**Project Report**

**On**

**EMPLOYEE RECOGNITION**

Submitted In The Partial Fulfillment Of The Requirements For The Award Of The Degree OF

**BACHELOR IN COMPUTER APPLICATIONS**

OF

**GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY**

**DELHI**

Under Supervision of: Submitted by:

**Ms. AMANDEEP KAUR SIDIT NISCHAL**

**RAVNEET KAUR**

**MUKUL KATHURIA**



**Session 2013 - 2016**

**SRI GURU TEGH BAHADUR INSTITUTE OF MANAGEMENT & INFORMATION TECHNOLOGY**

**Gurdwara Nanak Piao Campus, Delhi-110033**

(Affiliated To Guru Gobind Singh Indraprastha University)  
(Sector–16 C, Dwarka, Delhi - 110078, India)

**Certificate**

This to certify that **SIDIT NISCHAL** and **RAVNEET KAUR** and **MUKUL KATHURIA** university enrollment number 12090202013 and 13690202013 and 11190202013 has worked under my supervision of to prepare this project report of **“EMPLOYEE RECOGNITION”.** The work embodied in this report is original and is standard expected of BCA student and has not been submitted in part or full to this or any university for the award of any degree diploma, He has completed all requirement of guidelines for research project and the work is fit for evaluation.

Date

Signature of HOD(IT)

Signature of Guide:

**DECLARATION**

I hereby declared that work presented in work entitled HUMAN RESOURCE SOLUTION in partial fulfillment of the requirements for the award of degree of bachelor of computer application submitted in “**SRI GURU TEGH BAHADUAR & INFORMATION TECHNOLOGY”** affiliated to **GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY** is a authentic record of my own work carried out under the supervision of **MS AMANDEEP KAUR**

This project is not submitted by me for the award of any degree or diploma.

**DATE:**

**ACKNOWLEDGEMENT**

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our project. Whatever we have done is only due to such guidance and assistance and we would not forget to thank them.

We express our deepest gratitude to **MS AMANPREET KAUR, (HOD), SGTBIM&IT,**Sri Guru Tegh Bahadur Institute of Management & Information Technology **,** for extending her inevitable and valuable support to us.

We owe our profound gratitude to our **guide** for the project **Ms. AMANDEEP KAUR** who took keen interest in our project work and guided us all along till the completion of our project work by providing all the necessary information for developing a good system. She has taken immense effort and pain to go through the project and make necessary corrections as and when needed.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from our faculty members without whom this project would have been a distant reality. We would also like to extend our sincere regards to our families, and well wishers for their timely support.

SIDIT NISCHAL(12090202013)

RAVNEET KAUR(13690202013)

MUKUL KATHURIA(11190202013)

**Abstract**

Employee Performance and Recognition Portal is online Performance Appraisal and Recognition system used by all the employees in different sections of the company. Salary hike and promotion depends upon the employee performance. This portal is a one stop shop for all the employees to provide details like tasks performed and performance measures improved etc to their superiors. It allows superiors to evaluate and analyze the employee’s performance and work done by him and target achieved in a given period of time and to rate him. It provides a very good interface between superiors and subordinates. Based on these ratings and overall performance shown by employees ranks will be allotted to them among the group of employees with same designation. This rank is called consolidated rank. This is the basic criteria for recognizing employee’s performance and to provide salary hike/promotion to any employee.

This application maintains the entire data in a centralized and secured database server to maintain consistency in report generation and allows users to access from any location. This is an online application that allows multi user access of system and to track or manage the data simultaneously. Various roles and authentications have been provided and access to various areas in the tool is restricted according to the role given to users.

**Chapter-1: INTRODUCTION**

* 1. **INTRODUCTION**

Employee Performance and Recognition Portal is online Performance Appraisal and Recognition system used by all the employees in different sections of the company. Salary hike and promotion depends upon the employee performance. This portal is a one stop shop for all the employees to provide details like tasks performed and performance measures improved etc to their superiors. It allows superiors to evaluate and analyze the employee’s performance and work done by him and target achieved in a given period of time and to rate him. It provides a very good interface between superiors and subordinates. Based on these ratings and overall performance shown by employees ranks will be allotted to them among the group of employees with same designation. This rank is called consolidated rank. This is the basic criteria for recognizing employee’s performance and to provide salary hike/promotion to any employee.

**1.2 Objective of the System**

To develop a software application that supports the application specific to the ER automation in an intranet specific to a company there by allowing the integration of all the employees pertaining to that organization. To keep track of all the other departments related to that organization such as marketing, development etc.

