**Online Application Title: Administration of E-Placement System**

**Team: Girish Dhande (1026)**

**Gaurav Patil (1025)**

**SRS Document:**

**Introduction**

**Administration of E-Placement System**

**Purpose:**

The purpose of the “Administration of E-Placement System”, the manual work makes the process slow and other problems such as inconsistency and ambiguity on operations. In order to avoid this the E-Placement System is proposed, where the student information in the college with regard to placement is managed efficiently. It intends to help fast in fast access procedures in placement related activities and ensures to maintain the details of the student. Students logging should be able to upload their personal and educational information. The key feature of this system is that it is one-time registration enabled. The placement cell calls the companies to select their students for jobs via the campus interview. The placement cell allows the companies to view the student resumes in selective manner. They can filter the students profile as per their requirement. The job details of the placed students will be provided by the administrator. The administrator plays an important role in our system. Our system provides the facility of maintaining the details of the students and gets the requested list of candidates for the company who would like to recruit the students based on given query.

**Scope:**

This system has a big scope to do. Students can access previous information about placement. We can store information of all students. Various companies can access their information. Notifications are sent to students about the companies.

**Definitions:**

AES --> Administration of E-Placement System

TPO --> Training Placement Officer

SRS --> Software Requirement Specification

GUI --> Graphical User Interface

Portal --> Personalized Website

Stack holder --> The person who will participate in the System. And Owner of system

Ex. Student, TPO, Recruiter

UML---> Software Engineering Notation for visualising System in the form diagrams

SSL--->Secure Socket Layer used for providing restricted access to application.

BOD---> Board of Directors (Management).

RDBMS --> Relational Database Management System.

CLUSTERS---> Group of independent servers.

**Overview:**

This system is aim at developing a web application for the Placement Department of the College. The system is a web application that can be accessed throughout the organization with proper login provided. This system can be used as a web application for the Training and Placement Officers (TPO) of the college to manage the student information with regard to placement. Students logging should be able to upload their information in the form of a CV. The key feature of this project is that it is a onetime registration. Our project provides the facility of maintaining the details of the students. It also provides a requested list of candidates to recruit the students based on given query. Administrator logging in may also search any information put up by the students. This system will aid colleges to practice full IT deployment. This will also help in fast access procedures in placement related activities.

**General Description:**

The aim of the proposed system is to develop a system with improved facilities. This system can overcome all the limitation of the existing system, such as student’s information is maintained in the database, it gives more security to data, ensures data accuracy, reduces paper work and save time, only eligible students get chance, it makes information flow efficient and paves way for easy report generation, reduce the space. proposed system is cost effective.

**Functional Requirement:**

This section provides requirement overview of the system. Various functional modules that can be implemented by the system will be-

**Description:**

Registration if user wants to login then he/she must be registered, Unregistered user cannot get login. Login user logins to the system by entering valid user id and password for going to welcome page.

After login, user will be identified if user is Training and Placement Officers (TPO) then he/she can see the student list and can filter it as per requirement, also Training and Placement Officers (TPO) can add company information on portal and notification can be send eligible student list. If login user is student then he/she can edit the information of his/her profile and can see notification of company to which he/she is eligible.

And if login user is recruiter then they can add recruitment notification by all their requirement.

**Technical Issues:**

This system will work on client-Server architecture. It will require an internet server.

The system should support some commonly used browser such as Chrome etc.

Interface Requirement Various interfaces for the product could be

1.Login Page,

2.Registration form

There will be a screen displaying information about student that the eligible.

The user may select the different options which will be open in another screen as,

* + 1. Login Page
    2. Registration Form
    3. Skill Mapping
    4. Scheduling
    5. Placement record
    6. Notification
    7. Feedback

Account Settings

**Hardware Interface:**

The System must run over the internet and All the hardware shall require to connect to internet will be hardware interface for the system.

e.g. modem, WAN, LAN

Specialized Server Infrastructure Hardware

The system should use distributed servers i.e. cloud for managing large amount of data so as to make it appear as single unit for end-user.

