**Job Recruitment Application**

**Using Salesforce**

**A Project Report Submitted**

**in partial fulfillment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**by**

**KANUKA GOEL**

**University Roll No 1900290149052**

**Under the Supervision of**

**Ms. Neelam Rawat**

**KIET Group of Institutions, Ghaziabad**



**to the**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY**

**(Formerly Uttar Pradesh Technical University) LUCKNOW**

**(July 2021)**

**DECLARATION**

I hereby declare that the work presented in this report entitled “JOB RECRUITMENT APPLICATION", was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name : Kanuka Goel

Roll. No. : 1900290149052

(Signature)

**Training letter**

Text, letter

Description automatically generated

**CERTIFICATE**

Certified that **Kanuka Goel (University Roll No. 1900290149052)** have carried out the project work having “**Job** **Recruitment app using Salesforce**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Kanuka Goel**

**University Roll No. 1900290149052**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

**Ms.Neelam Rawat**

**Associate Professor**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner Signature of External Exam**

**Dr. Ajay Kr. Shrivastava**

**Professor Head, Department of Computer Application**

**KIET Group of Institutions, Ghaziabad**

**ABSTRACT**

This project Recruitment Application is a system in which HR can automate the hiring of human resources applicant can also register themselves online, view organization requirements and apply for the suitable job. Recruitment Application provides online help to the users all over the world. This kind of application plays an important role in simplifying the recruitment process. The system has facilities where prospective candidates can upload their CV’s and other academic achievements. Earlier recruitment was done manually and it was all at a time consuming work. Now it is all possible in a fraction of second. The system has been designed to do a whole lot more than just reduce paperwork. It can make a significant contribution to a company’s marketing and sales activities. Job Recruitment Application make possible for managers to access information that is crucial to managing their staff, which they can use for human resources management, staffing and planning activities. The primary purpose to develop this system is to optimize the recruitment process for an organization. Besides, the qualified applicants could be sort by this application based on their qualifications and company requirements.

**ACKNOWLEDGEMENT**

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Ms**. **Neelam Rawat** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Kanuka Goel (MCA VIA)**

**1900290149052**

**Table of Contents**

Declaration 2

Training Letter 3

|  |  |
| --- | --- |
| Certificate | 4 |
| Abstract | 5 |
| Acknowledgement | 6 |
| Table of Contents | 7-8 |
| List of Figures | 9 |
| CHAPTER 1: INTRODUCTION   |  |  | | --- | --- | | * 1. Project Description   2. Project Scope   3. Hardware/Software Requirement Specification      1. Introduction      2. Benefit of cloud apps      3. How cloud apps work   CHAPTER 2: LITERATURE REVIEW  CHAPTER 3: FEASIBILITY STUDY  3.1 Economical feasibility  3.2 Technical Feasibility  3.3 Behavioural Feasibility  3.4 Operational Feasibility  CHAPTER 4: SYSTEM DESIGN  4.1 E.R. Diagram  4.2 Database Design  4.3DFD and flowgraph  CHAPTER 5: FORM DESIGNS  5.1 Input/Output form(Screenshots)  CHAPTER 6: CODING  CHAPTER 7: TESTING  CHAPTER 8: CONCLUSION AND FUTURE SCOPE  8.1 Limitations  8.2Future Scope  CHAPTER 9: REFERENCES  **LIST OF FIGURES**  [Figure 1 How cloud apps work 15](#_Toc78997217)  [Figure 2 ER-DIAGRAM 29](#_Toc78997218)  [Figure 3 Interview Table 33](#_Toc78997219)  [Figure 4 Candidate Table 34](#_Toc78997220)  [Figure 5 POSITION TABLE PAGE 35](#_Toc78997221)  [Figure 6 Job Seeker Data Flow 37](#_Toc78997222)  [Figure 7Inserting Job Seeker Details 38](#_Toc78997223)  [Figure 8Recruitment App Data Flow 40](#_Toc78997224)  [Figure 9 HR roles with application and Database 41](#_Toc78997225)  [Figure 10 Recruitment application Authentication flow Salesforce 42](#_Toc78997226)  [Figure 11 HR Authentication flow 43](#_Toc78997227)  [Figure 12Candidate Selection Informing Flow 44](#_Toc78997228)  [Figure 13Database Table Information 45](#_Toc78997229)  [Figure 14 Home Page 47](#_Toc78997230)  [Figure 15 Login Page 48](#_Toc78997231)  [Figure 16 Login Responsive 49](#_Toc78997232)  [Figure 17 Logged Page 50](#_Toc78997233)  [Figure 18 Jobs Tab Page 51](#_Toc78997234)  [Figure 19 Job Description 52](#_Toc78997235)  [Figure 20 Application 53](#_Toc78997236)  [Figure 21 Salesforce Login 54](#_Toc78997237)  [Figure 22 New Position Page 55](#_Toc78997238)  [Figure 23 New Candidate 56](#_Toc78997239)  [Figure 24 New Job Application 57](#_Toc78997240) | 7  8  8  8 | | 10-17  10  13  13  14  14  16  18-20  21-28  22  24  26  27  29-46  28  31  36  47-57  48  58-74  75-77  78-79  78  79  80-81 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **PROJECT DESCRIPTION**

Recruitment is a process to discover the sources of manpower to meet the requirement of the staffing schedule and to employ effective measures for attracting that manpower inadequate numbers to facilitate effective selection of efficient personnel. Every organization needs to look after recruitment and selection in the initial period and thereafter as and when additional manpower is required due to expansion and development of business activities. “Right person for the right job” is the basic principle in recruitment and selection. Every organization has to give attention to the selection of its manpower, especially its Manager. The operative manpower is equally important and essential for the orderly working of an enterprise. Every business organization/unit needs manpower for carrying different business activities smoothly and efficiently and for this recruitment and selection of suitable candidates is essential. Human Resource Management in an organization will not be possible if unsuitable persons are selected and employment in a business unit.

We have studied fresherworld.com and Shine.com. These sides only provide the candidates to the organizations but there is no feature is available to automate the recruitment process after candidate is shortlisted. These web applications do not store the data of objects who are selected in interview rounds like Aptitude, GD­PI, Technical Interview, HR Interview. These applications do not provide auto mail generation to the candidates. To solve this problem Recruitment application not only track the candidates but also it can automate the interview procedure to recruit the candidate. This application provides the auto mail generation to shortlisted candidates. These sites do not provide to storage to store your data. This application holds the list of selected object after each round of interview. An Interview has many step to select object by checking the skills like aptitude test, Group Discussion, Technical Interview HR­ Interview etc. 2.4 SOLUTION The Recruitment application is the total solution for facilitating the hiring of personnel to your organization. The Recruitment application allows maintaining your Applicant and Interviewee Lists easily with Drag ­and ­Drop lists, maintaining your Job Vacancy details easily including job description, technical and non­technical, locations, departments etc. Divide the interview process into 3 stages, Written Test, Technical Interview, HR Interview and easily advance interviewees from one stage to the next, recording their 6 skills and printing reports on their technical and non­technical merits as well as observations during the face ­to­ face interview. HR can assign positions to selected candidates, their salary, location, duration and job description. This application provides HR to track the candidate detail before and after the recruitment HR can send them email about location, job profile, salary details and new updates about company.