To allow the ER of an organization to update the employee details when ever there is a change in the employee profile pertaining to that organization. To bring onto a string the employee specific suggestions and make them free to post their requirements to the ER thus bringing the organization more specific regarding the maintenance of the organization.

**1.3 PURPOSE OF THE PROJECT:**

The ER center is a powerful application designed to allow companies to streamline their employee recognizer tasks and manager their employees more efficiently

Employee and Company Information

Employee Time, Attendence, and Leave Request

ER Documentation Management (i.e. insurance forms, W-2’s etc.)

The ER Center includes a comprehensive employee information database, where employers can information, work information, beneficiary information, and more for each employee. It comes standard with employee self-service access allowing employees to update their personal information, request time off or input their daily timesheet entries. It also has role –based access level control that is functionally based on whether a user is an employee, a manager, or an HR admistrators. With ER center managers and ER admistrators can manage an track

**1.4 Scope of the Study**

This intranet application has been developed to be implemented in place of existing manual system. This application provides effective appraisal system for the employees by evaluating their performances and retains the present functionality available in the current system. The specific purpose of this system is to store and process information about different tasks, weightage, self ratings and final rating at different stages and generates the reports as and when required. The administrator is responsible for maintenance of this system. Based on the Type of user category of the user.

**PROJECT OVERVIEW:**

This project can be used to identify a employee in the organization. The project maintains the details of the entire employee. Each employee is given with different employee Id.

**Chapter-2: REQUIREMENT ANALYSIS**

**2.1 Introduction**

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

Requirements analysis is critical to the success of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

**2.2 SDLC Approach**

The software development life cycle (SDLC) is process based standard practice to develop any kind of software product. These processes are categories as phases in SDLC and related to different activities. Different models are described in software engineering text books, but none is fully satisfied all the need of a software companies. Some SDLC models are water fall model, spiral models, prototype models etc.

In this project we use the waterfall model which are described below:

The different phases of waterfall models are as:

**Requirements analysis and specification**

**Design and specification**

**Coding and module testing**

**Integration and**

**Implementation**

**Delivery and maintenance**

**Figure 1**

1. Requirement analysis and specification phase:

The goal of this phase is to understand the exact requirements of the customer and to documents them properly. This activity is usually executed together with the customer, as the goal is to document a functions, performance and interfacing requirements for the software. The requirements describes the “what’’ of a system, not the ‘’how’’, this phase produces a large document, written in normal language.

1. Design phase:

The SRS document is produced in the previous in the previous phase, which contains the exact requirements of the customer. The goal of this phase is to transform the requirements specification into structure that is suitable for implementation in some programming language. Here, overall software architecture is defined, and the high level and detailed design work is performed. This work is documented and known as software design description (SSD) document.

1. Implementation and unit testing phase:

During this phase, design is implemented , if the SSD is complete, the implementation or coding phase, design is implemented or coding phase proceeds smoothly, because all the conformation needed by the software developers is contained in the SSD.

During testing, the major activities are centered around the examination and modification of the code. Initially, small modules are tested in isolation from the rest of the software product. There is problems association with testing a module in isolation. How do we run a module without anything to call it, to be called by it or, possibly, to output intermediate values obtained during execution? Such problems are solved in this phase and modules are tested after writing some overhead code.

1. Integration and system testing phase:

This is very important phase. Effective testing will contribute to the delivery of higher quality software products, more satisfied users, lower maintenance costs, and more accurate and reliable results. It is a very expensive activity and consumers one –third to one- half of the cost of a typical development project.

The purpose of unit testing is to determine that each independent module is correctly implemented. This gives little chance to determine that the interface between modules is also correct, and for this reason integration testing is performed, System testing involves the testing of the entire system, whereas software is a part of the system.

1. Operation and maintenance phases:

Software maintenance is a task that every development group has to face, when the software is delivered to the customer’s sites, installed and Its operational. Therefore, release of software inaugurates the operation and maintenance phase of the cycle. The time spent and effort required to keep the software operational after release is very significant. Despite the fact that is a very important and challenging task; it is routinely the poorly managed headache that nobody wants to face.

