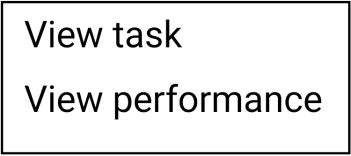
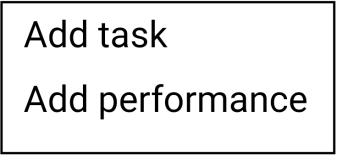














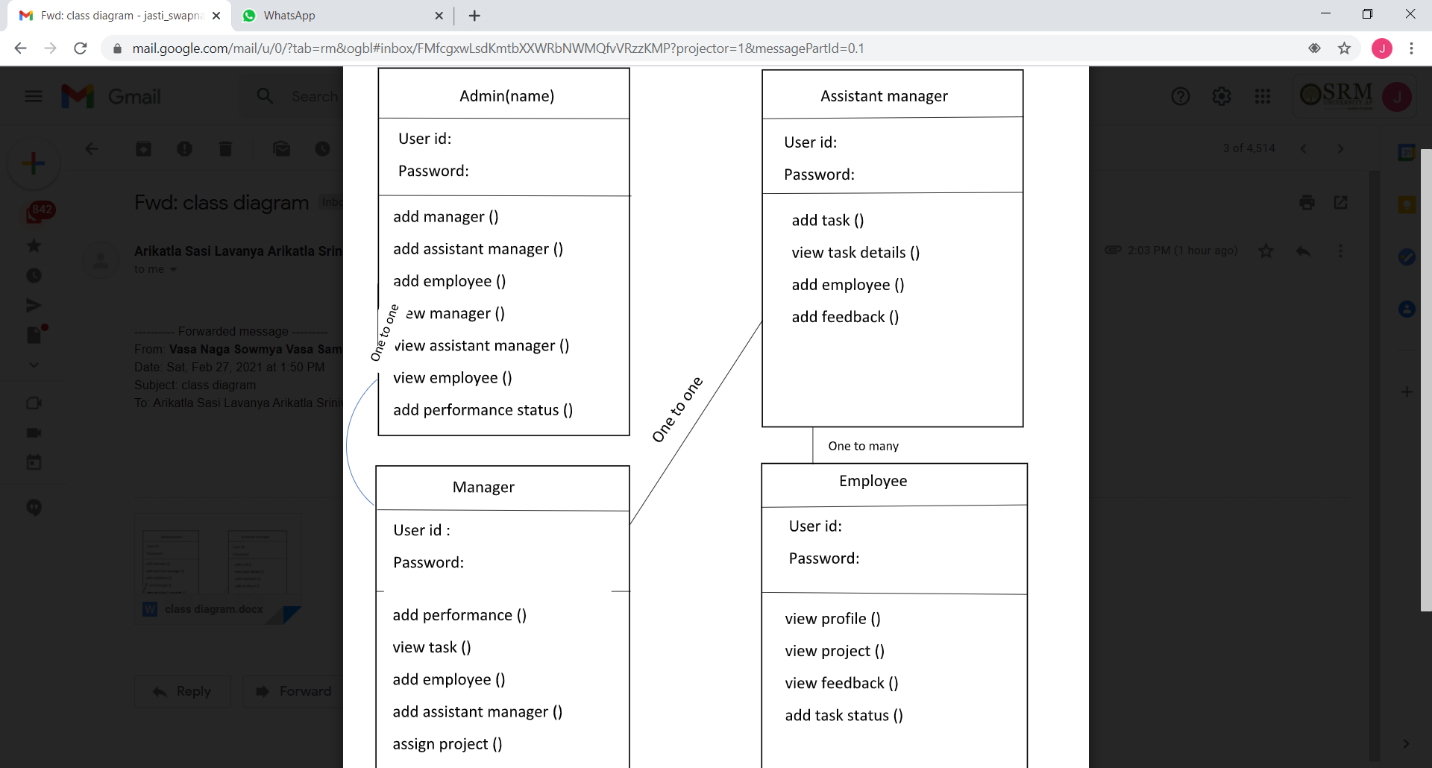






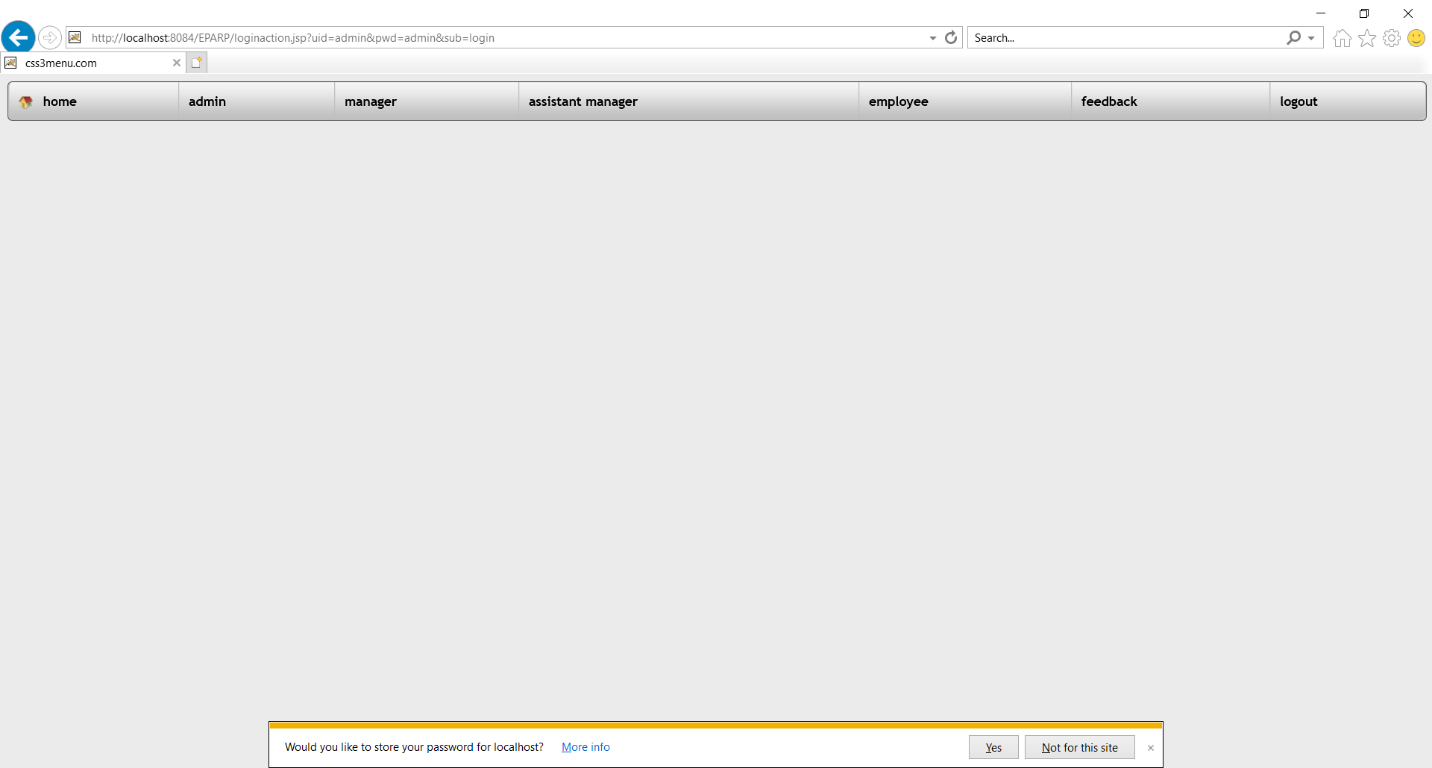


CLASS DIAGRAMS:

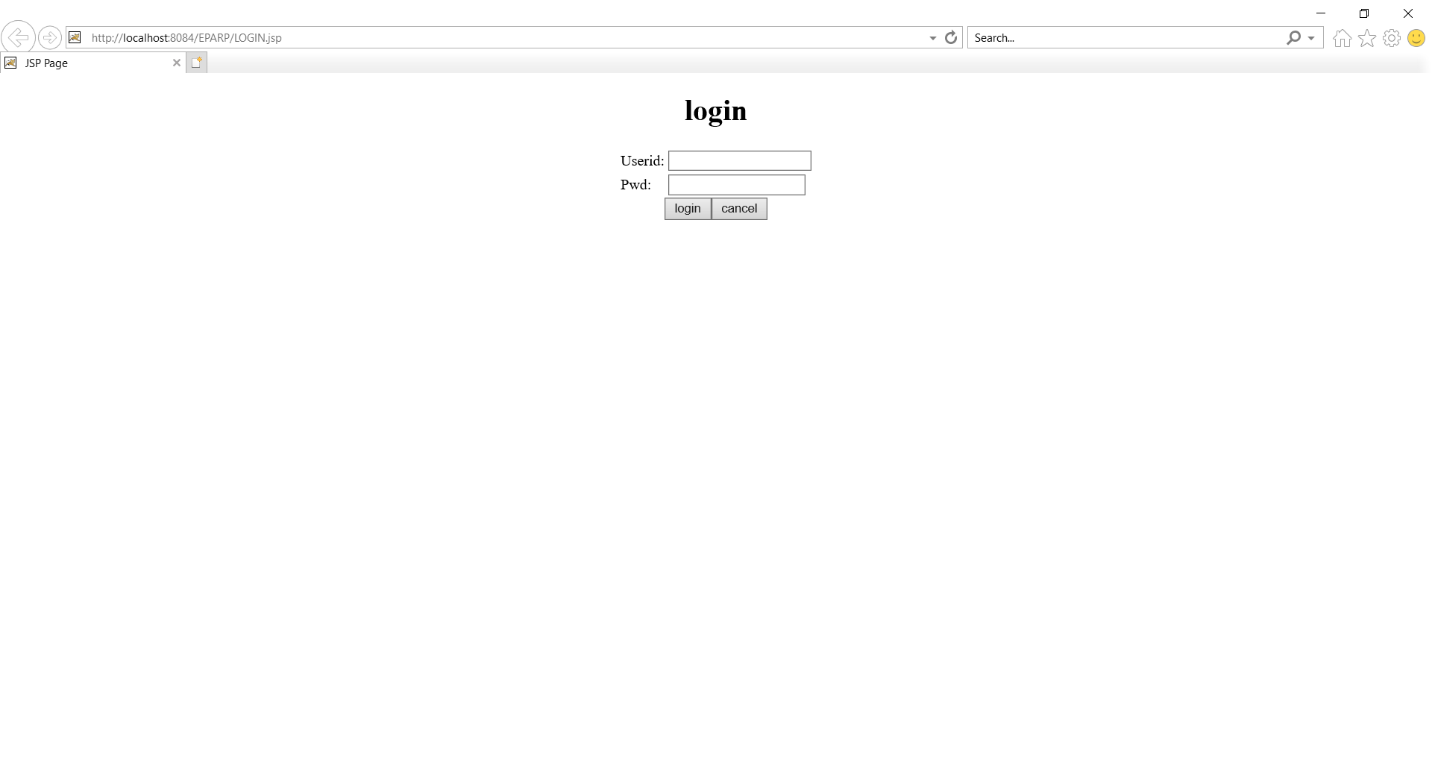


# SCREEN LAYOUT

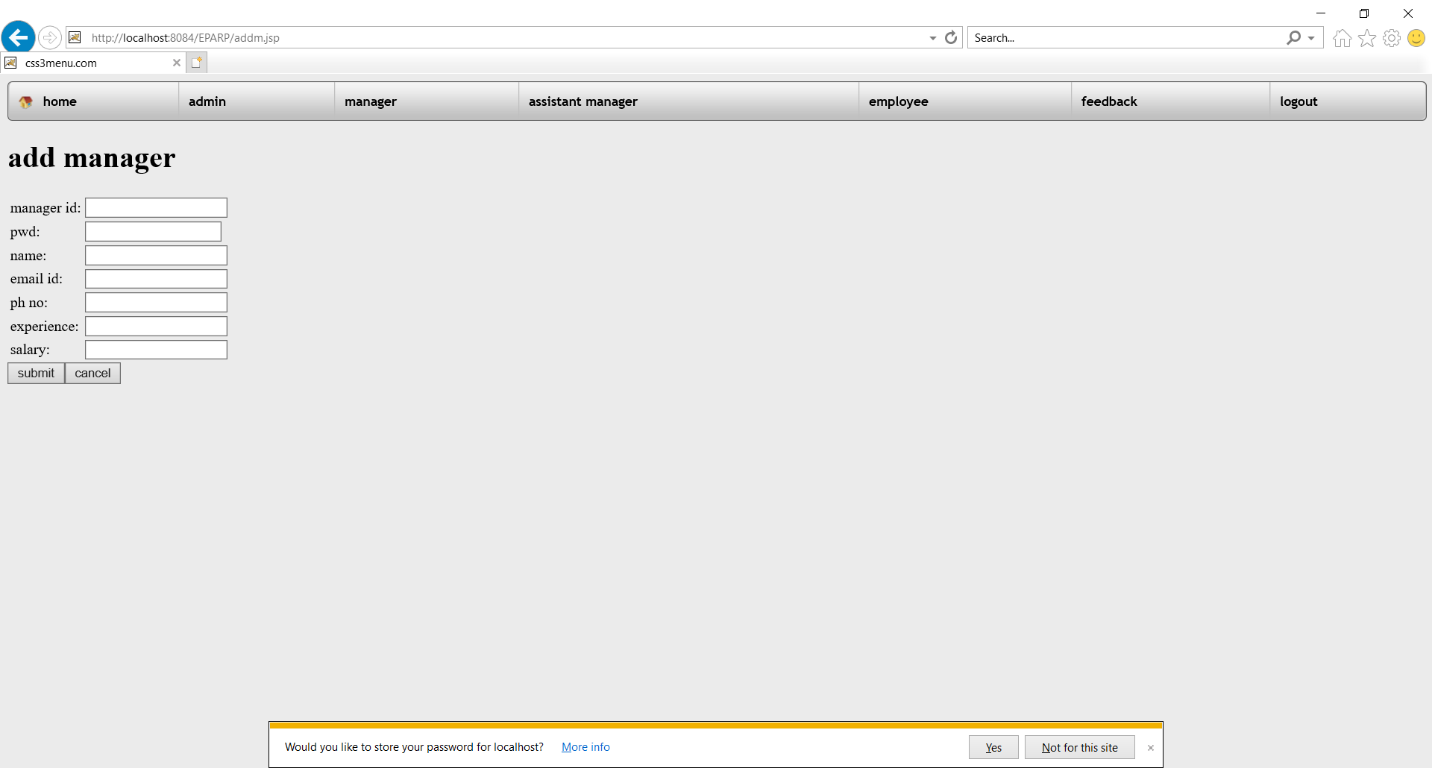
MAIN PAGE



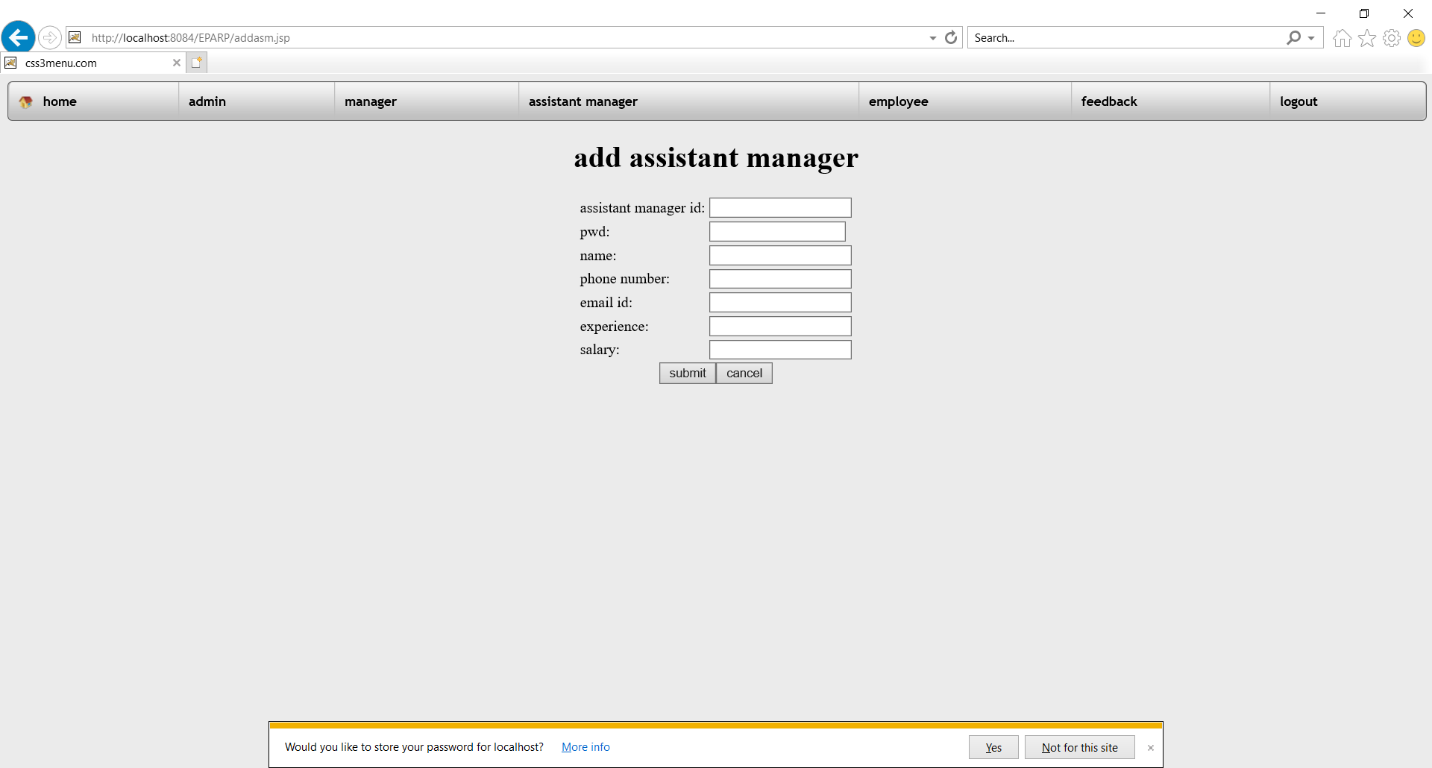
LOGIN PAGE:



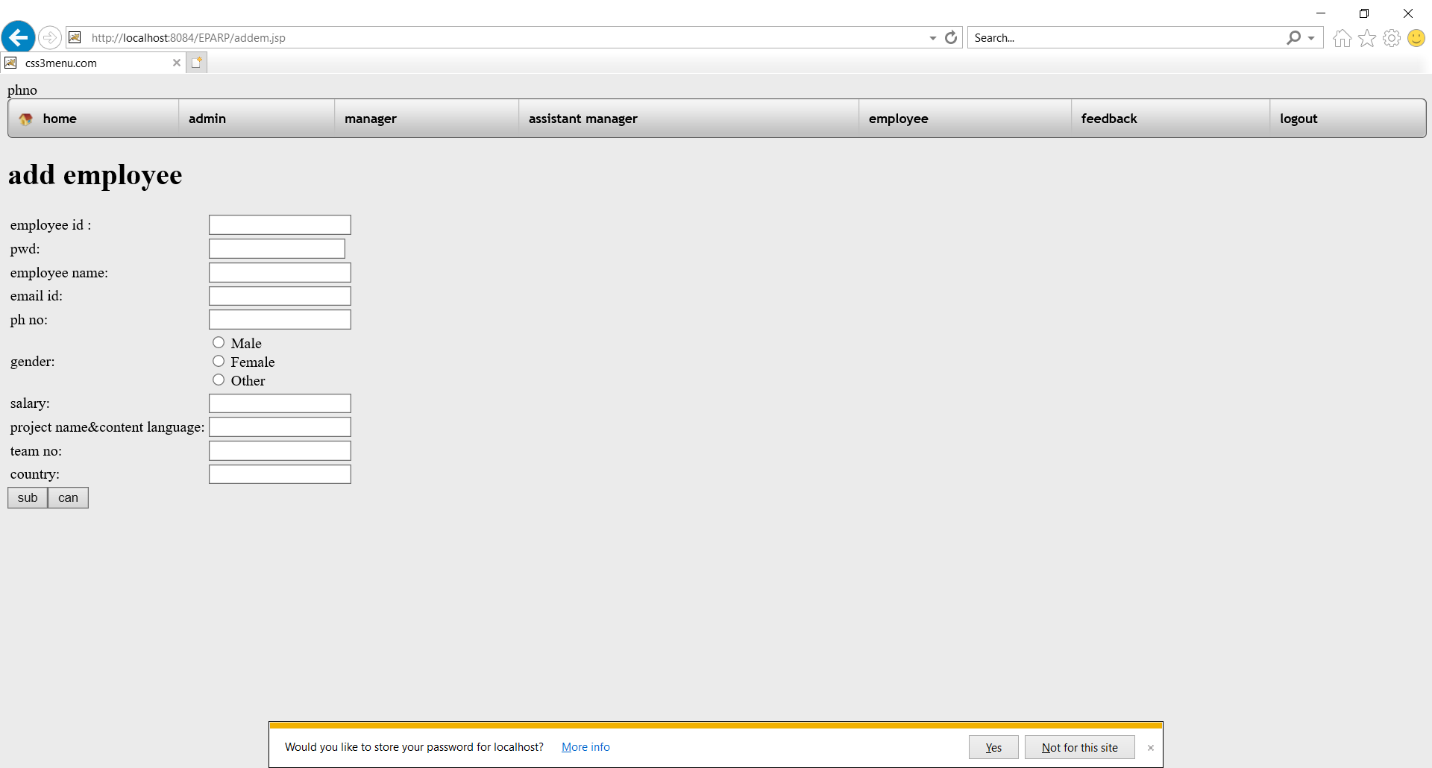
ADD MANAGER PAGE:



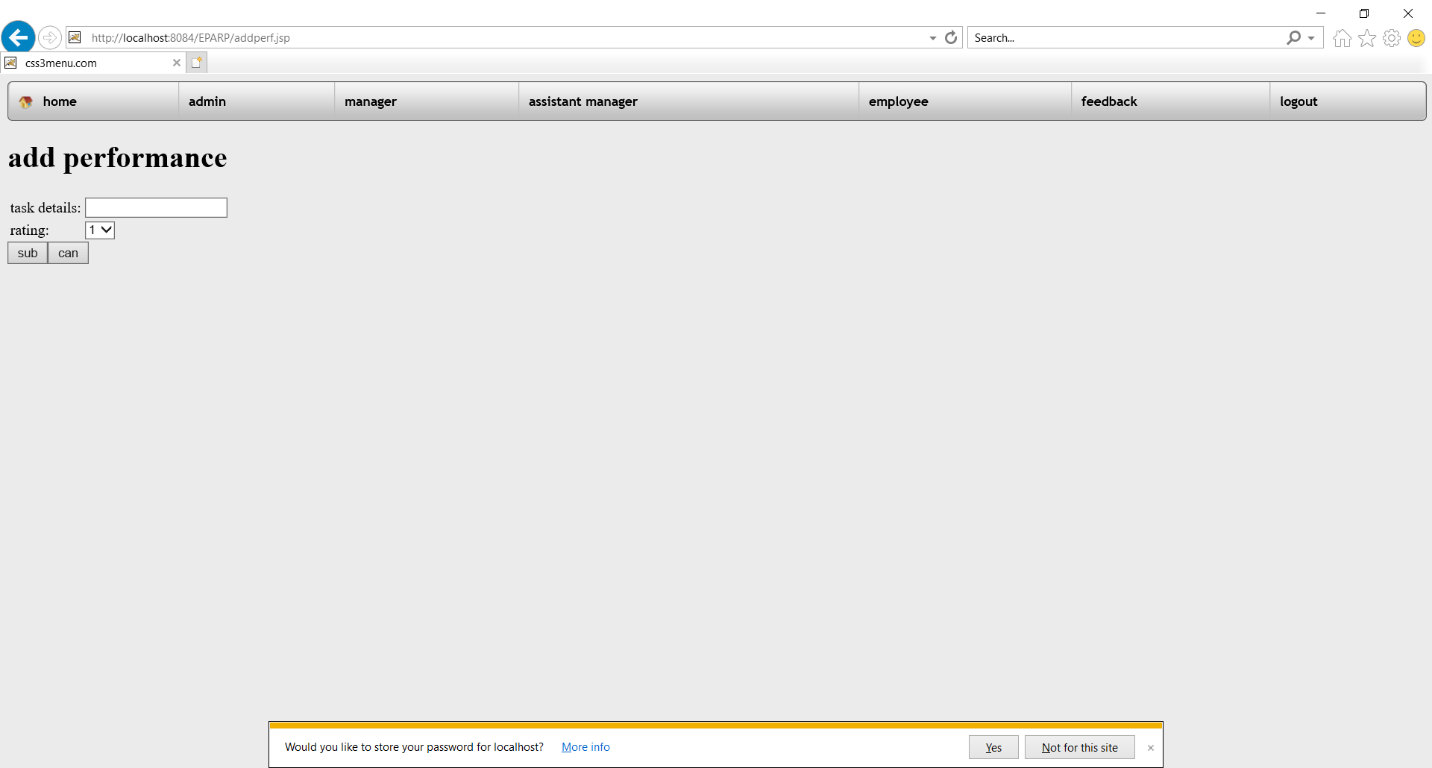
ADD ASSISTANT MANAGER PAGE:



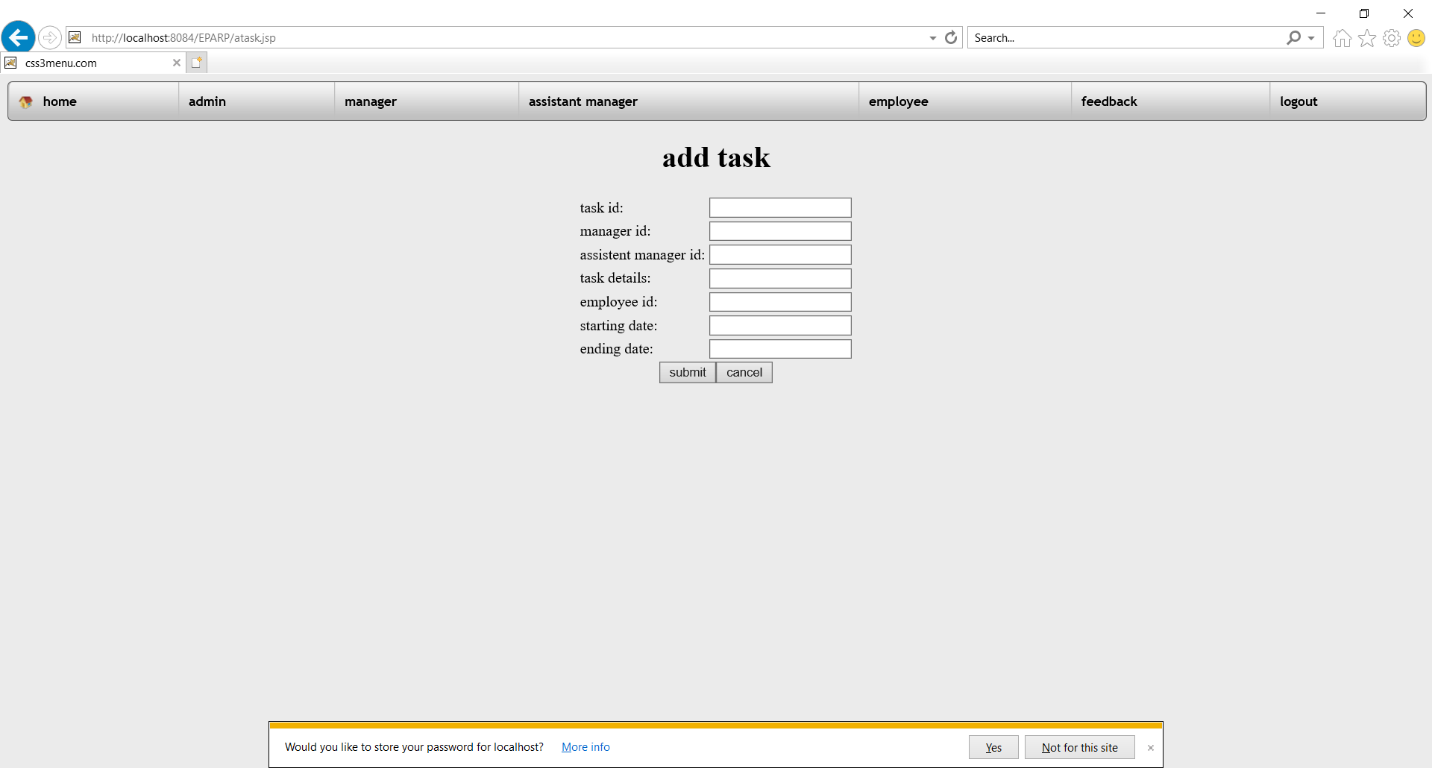
ADD EMPLOYEE PAGE:



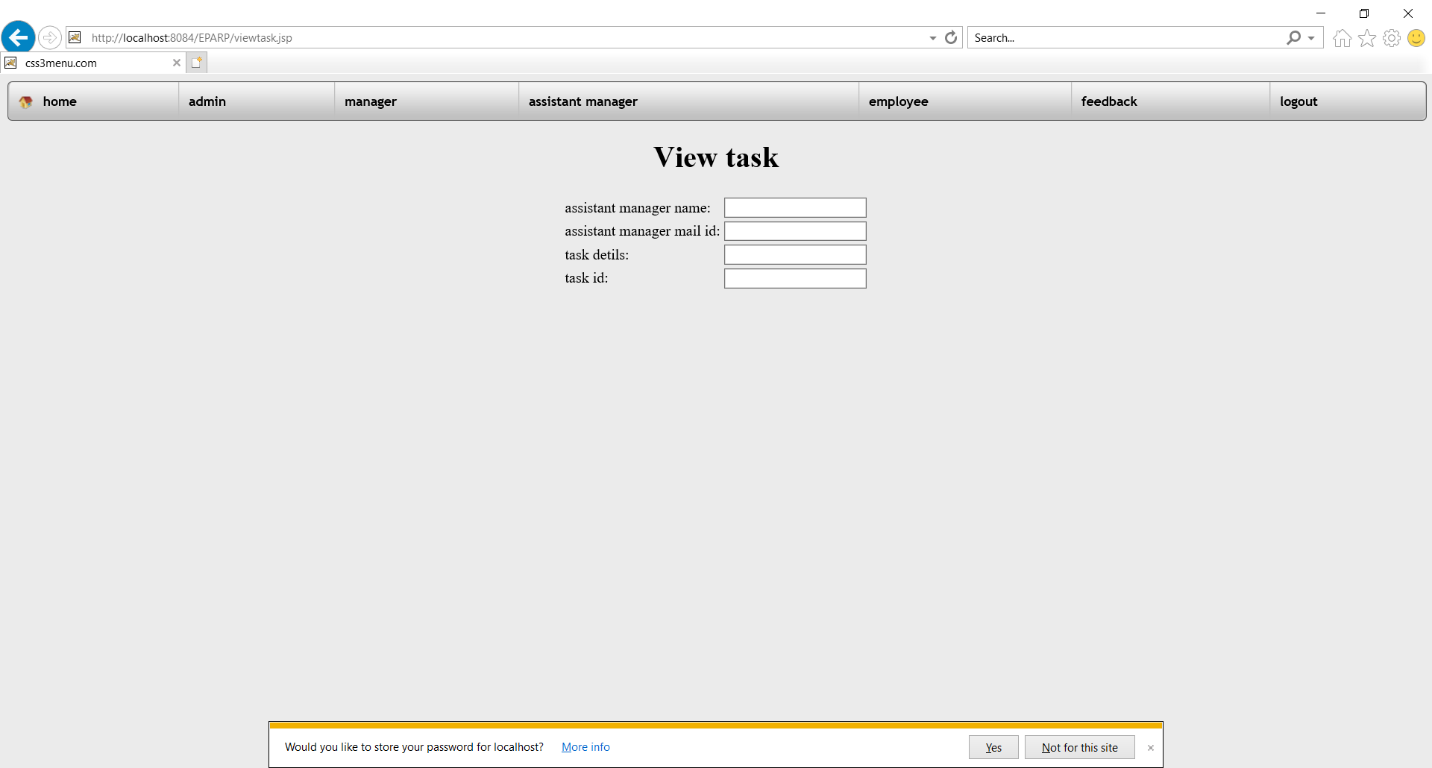
ADD PERFORMANCE:



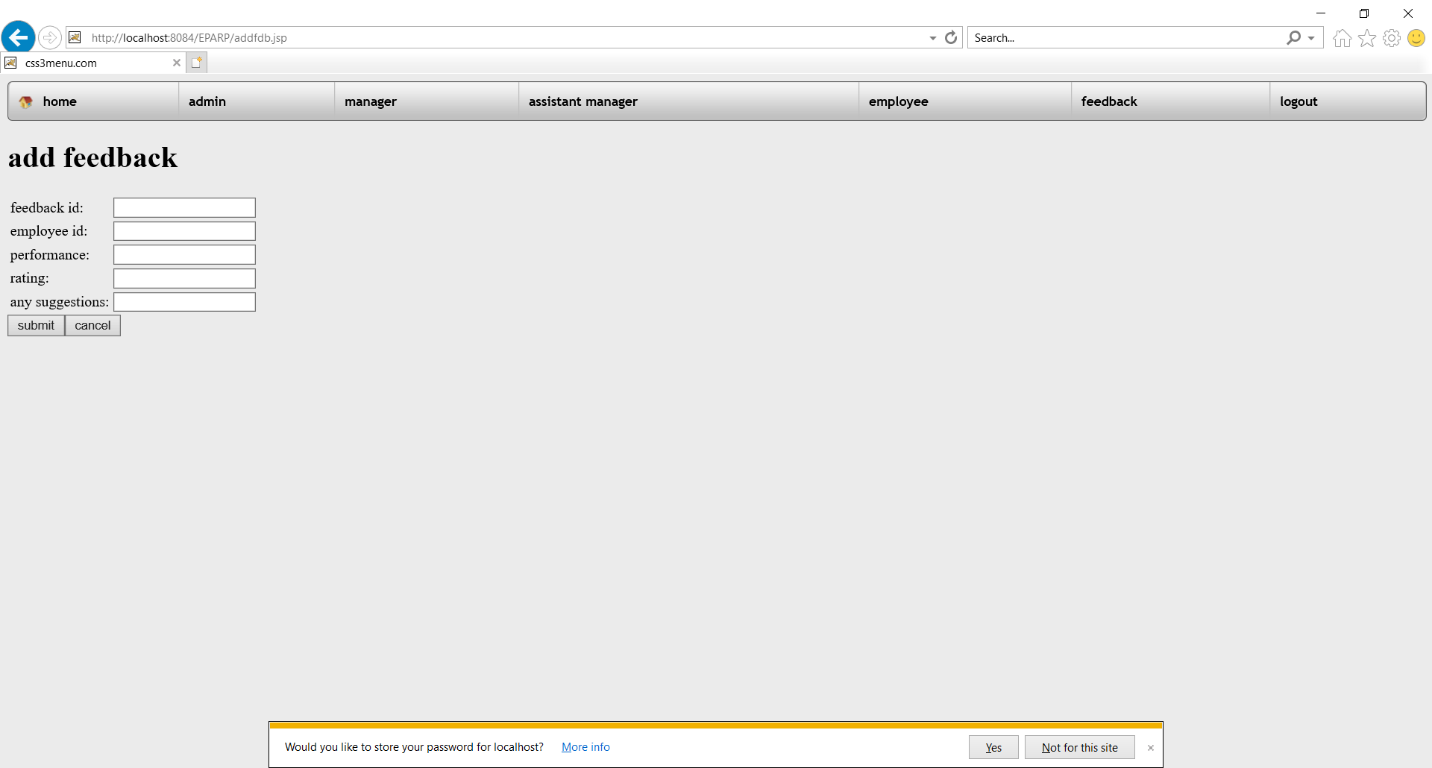
ADD TASK:



VIEW TASK:

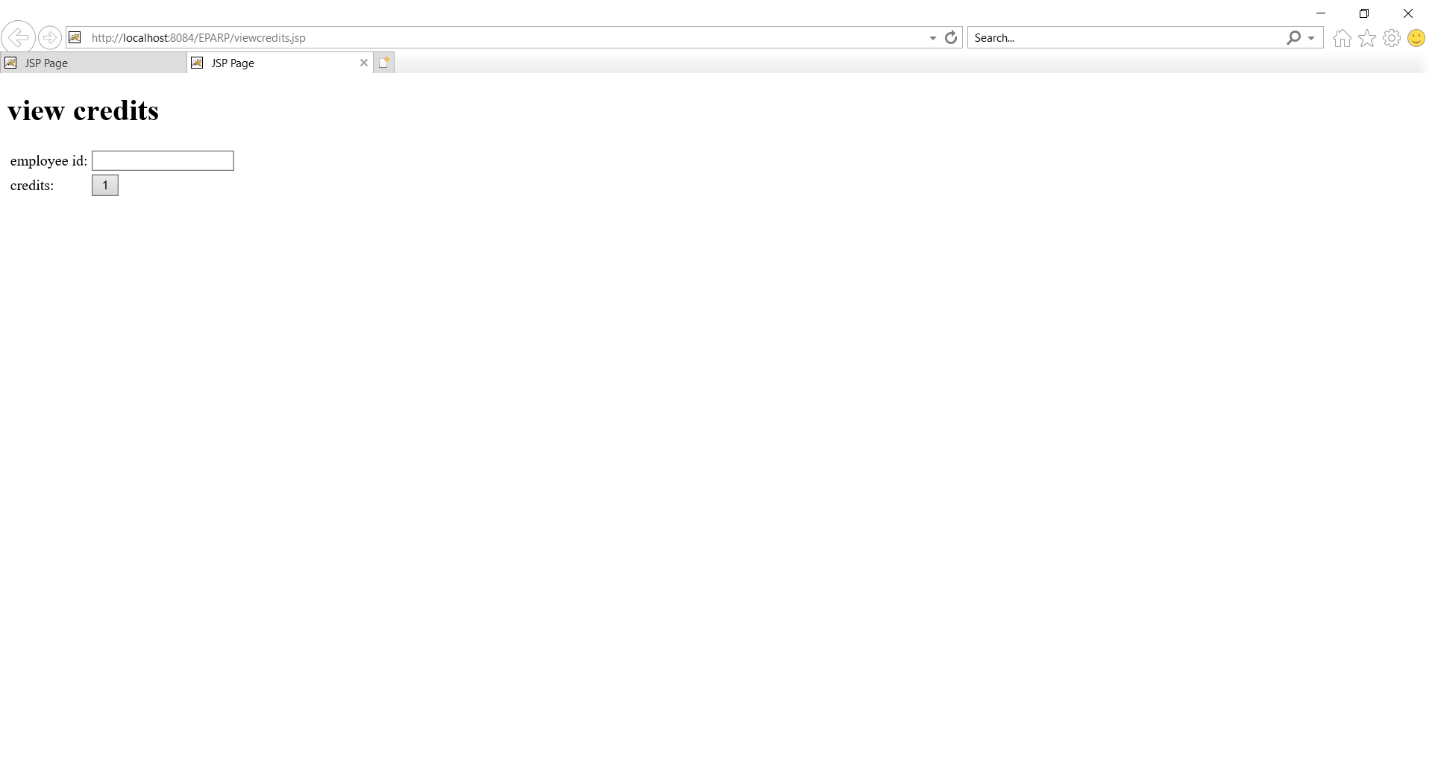


ADD FEEDBACK:



**VIEW FEEDBACK:**

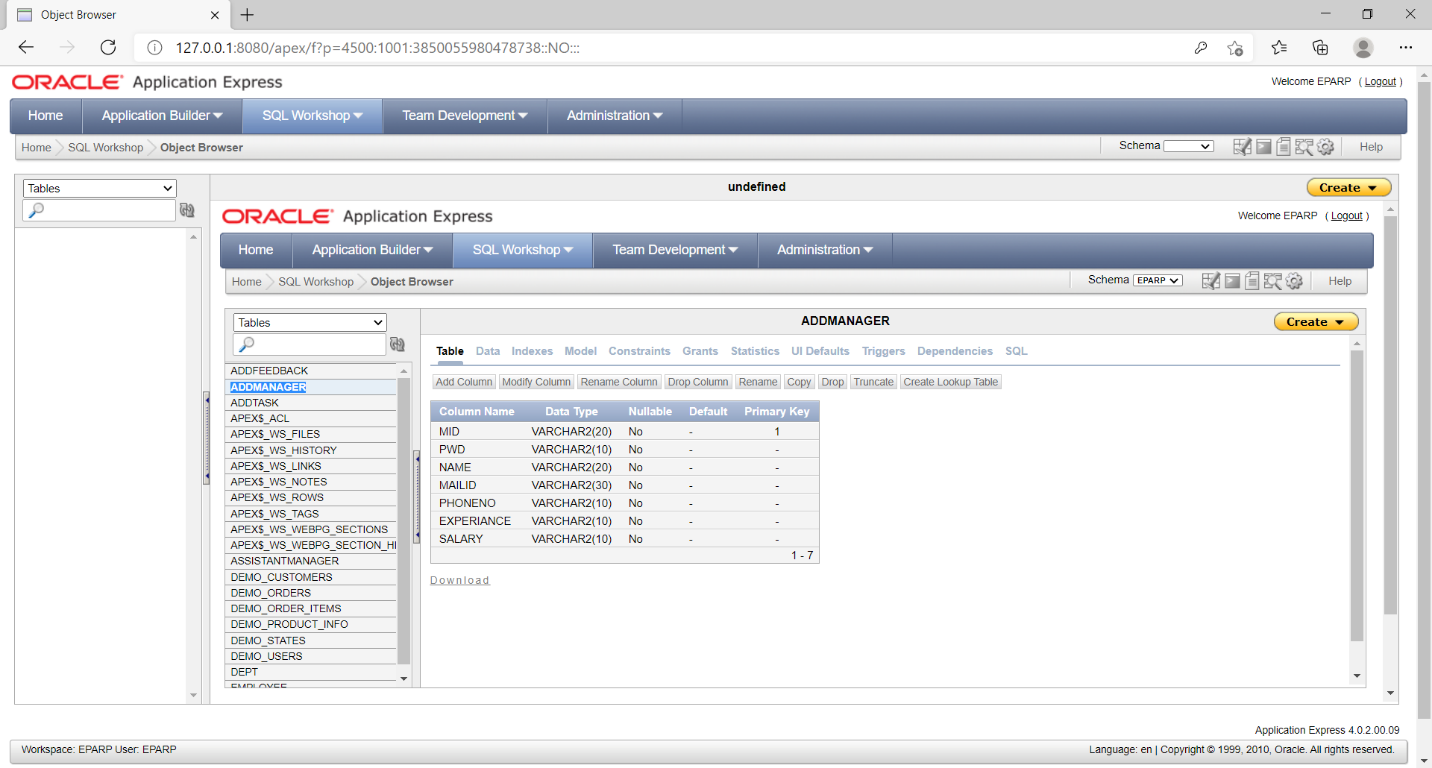
**VIEW CREDITS:**



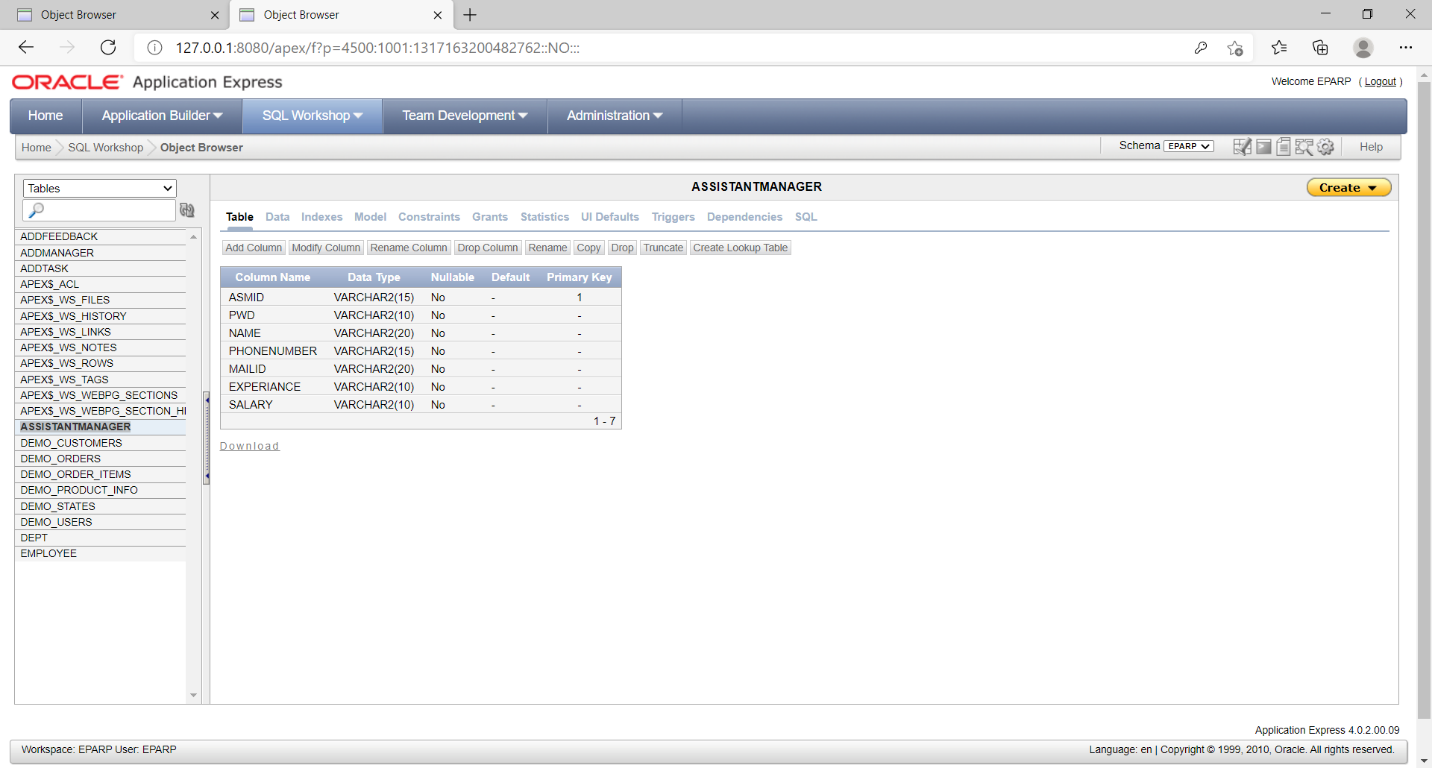
DATADASE DESIGN