Recruiting software is the umbrella term for all types of software related to and used in the recruitment process. It covers the various technological solutions available for each part of the recruitment funnel: from sourcing and attracting candidates to preselecting, interviewing and eventually hiring them. Although it officially only starts after a new hire joins the company, on boarding is increasingly seen as part of the recruitment process too.

The number one reason to use recruiting software is, of course, to automate and [optimize your recruitment process](https://harver.com/blog/enterprise-recruitment-process/). Especially when you’re facing high volumes of candidates, it quickly becomes impossible to manually keep track of each individual’s progress and stage in the funnel.   Often-named benefits of using recruiting software are:

### Saving time spent on tedious tasks

Even if you’ve only got a dozen candidates that applied for a certain job, scheduling interviews, answering their questions and confirming appointments quickly takes up a lot of your time. And that’s just one example. Automating the repetitive, tedious tasks that inevitably are a part of your recruitment process, saves recruiters a lot of time. Time they can, for example, spend on building a relationship with candidates instead. Recruiting software helps [reduce time to hire](https://harver.com/blog/reduce-time-to-hire/) overall by speeding up the recruitment process.

### Increasing recruiter productivity

This is something that goes hand in hand with the automation of time-consuming tasks. As such, the use of recruitment software will help recruiters to be more productive. Not only because many time-consuming tasks are automated, but also because most of the software is cloud-based which enables recruiters to work when- and wherever.

### Improving the candidate experience

Whenever a process is manual, there is room for error and inefficiencies. This is something you ideally want to avoid when you’re dealing with candidates. Recruitment software enables you to remove inefficiencies from your recruitment process. You can use a solution to make sure candidate interviews are scheduled and confirmed, to ensure that each applicant is assessed the same way, gets asked the exact same interview questions, etc. For candidates, this means a much more pleasant [candidate experience](https://harver.com/blog/candidate-selection/#CandidateExperience). As a result, they’ll be more engaged, spread the positive word about your company and maybe even have a little fun during the recruitment process!

### Increasing candidate diversity

Recruitment software can help companies [make their recruitment process less biased](https://harver.com/blog/reduce-hiring-bias/) – and hence attract a more diverse candidate pool. There are several examples of this. It starts with the job advert. An AI-driven tool can help you write a job ad that doesn’t use words or sentences that may turn off female candidates or candidates from an ethnic minority (more on this later). The same thing goes for an AI-powered pre-employment assessment tool. All candidates go through the same online experience and are assessed based on their actions and answers rather than on their gender, name, zip code or picture. Based on only data an algorithm predict their likelihood to succeed on the job.

## Recruiting software market

The recruiting software market is vast and highly fragmented. Since there are numerous solutions – and vendors – for every part of the recruitment process, the landscape is constantly evolving and new, innovative solutions pop up at any given moment. The recruitment technology market is estimated to [grow by $754.45 million](https://www.technavio.com/report/global-recruitment-software-market-industry-analysis) by 2023. The below image gives an idea about the variety in recruiting software but since the market is massive, this still only scratches the surface.

* 1. **PROJECT SCOPE**

To develop a Recruitment Application that helps the HR to conduct recruitment process and assign position to hired candidates Scope of this application is wide. This application automate the recruitment process and make it easy and simple. This application remove the manual creating multiple sheets which consume lot of time and hard work. Recruitment application provides HR to access the list of candidate and able to select the candidates that fulfil the requirement of the organization. This application provides the auto mail generation to candidates. This application keep the record of each step of selection process the list of candidate who are 5 selected in each step. HR can assign positions to selected candidates, their salary, location, duration and job description. This application provides HR to track the candidate detail before and after the recruitment HR can send them email about location, job profile, salary details and new updates about company. Every organization wants to sustain in this competitive world. To remain top there should be right manpower there to grow and to increase its productivity. So there is a need of application which is used to select manpower from the available talent pool. Hence there is huge scope of Recruitment application.

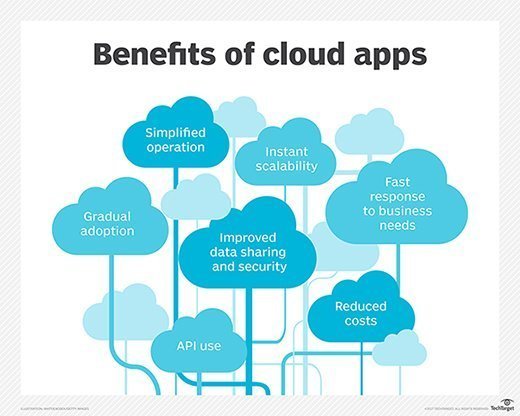
* 1. **Hardware/Software used in Project**

As this is a cloud based application so there is no need of sophisticated hardware only there is a need of Browser application and Internet connection. Cloud provider provides the development infrastructure to create application.

A cloud application, or cloud app, is a software program where cloud-based and local components work together. This model relies on remote [servers](https://whatis.techtarget.com/definition/server) for processing logic that is accessed through a web browser with a continual internet connection.

Cloud application servers typically are located in a remote [data center](https://searchdatacenter.techtarget.com/definition/data-center) operated by a [third-party](https://whatis.techtarget.com/definition/third-party) cloud services infrastructure provider. Cloud-based application tasks may encompass email, file storage and sharing, order entry, [inventory management](https://searcherp.techtarget.com/definition/inventory-management), word processing, customer relationship management (CRM), data collection, or financial accounting features.