**2.3 Technologies Required**

Following is a tabular representation of the technology to be used for the development of the proposed solution:

**Hard ware Specification**:

Processor **:** Intel P-III based system

Processor Speed  **:** 250 MHz to 833MHz

RAM **:** 64MB to 256MB

Hard Disk **:** 2GB to 30GB

Key Board **:** 104 keys

**Software Specification**

Language **:** JDK 1.4

Database **: Oracle 9i**

Operating System **:** WindowsNT/95/98/2000

RAM **:** 256MB

**2.4 USE CASE DIAGRAM**

The Use case describes what of a system not “how”. They only gives functional view of the system. Use cases are structured outline or templates for the description of user requirement, model in a structured language like English. Use case diagrams are graphical representation that may be decomposed into further levels of the abstraction.

**Actor**

An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects.

**Association**

An association specifies a semantic relationship that can occur between typed instances. It has at least two ends represented by properties, each of which is connected to the type of the end. More than one end of the association may have the same type.

**System**

If a subject (or system boundary) is displayed, the use case ellipse is visually located inside the system boundary rectangle. Note that this does not necessarily mean that the subject classifier owns the contained use cases, but merely that the use case applies to that classifier

**Use Case**

A use case is the specification of a set of actions performed by a system, which yields an observable result that, is, typically, of value for one or more actors or other stakeholders of the system

ER

Reviewer

Emp

**Chapter-3: SOFTWARE DESIGN**

**3.1 Introduction**

Software design is an engineering process by which we are representing of something that is to be built.

It is a blueprint for constructing the software. In the software engineering context, design focuses on four major area of concern: data, architecture, interfaces and component. Once the software requirements have been analyzed and specified, software design is the first of three technical activities—design, code generation, and test—are required to build the and verify the software. After making Software Requirement Specification of the project, now we are in position to design the software.

The system design shows how Software will be structured to satisfy the requirements identified during the analysis phase. The design process is a translation of requirements into a description of the software structure, software components, interfaces and data necessary for the implementation phase. The design phase provides a complete blueprint for the implementation activity.

System analysis begins by identifying the reports and the other outputs the system will produce. Then the specific on each are pin pointed. Usually, designers sketch the form or display as they expect it to appear when the system is complete. This may be done, on a paper or computer display, using one of the automated tools available. The system design also describes the data to be the input, calculated or stored. Individual data items and calculation procedures are written in detail. The procedures tell how to process the data and produce the output.

* 1. **DATA FLOW DIAGRAMS**

The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purposesingle process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process. Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical from, this lead to the modular design.

A DFD is also known as a “bubble Chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

**3.2.1 DFD SYMBOLS:**

In the DFD, there are four symbols

1. A square defines a source(originator) or destination of system data
2. An arrow identifies data flow. It is the pipeline through which the information flows
3. A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
4. An open rectangle is a data store, data at rest or a temporary repository of data

Process that transforms data flow.

Source or Destination of data

Data flow

Data Store

**3.2.2 CONSTRUCTING A DFD:**

Several rules of thumb are used in drawing DFD’S:

1. Process should be named and numbered for an easy reference. Each name should be representative of the process.
2. The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.
3. When a process is exploded into lower level details, they are numbered.

**0 LEVEL D.F.D.**

This is the context level D.F.D. of the proposed system the whole system has been depicted in a single bubble, primary input and output has been carefully noted and depicted in the way so that information flow continuity should not be lost in the next level. The proposed system is shown as a whole process and the inputs and outputs are shown with incoming and outgoing arrow from the system.

ER

EMPLOYEE

REVIEWER

**1 LEVEL D.F.D**

Is an

Employee

Manages

Database

ER

Access

Login

Registeration

Creates

Reviewer

Details

Reports

Creates

Employee

Details

Provides

Projects

**CHAPTER-4 DATABASE DESIGN**

* 1. **ENTITY–RELATIONSHIP DIAGRAMS:**

1. An **entity–relationship model** (**ER model**) is a data model for describing the data or information aspects of a business domain or its process requirements, in an abstract way that lends itself to ultimately being implemented in a database such as a relational database. The main components of ER models are entities (things) and the relationships that can exist among them.
2. Entity–relationship modeling was developed by Peter Chen and published in a 1976 paper. However, variants of the idea existed previously, and have been devised subsequently such as supertype and subtype data entities and commonality relationships

ER

Employee

CREATES

creates

Reviewer

**Chapter-5: TESTING**

**5.1 Introduction**

Executing a program with the intent of finding errors is called testing. Testing is vital to the success of any system. Testing is done at different stages within the development phase. System testing makes a logical assumption that if all parts of the system arte correct, the goals will be achieved successfully. Inadequate testing or no testing at all leads to errors that may come up after a long time when correction would be extremely implementation. The testing of the system was done on both artificial and live data. In order to test data test cases are developed. Following are the various methods that are employed for testing:

**Unit Testing**

In unit testing the module is tested independently. It is done to test that the module does satisfy the functional specification. This is done to check syntax and logical errors in programs. At the time of preparation of technical specifications, unit test data was also prepared. The coding for that program was considered after verifying its output against this test data.