The system should have proper clusters for backup.

**Software Interface:**

The system is on server so it requires the any scripting language like JSP or PHP or ASP, ETC.

The system should be able to exchange data using XML, JASON or any advance technology.

The system requires Database also for the store the any transaction of the system like MySQL or oracle, or SQL server etc.

System also require DNS (Domain Name space) for the naming on the internet.

<http://www.amazon.in>

At the end-user need web browser for interact with the system.

**Performance Requirement:**

There is no performance requirement in this system, because the server request and response to client is totally based on internet connection of end-user.

**Design Constrains:**

This system should be developed using Standard Web Page Development Tool, which conforms GUI standards such like HTML, XML, JSON, etc.

The system should support various RDMS and Cloud Technologies.

**Non-Functional Requirements**

**1.Security:**

SSL

The System use SSL (Secure Socket Layer) in all transaction that include any confidential customer information.

The system must automatically log out all customers after a period of inactivity.

The system should not leave any cookies on the customer's computer containing user’s password.

The system's back-end servers shall only be accessible to authenticated administrators.

Sensitive data will be encrypted before being sent over insecure connections like internet.

The proper firewalls should be developed to avoid intrusions from the internal or external sources.

**2.Reliability:**

The system provides storage of all databases on redundant computers with automatic switchover.

The main pillar of reliability of the system is the backup of the database

which is continuously maintained and update to reflect the most recent changes.

**3: Availability:**

The system should be available at all times. Meaning the user can access it using web browser,

only restricted by the down time of the server on which the system runs.

In case of a of a hardware failure or database corruption, a replacement page will be shown.

uptime: It mean 24 \* 7 availability

100%--------------

99.9%

99.999%

99.9999%

**4: Maintainability:**

A commercial database is used for maintaining the database and application server takes care of the site.

The maintainability can be done efficiently.

**5.Portability:**

The application is HTML and scripting language based (JavaScript). So the end user part is fully portable and any system using

any web browser should be able to use the features of the system, including any hardware platform that is available

or will be available in the future.

An end-user is used this system on an OS; either it is Windows or Linux.

The System shall run on PC, Laptops and PDA, etc.

The technology should be transferable to different environments easily.

**6.Accessibility:**

Only registered users should be allowed to see their profile after authentications.

Only GUI access of the system should be permitted to end users.

**7.Policies:**

The system should adhere to all the legal formalities of the particular countries.

The system should maintain security related to sensitive data.

**8.Efficiency:**

The system should provide good throughput and response to multiple users without burdening the system by using appropriate number of servers.

**9.Safety:**

Software should not harm ethical and environmental conditions of the end users machine.

**10.Modulariy:**

The system should have user friendly interface.

It should be easily updated, modified and reused.

**Operational Scenario:**

**Training and Placement Officers (TPO) Interaction**

The Training and Placement Officers (TPO) can filter student as per company requirement. The system shows all the student to Training and Placement Officers (TPO) who are eligible to apply for this company.

Then Training and Placement Officers (TPO) will select student which are not placed among all filter student. The notification will be send to student profile which are selected by the Training and Placement Officers (TPO).

Training and Placement Officers (TPO) will collect feedback from student about who want to apply for company. A then this information will be forwarded to company and a campus drive is arranging as per time set by company and

Training and Placement Officers (TPO). A timetable will to forwarded to student. After campus drive Training and Placement Officers (TPO) will collect feedback from company the list of selected student. And this feedback is forwarded to student.

**Student Interaction:**

The Student can create the resume by filling their information on this system. They can edit their information. Student can see the notification send by Training and Placement Officers (TPO) and can collect information about company from database.

and depending on their requirement they can filter job which are posted by Training and Placement Officers (TPO) on their profile and can apply to their selected company.

**Company Interaction:**

The Company can create their profile on system and can edit their profile. They can post their job by filling form of job post by filling all their requirement which they need in candidate

and can forward this information to Training and Placement Officers (TPO) and get student list who are eligible for this job and can arrange campus drive.