* **Fast response to business needs.** Cloud applications can be updated, tested and deployed quickly, providing enterprises with fast time to market and agility. This speed can lead to culture shifts in business operations.
* **Simplified operation.** Infrastructure management can be outsourced to third-party cloud providers.
* **Instant scalability.** As demand rises or falls, available capacity can be adjusted.
* **API use.** Third-party data sources and storage services can be accessed with an [application programming interface (API)](https://searchapparchitecture.techtarget.com/definition/application-program-interface-API). Cloud applications can be kept smaller by using APIs to hand data to applications or API-based back-end services for processing or analytics computations, with the results handed back to the cloud application. Vetted APIs impose passive consistency that can speed development and yield predictable results.
* **Gradual adoption.** [Refactoring](https://searchapparchitecture.techtarget.com/definition/refactoring) legacy, on-premises applications to a cloud architecture in steps, allows components to be implemented on a gradual basis.
* **Reduced costs.** The size and scale of data centers run by major cloud infrastructure and service providers, along with competition among providers, has led to lower prices. Cloud-based applications can be less expensive to operate and maintain than equivalents on-premises installation.
* **Improved data sharing and security.** Data stored on cloud services is instantly available to authorized users. Due to their massive scale, cloud providers can hire world-class security experts and implement infrastructure security measures that typically only large enterprises can obtain. Centralized data managed by IT operations personnel is more easily backed up on a regular schedule and restored should [disaster recovery](https://searchdisasterrecovery.techtarget.com/definition/disaster-recovery) become necessary. cloud application, or cloud app, is a software program where cloud-based and local components work together. This model relies on remote [servers](https://whatis.techtarget.com/definition/server) for processing logic that is accessed through a web browser with a continual internet connection. Cloud application servers typically are located in a remote [data center](https://searchdatacenter.techtarget.com/definition/data-center) operated by a [third-party](https://whatis.techtarget.com/definition/third-party) cloud services infrastructure provider. Cloud-based application tasks may encompass email, file storage and sharing, order entry, [inventory management](https://searcherp.techtarget.com/definition/inventory-management), word processing, customer relationship management (CRM), data collection, or financial accounting features.
* **Fast response to business needs:** Cloud applications can be updated, tested and deployed quickly, providing enterprises with fast time to market and agility. This speed can lead to culture shifts in business operations.
* **Simplified operation:** Infrastructure management can be outsourced to third-party cloud providers.
* **Instant scalability:** As demand rises or falls, available capacity can be adjusted.
* **API use:** Third-party data sources and storage services can be accessed with an [application programming interface (API)](https://searchapparchitecture.techtarget.com/definition/application-program-interface-API). Cloud applications can be kept smaller by using APIs to hand data to applications or API-based back-end services for processing or analytics computations, with the results handed back to the cloud application. Vetted APIs impose passive consistency that can speed development and yield predictable results.
* **Gradual adoption:** [Refactoring](https://searchapparchitecture.techtarget.com/definition/refactoring) legacy, on-premises applications to a cloud architecture in steps, allows components to be implemented on a gradual basis.
* **Reduced costs:** The size and scale of data centers run by major cloud infrastructure and service providers, along with competition among providers, has led to lower prices. Cloud-based applications can be less expensive to operate and maintain than equivalents on-premises installation.
* **Improved data sharing and security:** Data stored on cloud services is instantly available to authorized users. Due to their massive scale, cloud providers can hire world-class security experts and implement infrastructure security measures that typically only large enterprises can obtain. Centralized data managed by IT operations personnel is more easily backed up on a regular schedule and restored should [disaster recovery](https://searchdisasterrecovery.techtarget.com/definition/disaster-recovery) become necessary.



**Figure 1 How cloud apps work**

Data is stored and compute cycles occur in a remote data center typically operated by a third-party company. A back end ensures uptime, security and integration and supports multiple access methods.

Cloud applications provide quick responsiveness and don't need to permanently reside on the local device. They can function offline, but can be updated online.

While under constant control, cloud applications don't always consume storage space on a computer or communications device. Assuming a reasonably fast internet connection, a well-written cloud application offers all the [interactivity](https://whatis.techtarget.com/definition/interactivity) of a desktop application, along with the [portability](https://searchstorage.techtarget.com/definition/portability) of a web application.

With the advancement of remote computing technology, clear lines between cloud and web applications have blurred. The term *cloud application* has gained great cachet, sometimes leading application vendors with any online aspect to brand them as cloud applications.

Cloud and web applications access data residing on distant storage. Both use server processing power that may be located on premises or in a distant data center.

A key difference between cloud and web applications is architecture. A web application or web-based application must have a continuous internet connection to function. Conversely, a cloud application or cloud-based application performs processing tasks on a local computer or workstation. An internet connection is required primarily for downloading or uploading data.

A web application is unusable if the remote server is unavailable. If the remote server becomes unavailable in a cloud application, the software installed on the local user device can still operate, although it cannot upload and download data until service at the remote server is restored.

The difference between cloud and web applications can be illustrated with two common productivity tools, email and word processing. [Gmail](https://whatis.techtarget.com/definition/Gmail), for example, is a web application that requires only a browser and internet connection. Through the browser, it's possible to open, write and organize messages using search and sort capabilities. All processing logic occurs on the servers of the service provider (Google, in this example) via either the internet's [HTTP](https://whatis.techtarget.com/definition/HTTP-Hypertext-Transfer-Protocol) or [HTTPS](https://searchsoftwarequality.techtarget.com/definition/HTTPS) protocols.

A CRM application accessed through a browser under a fee-based [software as a service (SaaS)](https://searchcloudcomputing.techtarget.com/definition/Software-as-a-Service) arrangement is a web application. Online banking and daily crossword puzzles are also considered web applications that don't install software locally.

An example of a word-processing cloud application that is installed on a workstation is Word's [Microsoft Office 365](https://searchenterprisedesktop.techtarget.com/definition/Microsoft-Office-365-suite). The application performs tasks locally on a machine without an internet connection. The cloud aspect comes into play when users save work to an Office 365 cloud server.

**CHAPTER 2**

**LITERATURE REVIEW**

**Increasing resilience by creating an adaptive salesforce**

During disruptions such as the COVID-19 pandemic, the resilience of any commercial organization becomes a critical characteristic. This paper examines the flexibility of the sales process—that is, adaptive selling—as an analog to resiliency, and recommends that firms create an adaptive salesforce to increase resilience. By examining the extant research and conducting interviews with sales leaders, the findings of this paper suggest that an adaptive salesforce should focus on three areas of change within team structures. The first is an increase in flexibility and adaptiveness for the functions that the salesforce performs. The second area is an improvement of the adaptiveness of scale in which sales functions can be rapidly in sourced or outsourced. The final area is technology adaptiveness, in which the salesforce adopts the use of technologies that are most relevant to customers. Implications for the firm, sales function, and individual salespeople are also derived.[1]

**Regaining control of salesforce**

This study develops a theory explaining how an employer can regain control over its salespeople. We posit that two forms of salesforce opportunism – shirking and influence activities – are the key sources of control loss. We theorize that an employer can regain control through a selective match of a formal control mechanism with a salesforce opportunistic behavior.