TABLES

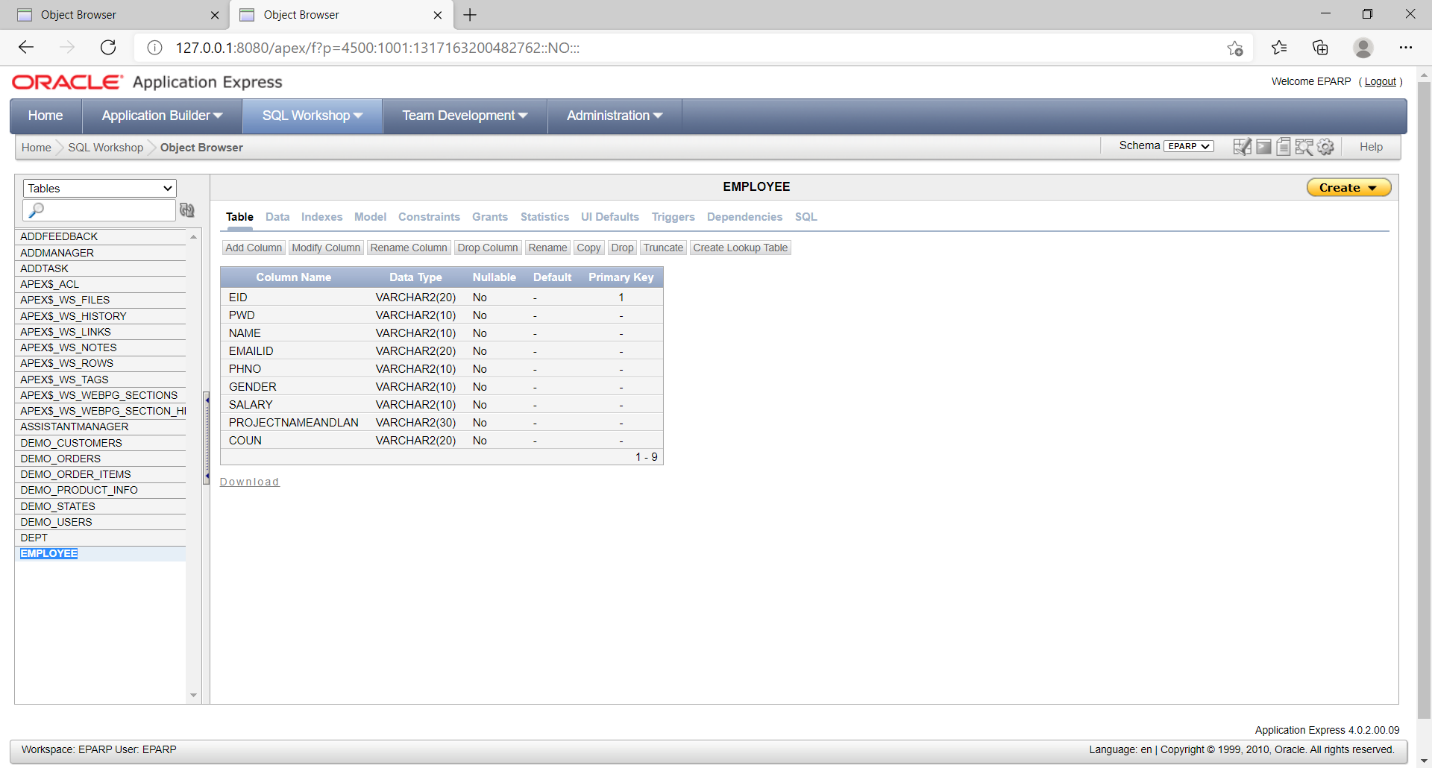
**Manager:**



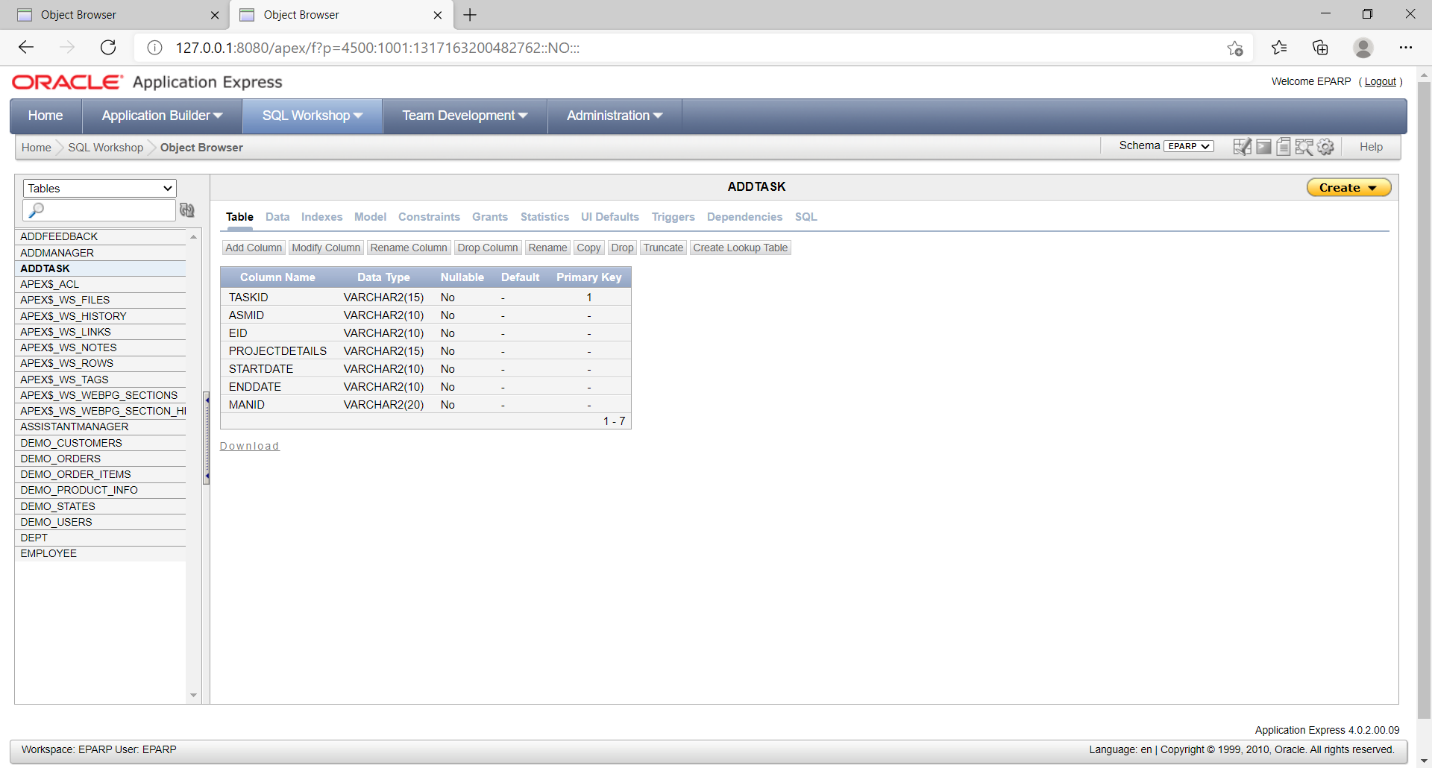
Assistant manager:



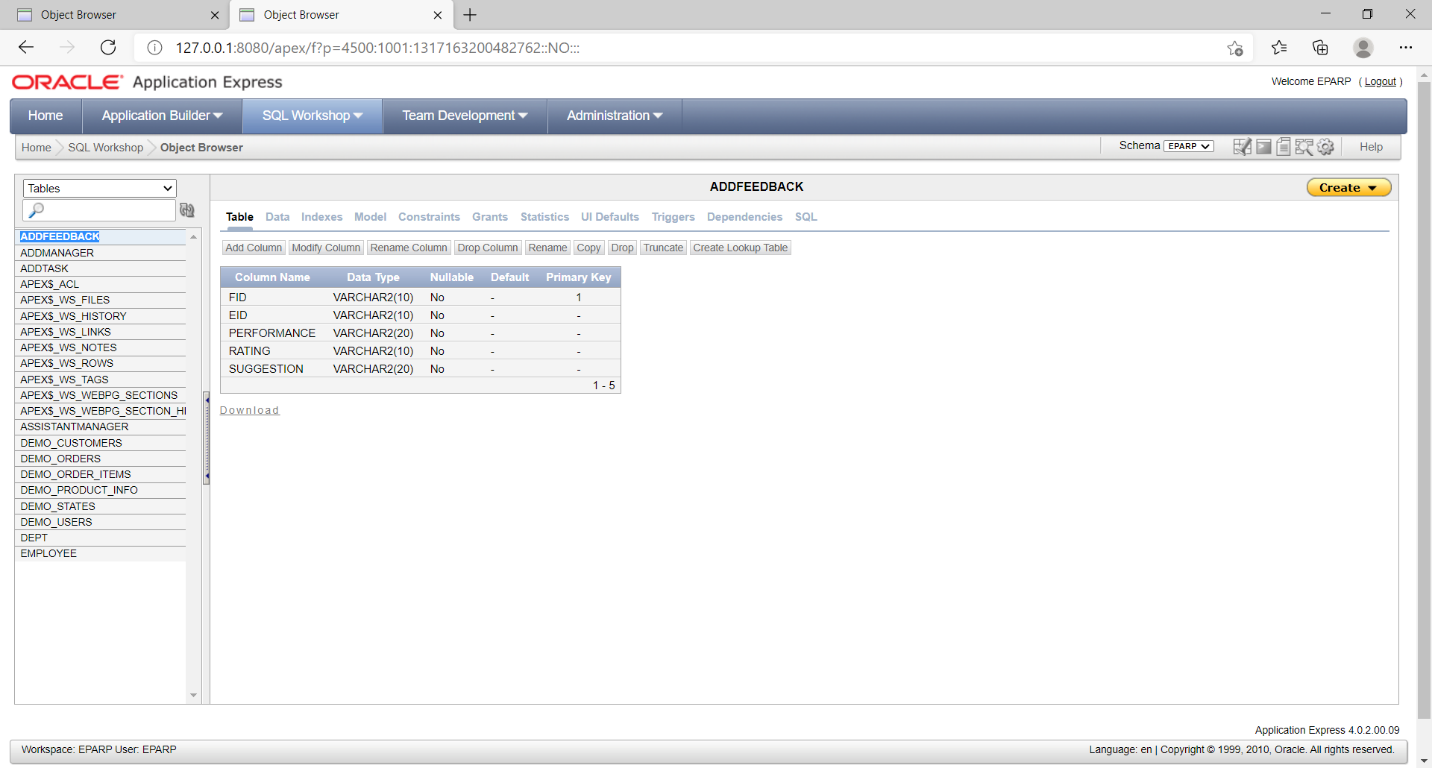
**EMPLOYEE:**



**TASK:**



**FEEDBACK:**



# TEST REPORT AND TEST PLAN

**INTRODUCTION:**

**Test Reports, Test Plan:**

Software testing is a process of validating and verifying that a software program /application/product meets:

1. The business and technical requirements that guided its design and development

2. Works as expected can be implemented with the same characteristics.

**Purpose:**

A primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. Software testing includes examination of code as well as execution of that code in various environments and conditions.

It also includes examining the aspects of code: does it do what it is supposed to do and do what it needs to do. Information derived from software testing may be used to correct the process by which software is developed.

**Testing Methods:**

Software testing methods are divided into White, Black and Grey box testing. These approaches are used to describe the point of view that a test engineer takes when designing test cases.

**White Box Testing:**

White box testing is done when the tester has access to the internal data structures and algorithms including the code that implements it.

**Black Box Testing:**

Black box testing treats the software as a “black box” without any knowledge of internal implementation.

**Grey Box Testing:**

Grey box testing involves having knowledge of internal data structures and algorithms for purpose of designing the test cases, but testing at the user, or black-box level.

Grey box testing may also include reverse engineering to determine, for instance, boundary values or error messages.

**Testing Levels:**

Tests are frequently grouped by where they are added in the software development process, or by the level of specificity of the test.

**Unit Testing:**

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In the object-oriented environment, this usually at class level, and the minimal unit tests include the constructors and destructors.

**Integration testing:**

Integration testing is any type of software testing that seeks to verify the interfaces between COMPONENTS

**System testing:**

System testing tests a completely integrated system to verify that it meets its requirements.

**System Integration Testing:**

System integration testing verifies that a system is integrated to any external or third party systems defined in the system requirements.

**Test Report 1:**

**Project Name:**  EMPLOYEE PERFORMANCE APPRISAL AND RATING PORTAL

**Form Name:** Login

**Unit Name:** User-id, Password

**Test Result:** After entering two fields the user successfully logs into the system

**Test Plan 1:**

**Unit id** : Login

**Test Case id** : User-id

**Test Type** : Unit Testing

**Form Name** : Login

**Base Table** : Admin/Citizen/Station

**Purpose** : To give access to the user after he/she enters valid user-id and password.

**Description:**

User id : varchar (40)

Password : varchar (40)

**Test data:**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Input Specification** | **Output Specification** |
| **1** | **User id:**  Valid Input  Invalid input | Navigates to the respective home pages.  It will ask to enter correct values again. |
| **2** | **Password:**  Valid Input  Invalid input | Navigates to the respective home page.  It will ask to enter correct values again. |

**Test Process:**

Login form will be used for allowing the correct user to use the software. Every person will be given a user id and password. After successful login the user can use the software as per the privileges given to him. The user id will be entered in the textbox given for user id. The password will be entered in the textbox given for password**.**

**Test completion criteria:**

When expected results match the actual results after performing the test, the test is considered to be complete.

# CONCLUSION:

The **employee performance appraisal and rating portal** is a web-based application for primarily providing training to the employees who provide customized solutions to meet organizational needs.

This application software has been computed successfully and was also tested successfully by taking “test cases”. It is user friendly, and has required options, which can be utilized by the user to perform the desired operations.

The software is developed using ASP.Net as front end and SQL as back end in Windows environment. The goals that are achieved by the software are:

* Instant access.
* Improved productivity.
* Optimum utilization of resources.
* Efficient management of records.
* Simplification of the operations.
* Less processing time and getting required information.
* User friendly.

**BIBLIOGRAPHY:**

[www.java.com](http://www.java.com)

[www.oracle10g.com](http://www.oracle10g.com)

[www.google.com](http://www.google.com)

The Complete Reference Java Fourth Edition