**Head of Department (HOD):**

The Head of Department (HOD) can monitor all the activity of placement. He/she can see the record of placed and non-placed student.

**Preliminary Schedule:**

1. Login
2. Manage Training and Placement Officers (TPO) database Browse category
3. Manage student database
4. Manage company database
5. add or remove student from list
6. update information
7. approve/reject student
8. Logout
9. Give feedback
10. Create new account
11. View account details
12. Registration
13. Placement tracking
14. Manage Alumina Database
15. Manage Head Of Department(HOD) Database

**ER listing:**

**Application Architecture:**

Application = Logic + data

Logic = (UI Logic + Business Logic + Data Access Logic)

Data = (structured data, Non Structured data)

**Online Application:**

Web based Application deployed on web and accessed by users from anywhere Administration of E-Placement System (OAPMS)-----Web portal-- used remotely by HOD, TPO, Students, Company.

**Logic:**

**UI Logic:**

Web Pages + HTML controls + Web Components (Angular)

Navigation: (UI Routing) HTML Links, Routing mechanism

Data Binding: DOM + JSP tags (JSTL) + {{}} model,

Event Binding: action handlers

HTTP Request: GET: ----------------Do get

POST: ---------------Do post

PUT:

DELETE: Client Side UI----HTML, CSS, JavaScript, bootstrap

UI (Client Side UI Framework)

Angular, React, Vue

**Web Logic: (Server Side processing)**

Server UI---------------- JSP, servlet, (classical java web technology)

spring MVC (to take advantage of MVC design Pattern using readymade formwork)

Model, View, Controller

Router

(SOA layer)

Spring Boot API

CRUD REST API

ORM Technique: Hibernate (ORM)

JPA

JDBC (database Connectivity)

**State management**

Client Side state management

cookies, query string, form collection, hidden variables

local storage, session storage, Web sql,

Server Side state management

session, Cache,

database

**Business Logic:**

Java console application will be used to test your business Logic

Core Java:will contain

1.business query processing

2.business operation management

3.Business data manipulation

from Administration of E-Placement System (OAPMS) system point of view Modules

**1.Registration:**

: Student, Head of Department(HOD), Company, Training and Placement Officer (TPO)

create, insert, delete, update

**2.Skill Mapping**

: Training and Placement Officer (TPO)

get Company Detail

sort Student List

get sorted students list.

**3.Scheduling**

: Training and Placement Officer (TPO)

: get schedule from recruiter

: send notification about Schedule to student

: Company

: send schedule to Training and Placement Officer (TPO)

: Student

: see notification about company schedule by Training and Placement Officer (TPO)

**4. Placement record**

: Training and Placement Officer (TPO)

: insert placed student list in alumina database

: Head of Department (HOD)

: get detail about placed students

**5.Notification**

: Training and Placement Officer (TPO)

: get Company notification

: Student

: get notification from Training and Placement Officer (TPO)

: get Company Detail

**6.Feedback**

: Training and Placement Officer (TPO)

: send sorted student list to company

: send feedback to student about placement

: Student

: get feedback about placement

**7.Authentication**

: validate user

: identify user

**Data:**

Structured Data

RDBMS

fields

tables

constraints

Add some dummy records in your newly created database

write reusable SQL queries against those database tables to check business Queries

Test those SQL queries on existing dummy database you built

**List of Tables**

**Head of Department(HOD)**

Fields:

Id, password, name.

Primary Key: Id

**Training and Placement Officer (TPO)**

Fields:Id,password,collegename,collegeCode,emailId,contactNo,address,website.

Primary Key: Id

**Students**

Fields:

Id,password,name,dob,branch,emailId,contactNo,address,collegename,collegeCode.

Primary Key: Id

**Company’s**

Fields:

Id,rname,company,emailId,contactNo,address

Primary Key: Id

**Alumni’s:**

Fields:

Student\_name,company\_name,CTC,year.

**Recruitment**

Fields:

company\_name,Student\_name.

**Unstructured:**

**NoSQL**

Social Media

MongoDB

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_