We test our predictions using data from 304 South Korean automobile dealers. We found that greater output control mitigates control loss when it is matched with salespeople's shirking, whereas it aggravates control loss when it is matched with salespeople's influence activities. Conversely, greater behavior control mitigates control loss when it is matched with salespeople's influence activities, whereas it aggravates control loss when it is matched with salespeople's shirking[2]

**APEX-MODFLOW: A New integrated model to simulate hydrological processes in watershed systems**

APEX (Agricultural Policy/Environmental eXtender) is an oft-used agro ecosystem model but has limited use in groundwater-driven watersheds due to a simplistic representation of groundwater processes. This paper presents the linkage of APEX and the [groundwater flow](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/groundwater-flow) model MODFLOW into a single modeling code. The mapping of recharge, groundwater head, and groundwater-surface water interactions are handled internally via subroutines[3]

**Minor obstructions for apex-pseudo forests**

A graph is called a pseudo forest if none of its connected components contains more than one cycle. A graph is an apex-pseudo forest if it can become a pseudo forest by removing one of its vertices. We identify 33 graphs that form the minor obstruction set of the class of apex-pseudo forests, i.e., the set of all minor-minimal graphs that are not apex-pseudo forests.[4]

**Salesforce management factors for successful new product launch**

New innovative products constitute a central source of economic value creation, but in many industries, salesforce management significantly conditions the appropriation of innovative products during their launch. Very little previous research addresses the salesforce management factors that contribute to successful new product launches. This study identifies and examines a set of salesforce management factors that contribute to successful new product launches by drawing on previous studies related to new product launches and salesforce management.

The multivariate analysis in this study uses data covering new product launches in the Swedish pharmaceutical industry. This study unearths a complex and unique complementarity pattern of factors resting upon the duality of a highly dynamic marketplace and sales representatives with an innovative personality type, which are complementary with other specific factors such as training, management control, and reward system[5]

**SQL**

Recently, attention has been focused on spatial databases, which combine conventional and spatially related data, such as geographic information systems, CAD/CAM, or VLSI. A language has been developed to query such spatial databases. It recognizes the significantly different requirements of spatial data handling and overcomes the inherent problems of the application of conventional database query languages. The spatial query language has been designed as a minimal extension to the interrogative part of SQL and distinguishes from previously designed SQL extensions by: the preservation of SQL concepts; the high-level treatment of spatial objects; and the incorporation of spatial operations and relationships. It consists of two components, a query language to describe what information to retrieve and a presentation language to specify how to display query results. Users can ask standard SQL queries to retrieve nonspatial data based on nonspatial constraints[6]

**Preference SQL — Design, Implementation, Experiences**

When searching for items to be purchased over the Internet, customer wishes and preferences are becoming increasingly important. Current search engines can hardly cope adequately with fuzzy predicates defined by complex preferences. The biggest problem of search engines implemented with standard SQL is that SQL does not directly understand the notion of preferences

Preference SQL extends SQL by a preference model based on strict partial orders, where preference queries behave like soft selection constraints. Several built-in base preference types and the powerful Pareto operator, combined with the adherence to declarative SQL programming style, guarantees great programming productivity.

The Preference SQL optimizer does an efficient rewriting into standard SQL, including a high-level implementation of the skyline operator for Pareto-optimal sets. This pre-processor approach enables seamless application integration, making Preference SQL available on all major SQL platforms. Several commercial Business-to-Customer (B2C) portals are powered by Preference SQL[7]

**CHAPTER 3**

**FEASIBILITY STUDY**

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and [technically feasible](https://en.wikipedia.org/wiki/Feasibility_study#Technical_feasibility) as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn’t profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

## ****Types of Feasibility Study****

A feasibility analysis evaluates the project’s potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are four types of feasibility study—separate areas that a feasibility study examines, described below.

**3.1Economical Feasibility**

System is economical feasible and can be easily implement with minimum hardware and software resources as this is a cloud based application platform is provided by cloud provider only there is a need of Internet connection and a Browser application. It is very important for designer to first analyze the system economically and determines that project is economical feasible or not. Costs and benefits of the proposed computer system must always be considered together, because they are interrelated and often interdependent. Although the systems analyst is trying to propose a system that fulfills various information requirements, decisions to continue with the proposed system will be based on a cost-benefit analysis, not on information requirements. In many ways, benefits are measured by costs, as becomes apparent in the next section.

Systems analysts are required to predict certain key variables before the proposal is submitted to the client. To some degree, a systems analyst will rely on a what-if analysis, such as, “What if labor costs rise only 5 percent per year for the next three years, rather than 10 percent?” The systems analyst should realize, however, that he or she cannot rely on what-if analysis for everything if the proposal is to be credible, meaningful, and valuable.

The systems analyst has many forecasting models available. The main condition for choosing a model is the availability of historical data. If they are unavailable, the analyst must turn to one of the judgment methods: estimates from the sales force, surveys to estimate customer demand, Delphi studies (a consensus forecast developed independently by a group of experts through a series of iterations), creating scenarios, or drawing historical analogies.

If historical data are available, the next differentiation between classes of techniques involves whether the forecast is conditional or unconditional. Conditional implies that there is an association among variables in the model or that such a causal relationship exists. Common methods in this group include correlation, regression, leading indicators, econometrics, and input/output models.

Unconditional forecasting means the analyst isn’t required to find or identify any causal relationships. Consequently, systems analysts find that these methods are low-cost, easy-to-implement alternatives. Included in this group are graphical judgment, moving averages, and analysis of time-series data. Because these methods are simple, reliable, and cost effective, the remainder of the section focuses on them.

### Estimation of Trends

Trends can be estimated in a number of different ways. One way to estimate trends is to use a moving average. This method is useful because some seasonal, cyclical, or random patterns may be smoothed, leaving the trend pattern. The principle behind moving averages is to calculate the arithmetic mean of data from a fixed number of periods; a three-month moving average is simply the average of the last three months. For example, the average sales for January, February, and March is used to predict the sales for April. Then the average sales for February, March, and April are used to predict the sales for May, and so on.

When the results are graphed, it is easily noticeable that the widely fluctuating data are smoothed. The moving average method is useful for its smoothing ability, but at the same time it has many disadvantages. Moving averages are more strongly affected by extreme values than by using graphical judgment or estimating using other methods such as least squares. The analyst should learn forecasting well, as it often provides information valuable in justifying the entire project.

### Tangible Costs

The concepts of tangible and intangible costs present a conceptual parallel to the tangible and intangible benefits discussed already. Tangible costs are those that can be accurately projected by the systems analyst and the business’s accounting personnel.

Included in tangible costs are the cost of equipment such as computers and terminals, the cost of resources, the cost of systems analysts’ time, the cost of programmers’ time, and other employees salaries. These costs are usually well established or can be discovered quite easily, and are the costs that will require a cash outlay of the business.