Following are the unit testing methods:

In Conditional Testing, the logical conditions that are given in the module were checked to see whether they satisfy the functionality of the module. This is done by using the test data was prepared.

In Loop Testing, different loops

in the module like nested loops were tested using the data. Attempts to execute the loops to their maximum range are done.

**Integration Testing**

In Integration testing whole system was checked when all the individual modules were integrated together in order to test whether the system is performing as according to the requirements specified. Interface errors if any were corrected.

Test data was prepared was fed into the system to check whether the system fails to detects an error.

**Functional Testing**

This is done for each module/sub module of the system. Functional testing serve as a means of validating whether the functionality of the system confers the original user requirement i.e. does the module do what it was supposed to do? Separate schedules were made for functional testing. It involves preparation of test data, writing of test cases, testing for conformance to test cases and preparation of bugs’ listing for non-conformities.

**System Testing**

System testing is done when the entire system has been fully integrated. The purpose of the system testing is to test how the different modules interact with each other and whether the entire system provides the functionality that was expected.System testing consists of the following steps:

* Program Testing
* System Testing
* System Documentation
* User Acceptance Testing

**5.2 Test Plan (TP)**

**Module : ER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Actual Output | Obtained Output | Desc |
| Username Status | Username | Success | Success | Test Passed.  Issue Account, associate expiry info. |
| Username Status | Username | Failed | Failed | Test Passed.  Invalid data |

**Module : Password**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Actual Output | Obtained Output | Desc |
| Password Status | Pasword Info | Success | Success | Test Passed.  Password issued & saved . |
| Password Status | Password Info | Failed | Failed | Test Passed.  Invalid data or test failed. Try Again |

**EMPLOYEE**

**Module : E\_Code**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Actual Output | Obtained Output | Desc |
| E code Status | E code | Success | Success | Test Passed.  Issue Account, associate expiry info. |
| E code Status | E code | Failed | Failed | Test Passed.  Invalid data |

**Module : Password**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Actual Output | Obtained Output | Desc |
| Password Status | Pasword Info | Success | Success | Test Passed.  Password issued & saved . |
| Password Status | Password Info | Failed | Failed | Test Passed.  Invalid data  Try Again |