### Intangible Costs

Intangible costs are difficult to estimate and may not be known. They include losing a competitive edge, losing the reputation for being first with an innovation or the leader in a field, declining company image due to increased customer dissatisfaction, and ineffective decision making due to untimely or inaccessible information. As you can imagine, it is next to impossible to project a dollar amount for intangible costs accurately. To aid decision makers who want to weigh the proposed system and all its implications, you must include intangible costs even though they are not quantifiable.

**3.2 Technical Feasibility**

It is the study of the function performance and constraints that may affect the ability to achieve an acceptable system. The project development requires designer to have technical knowledge of salesforce.com for both application development and database system. Technical feasibility is one of the most important criteria for selecting material for digitisation. The physical characteristics of source material and the project goals for capturing, presenting and storing the [digital surrogates](https://www.sciencedirect.com/topics/computer-science/digital-surrogate) dictate the technical requirements. Libraries must evaluate those requirements for each project and determine whether they can be met with the resources available. If the existing staff, hardware and software resources cannot meet the requirements, then the project will need funding to upgrade equipment or hire an outside conversion agency. If these resources are not available, or if the technology does not exist to meet the requirements, then it is not technically feasible to digitise that material.

Considerations for technical feasibility include:

* **Image capture:** Image capture requires equipment, such as a scanner or a digital camera. Different types of material require different equipment, and different equipment produces images of differing quality. When selecting materials for digitising, technical questions that need to be addressed include: does the original source material require high resolution to capture? Are there any oversized items in the collection? Are there any bound volumes in the collection? What critical features of the source material must be captured in the digital product? In what condition are the source materials? Will they be damaged by the [digitisation process](https://www.sciencedirect.com/topics/computer-science/digitisation-process)?
* **Presentation**: Presentation refers to how the [digitised materials](https://www.sciencedirect.com/topics/computer-science/digitized-material) will be displayed online. Consider the following questions to determine the technical feasibility of presenting the digitised material:

Will the materials display well digitally?

How will users use the digital versions?

How will users navigate within and among digital collections?

Do the institutionally supported platforms and networked environment have the capability for accessing the images and delivering them with reasonable speed to the target audience?

Do the images need to be restricted to a specified community?

Do the images need special display features such as zooming, panning and page turning?

* **Description:** Some archival and special collections have been catalogued for public use and contain detailed finding aids with descriptions about each item and the collection as a whole. Other collections may not have been reviewed and documented in detail and do not have much information on individual items. Those collections will require more time, human resources and significant additional expense to research the materials, check the accuracy of the information obtained, and write appropriate descriptions to aid in discovery and use of the digital items. Typewritten documents, like the Drew Pearson columns described above, can have reasonably accurate OCR applied to them to replace, for some uses, the detailed descriptions required for discovery of hand-written or picture materials. The selection criteria should clearly state whether the items and collections that do not contain descriptions should be considered for digitisation.
* **Human resources**: When selecting materials for digitisation, the library should consider whether it has the staff and skill sets to support the digitisation, metadata entry, user interface design, programming and search engine configuration that is required for the project to implement the desired functionality. For large collaborative projects, dedicated staff are usually required from each partner. Digital collections also require long-term maintenance, which needs to be considered and planned for. If a project does not have the necessary staff and skills in-house, but funding is available, outsourcing may be a good choice.

**3.3 Behavioural Feasibility**

In the application domain our system works as an application. There are simple form to fill and service requires no ambiguous entries, all the behavioural entries are simple and GUI based. The system design is very user friendly, interactive. The application should be used by Administrator. People are inherently resistant to change, and computers have been known to facilitate change. An estimate should be made of how strong a reaction the user staff is likely to have toward the development of a computerized system. [t is common knowledge that computer installations have something to do with turnover, transfers, retraining, and changes in employee job status. Therefore, it is understandable that the introduction of a candidate system requires special effort to educate, sell, and train the business.  
  
In our safe deposit example, three employees are more than 50 years old and have been with the bank over 14 years, four years of which have been in safe deposit. The remaining two employees are in their early thirties. They joined safe deposit about two years before the study. Based on data gathered from extensive interviews, the younger employees want the programmable aspects of safe deposit (essentially billing) put on a computer. Two of the three older employees have voiced resistance to the idea. Their view is that billing is no problem. The main emphasis is customer service-personal contacts with customers. The decision in this case was to go ahead and pursue the project.

* 1. **Operational Feasibility**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.[[10]](https://en.wikipedia.org/wiki/Feasibility_study#cite_note-SAD-Global_Enterprise-10)

The operational feasibility assessment focuses on the degree to which the proposed development project fits in with the existing business environment and objectives with regard to development schedule, delivery date, [corporate culture](https://en.wikipedia.org/wiki/Corporate_culture) and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviour are to be realised. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases

An example of an operational feasibility study, or the fourth type, analyzes the inside operations on how a deemed process will work, be implemented, and how to deal with change resistance and acceptance.

Operational feasibility studies are generally utilized to answer the following questions:

* **Process** – How do the end-users feel about a new process that may be implemented?
* **In-House Strategies** – How will the work environment be affected? How much will it change?
* **Adapt & Review** – Once change resistance is overcome, explain how the new process will be implemented along with a review process to monitor the process change.

If an operational feasibility study must answer the six items above, how is it used in the real world? A good example might be if a company has determined that it needs to totally redesign the workspace environment.

After analyzing the technical, economic, and scheduling feasibility studies, next would come the operational analysis. In order to determine if the redesign of the workspace environment would work, an example of an operational feasibility study would follow this path based on six elements:

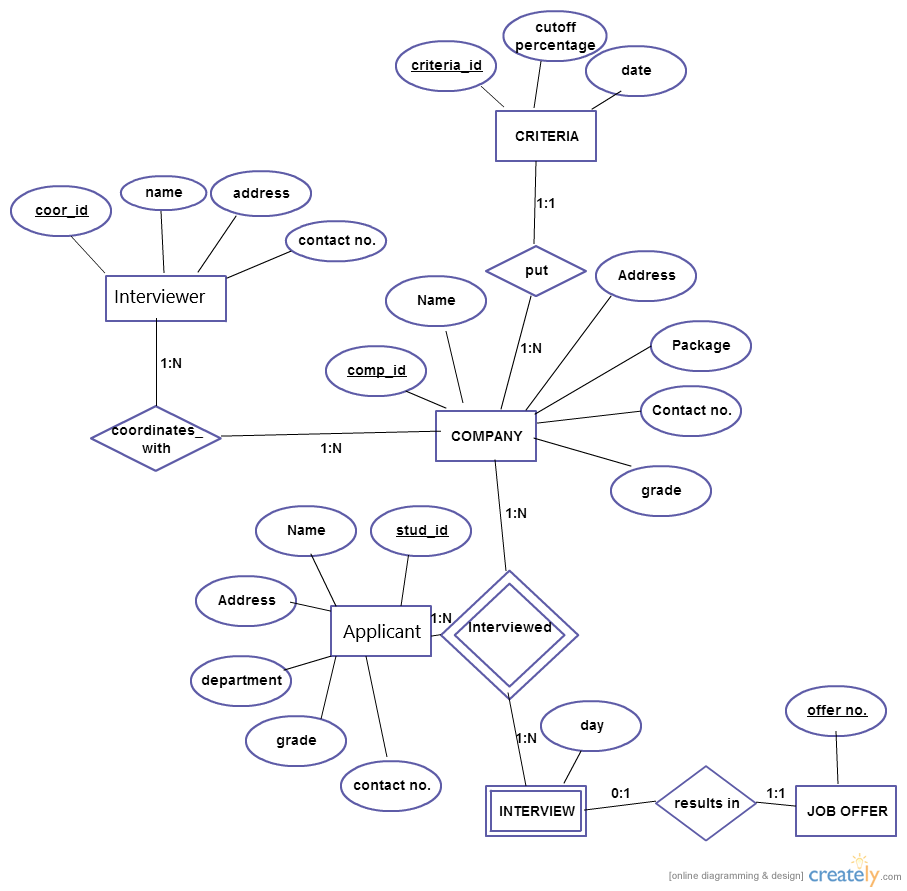
* **Process** – Input and analysis from everyone the new redesign will affect along with a data matrix on ideas and suggestions from the original plans.
* **Evaluation** – Determinations from the process suggestions; will the redesign benefit everyone? Who is left behind? Who feels threatened?
* **Implementation** – Identify resources both inside and out that will work on the redesign. How will the redesign construction interfere with current work?
* **Resistance** – What areas and individuals will be most resistant? Develop a [change resistance plan](https://www.brighthubpm.com/change-management/15306-change-management-methodology-for-beginners/).
* **Strategies** – How will the organization deal with the changed workspace environment? Do new processes or structures need to be reviewed or implemented in order for the redesign to be effective?
* **Adapt & Review** – How much time does the organization need to adapt to the new redesign? How will it be reviewed and monitored? What will happen if through a monitoring process, additional changes must be made?

The most important part of operational feasibility study is input—from everyone, especially when it affects how or what an organization does as far as processes. If the process were to build a new sports arena for a client, then a study determining how the arena will operate in a way that is conducive to its inhabitants, parking, human flow, accessibility and other elements is a good example of an operational feasibility study.

Create a [sample operational feasibility study](https://www.brighthubpm.com/project-planning/63692-project-feasibility-study-samples/) if you plan to change something inside the company that will affect how the organization runs or when a client asks you to explore a new product or process that will affect elements within their own organization.

**CHAPTER 4**

**System Design**

**4.1 ER-Diagram**

**Figure 2 ER-DIAGRAM**

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in a table. ER diagrams represent the logical structure of databases. ER Diagram represent relationship between two database tables.

E-R diagram means Entity Relationship diagram. Entity is a object of system, generally we refer entity as database table , the e-r diagram represent the relationship between each table of database. E-R diagram represent entity with attributes, attributes is a properties of entity. If we assume entity is a database table then all the columns of table are treat as attributes.

Here are the geometric shapes and their meaning in an E-R Diagram. We will discuss these terms in detail in the next section(Components of a ER Diagram) of this guide so don’t worry too much about these terms now, just go through them once.

**Rectangle**: Represents Entity sets.  
**Ellipses**: Attributes  
**Diamonds**: Relationship Set  
**Lines**: They link attributes to Entity Sets and Entity sets to Relationship Set  
**Double Ellipses:** Multivalued Attributes  
**Dashed Ellipses**: Derived Attributes  
**Double Rectangles**: Weak Entity Sets  
**Double Lines**: Total participation of an entity in a relationship set

The User applies to jobs and the company posts jobs and the user has applied to certain jobs. The tables are represented in the form of rectangular boxes as shown in the diagram. The primary keys are underlined. The relationships are shown through diamond boxes. The cardinalities are mentioned through the numbers.

* 1. **Database Design**
* **A FORCE.COM DATABASE**

Database technology is the persistence layer at the heart of all data­ centric applications, the tier that’s in charge of organizing, protecting, and managing shared database access reliably, securely, efficiently. The persistence layer underlying Force.com (and Database.com) is proven database technology that powers all of salesforce.com’s products today, serving more than 100,000 organizations, 135,000 applications, 3 million users, 1 billion transactions per day with an average request response time of less than 300ms, all with an average up time of 99.9+ percent.

* **FEATURES OF FORCE.COM DATABASE**

EASY TO USE With Force.com, there’s nothing to manage — salesforce.com takes care of everything for you. There’s no software to install, update, and patch. No waiting on someone else when you want to provision databases. No worries about database backup and disaster recovery. No complex documentation set with thousands of pages and parameters to tune for performance or elasticity. There’s even automatic indexing. Whether you have 1 database or 1,000 databases, all you need to focus on is building great apps.

TRUSTWORTHY Force.com is built with the security and privacy of customer information in mind. Salesforce.com’s infrastructure and corporate workplace meet all of the highest industry standards, including SAS 70 Type II, Sys Trust, and ISO 27001 certifications.

MODERN Force.com is more than just another database system — it’s jam­packed with next­ generation features that make building and maintaining highly functional, secure, social, and mobile apps a snap.

● Force.com users, profiles, roles, groups, and row­level sharing rules help you build secure apps without the need to code, test, and maintain your own complicated security logic.

● With Force.com, it’s easy to implement common application logic without writing complicated and error­ prone code. Such features include declarative, 36 point­ and­ click configuration for work flows, encrypted/masked fields, validation rules, formula fields, roll­up summary fields, and cross­ object validation rules.

● Force.com is "social" because it includes the Salesforce Chatter API, a built ­in data model apps can leverage to become instantly social and collaborative.

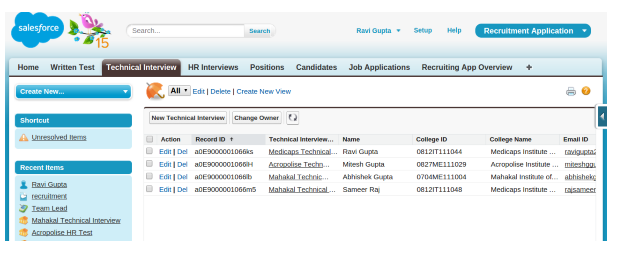
● Force.com’s REST APIs, OAuth implementation for user authentication/authorization, data feeds, custom Web services, embedded security model, and other features make it a perfect fit for easily building secure, scalable mobile apps, either native or HTML5.

OPEN Force.com’s full complement of open APIs lets you build and integrate applications using the approach of your choice. REST­ and SOAP ­based APIs are standards­ based APIs that make Force.com open to whatever programming language you want to use. Using various APIs, your applications can do many things such as create ­read ­update ­delete (CRUD) business data, load a large number of records asynchronously, and take advantage of the Chatter API to provide collaboration and social networking capabilities to any application.

POWERFUL Most modern apps use server­ side logic to centralize complex business logic and enforce complex data integrity rules. Apex, with syntax much like Java, is Force.com’s procedural language that you can use to create server­ side logic for an application. For example, Apex lets you create stored procedures that modify the database within the context of ACID transactions, and expose them as a custom Web services API (RESTful or SOAP) for your apps. You can also use Apex to build database triggers, routines that automatically fire (execute) when apps modify records in your database.

7.3 DATABASE DESIGN IN RECRUITMENT APPLICATION Salesforce provide force.com database to store, manage and retrieve data. In Recruitment Application we used force.com database to store candidate details, positions, job level, salary. The tables we have designed are as follows.

**Interview Table**

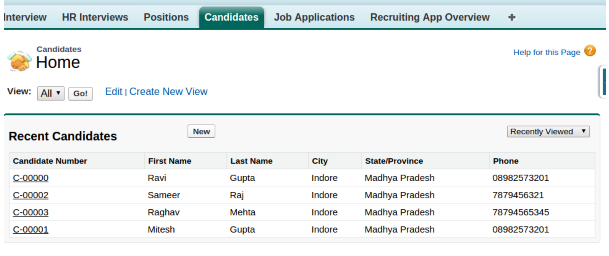
****

**Figure 3 Interview Table**

Interview Table consists details of candidate who have applied-

* Record Id – generates when a user applied for a job
* Technical Interview- Interview type or name
* Name- Name of the applicant
* College Id- Class or University Roll no.
* College Name- College name from where graduated or pursuing
* Email- Email id of the applicant

**Candidate Table**

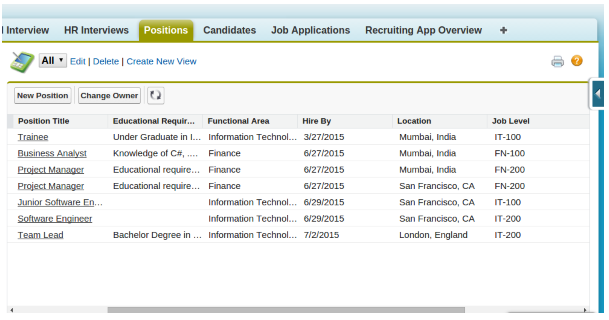
****

**Figure 4 Candidate Table**

Candidate Table consists details of user profile-

* Candidate No. – auto generate no. when a user created a profile
* First Name- First name of the user.
* Last Name- Last name of the user.
* City- where’s the user belongs.
* State- State in which user lived.
* Phone- Phone no. of the user.

**Position Table**

****

**Figure 5 POSITION TABLE PAGE**

Position Table consists details of job profile-

* Position Title- Name of the position.
* Educational Requirement- Profile requirement.
* Functional Area- Job field.
* Hire by- Date when job released.
* Location- Location of Job.
* Job Level- a level of the job profile.

**4.2DFD and Flow Graph**

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

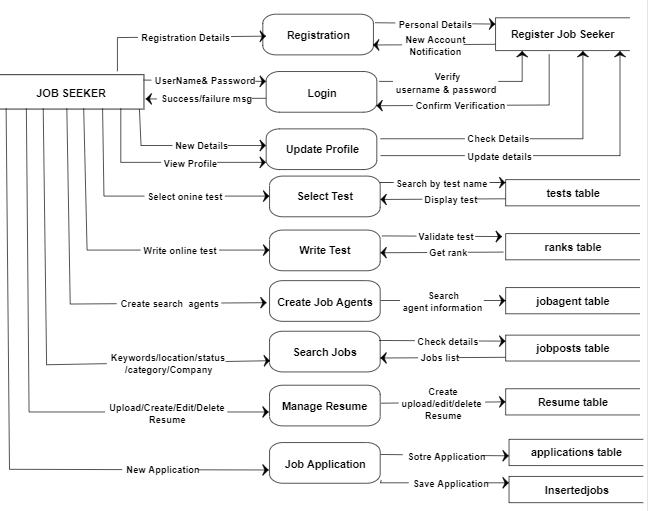
Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

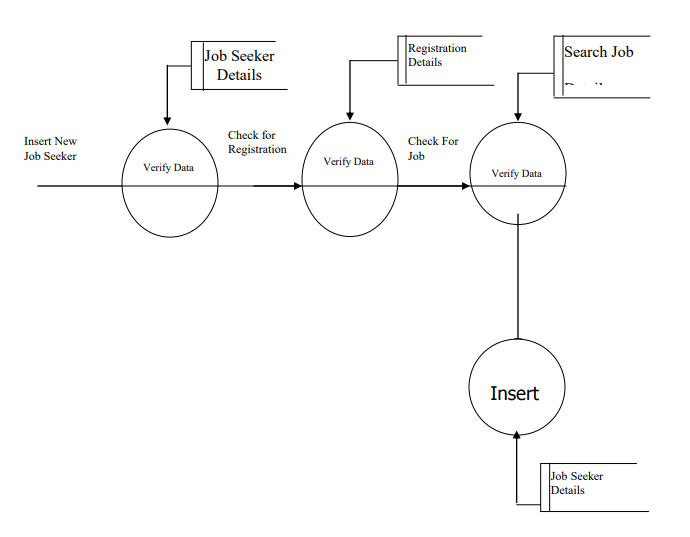
* Logical information flow of the system
* Determination of physical system construction requirements
* Simplicity of notation
* Establishment of manual and automated systems requirements

### Process

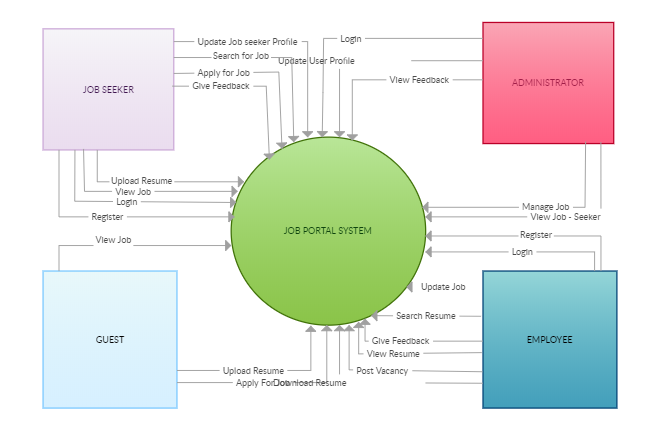
A process receives input data and produces output with a different content or form. Processes can be as simple as collecting input data and saving in the database, or it can be complex as producing a report containing monthly sales of all retail stores in the northwest region.