**Reviewer**

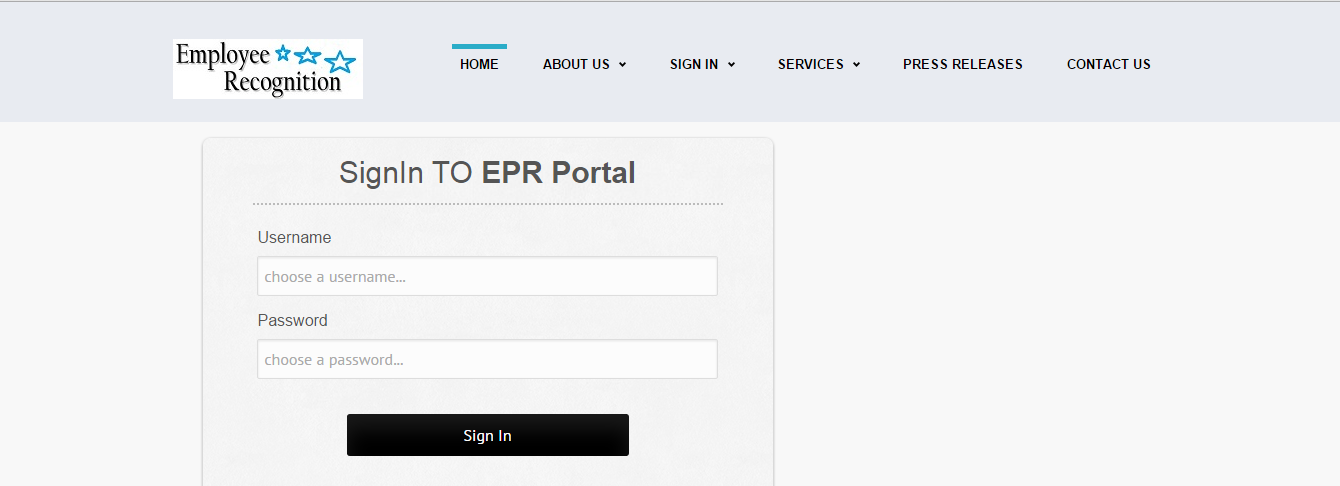
**Module : R Code**

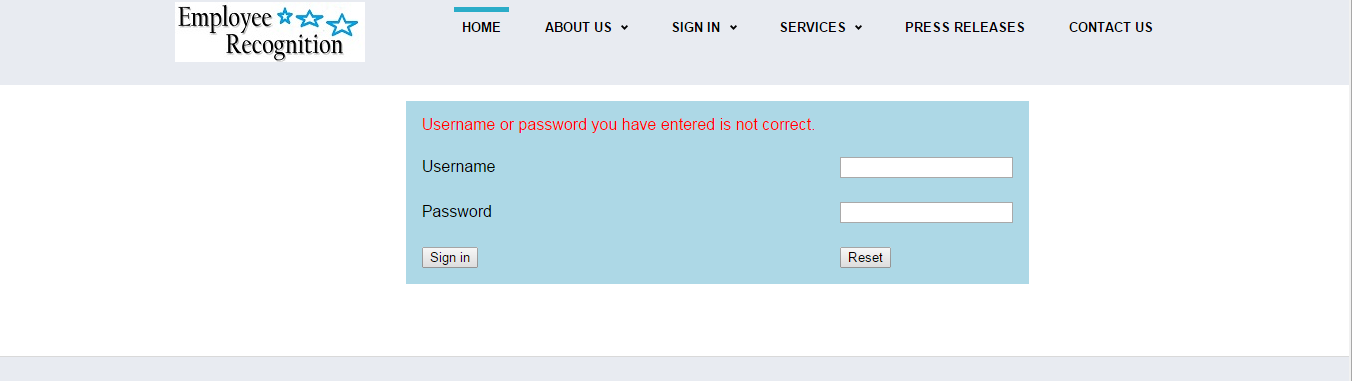
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Actual Output | Obtained Output | Desc |
| R Code Status | R Code | Success | Success | Test Passed.  Issue Account, associate expiry info. |
| R Code Status | R Code | Failed | Failed | Test Passed.  Invalid data |

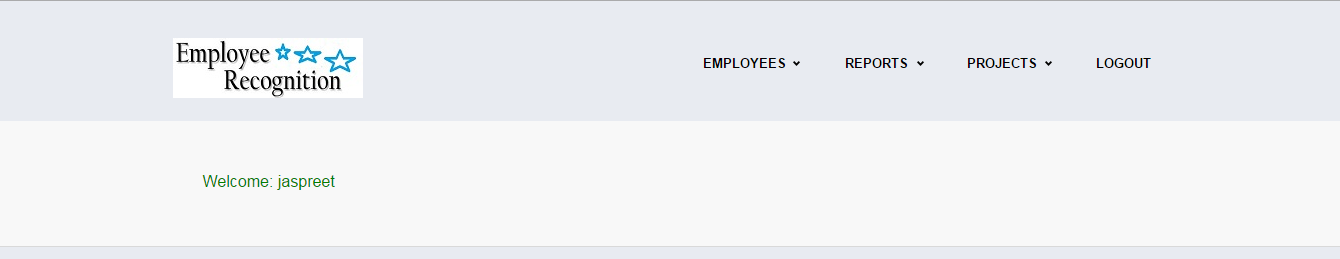
**Module : Password**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Actual Output | Obtained Output | Desc |
| Password Status | Pasword Info | Success | Success | Test Passed.  Password issued & saved . |
| Password Status | Password Info | Failed | Failed | Test Passed.  Invalid data  Try Again |

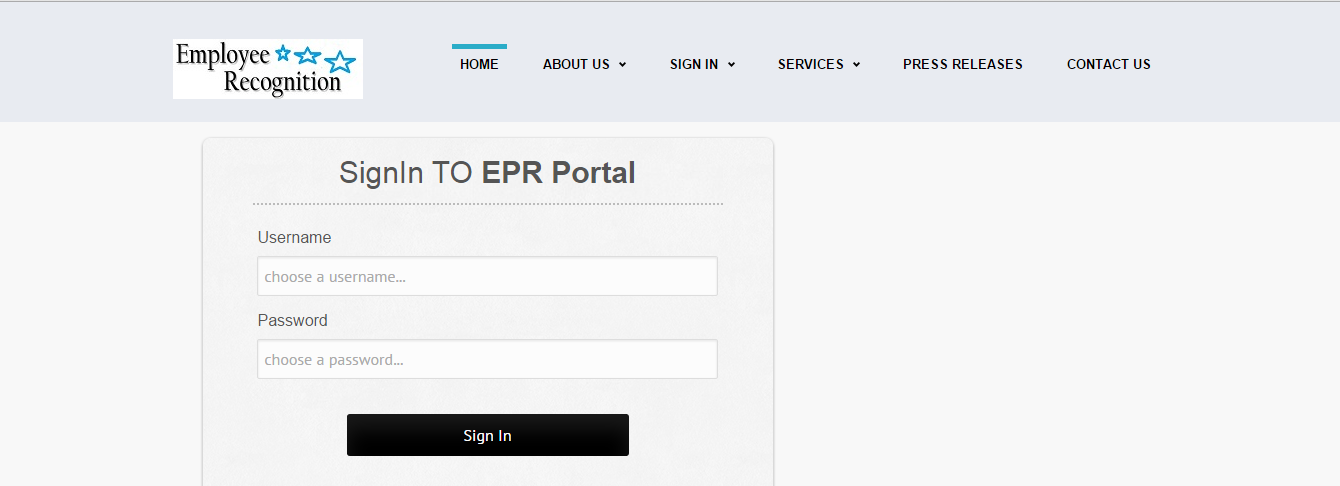
For ER

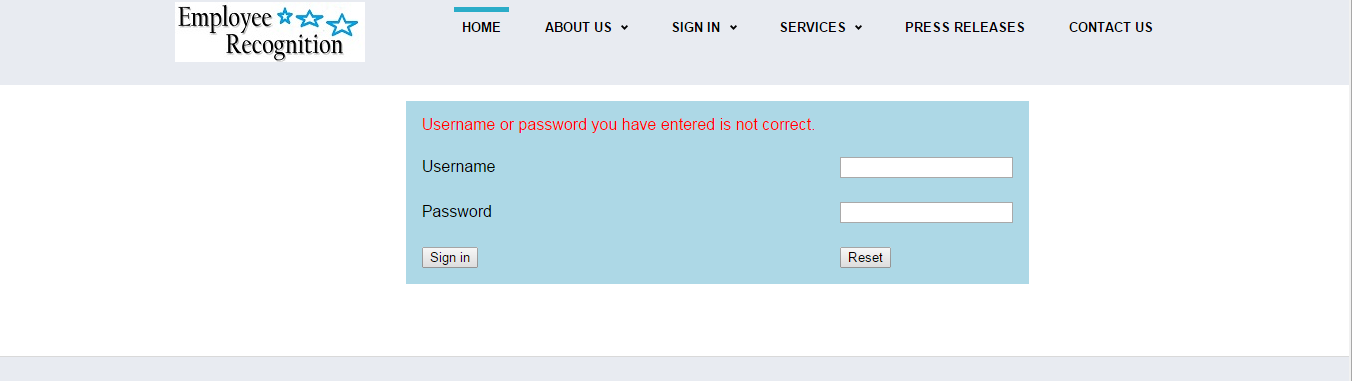


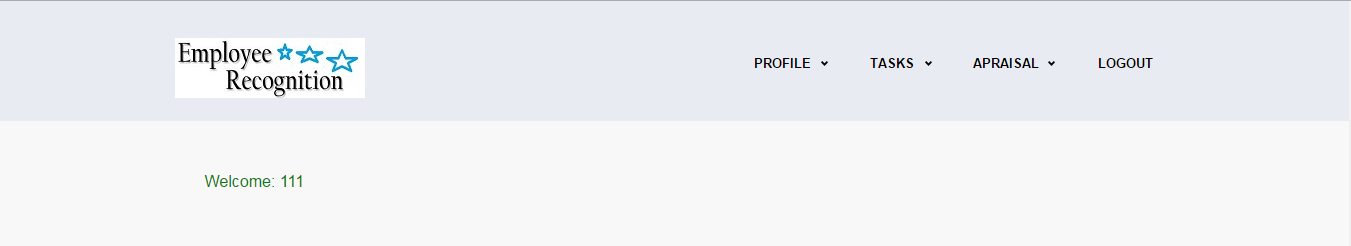




FOR EMPLOEE







FOR EMPLOEE