****

**Figure 6 Job Seeker Data Flow**

****

**Figure 7Inserting Job Seeker Details**

****

Recruitment App Portal

**Recruitment App Data Flow**

FLOW OF SEQUENCE

● This application starts with Registration of HR in which HR enters username and password and other details.

● If HR registered then he/she can login to the system by providing username and password.

● HR store the details of candidate, their college id, college name, email­Id, city, state etc.

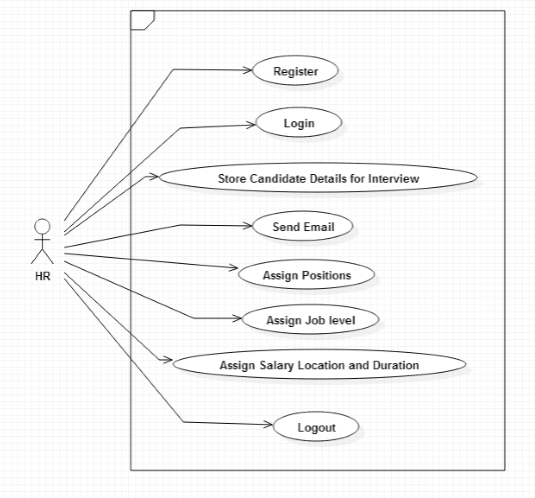
● HR send Email alerts to list of selected candidates in each round.

● HR can see the applicant details and status and send confirmation or rejection letter.

● HR can assign positions to selected candidates, their salary, location, duration and job description.

● HR can assign job level, responsibility and detail description of job.

● At the end HR Logout from the system as no one can change the content.

****

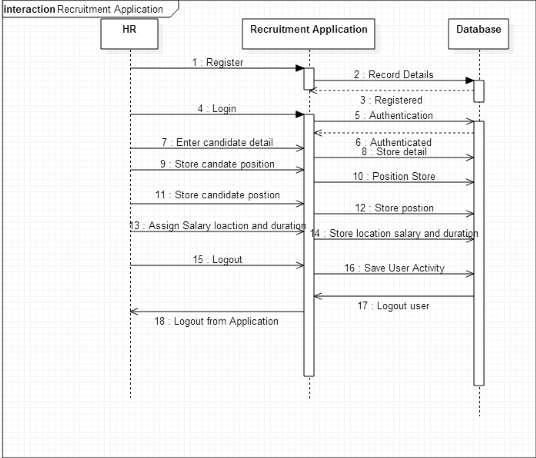
**Figure 8Recruitment App Data Flow**

HR can Register, login and logout themselves.

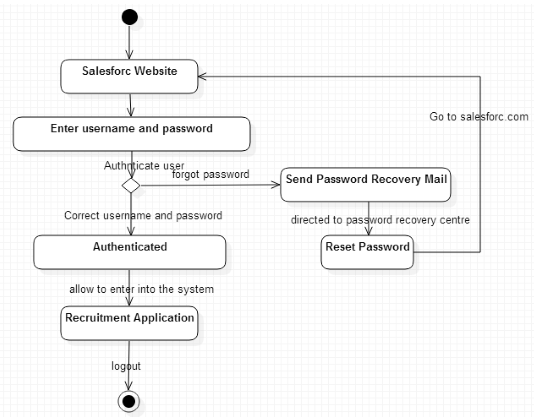
HR store Candidate Details for Interview and also update.

HR can send Email, assign position and job level.

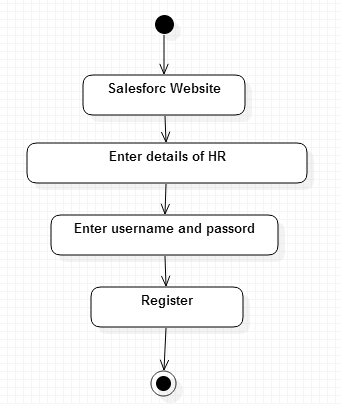
HR can assign Salary, Location and Duration.

****

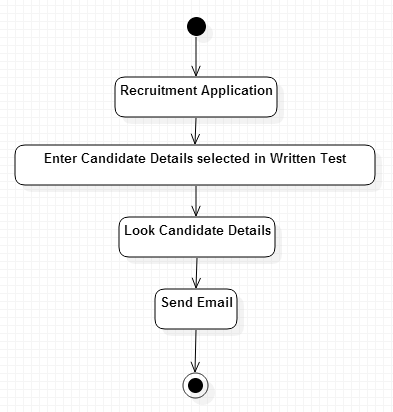
**Figure 9 HR roles with application and Database**

****

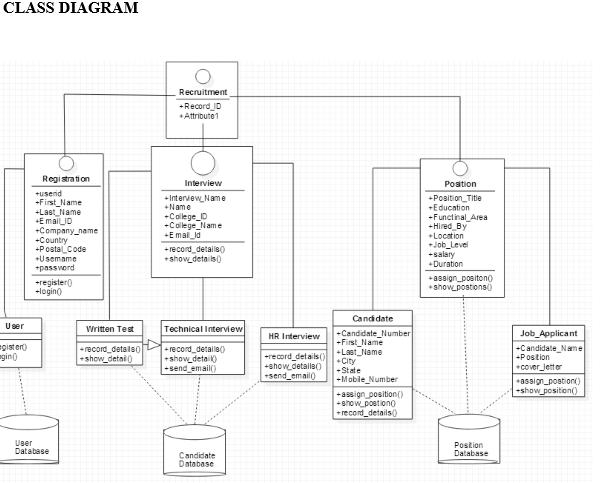
**Figure 10 Recruitment application Authentication flow Salesforce**

****

**Figure 11 HR Authentication flow**

****

**Figure 12 Candidate Selection Informing Flow**

****

**Figure 13 Database Table Information**

Recruitment application is developed for HR to conduct recruitment process and assign the positions to selected candidates. This application provides HR to store and manage data of objects and conduct the interview process very easily and smoothly. There is no need to make extra sheets, to store the details of the candidates. This application provides you to store all the data of candidates on cloud storage. This application works on 5 layer model of cloud which are defined as follows:

1. Cloud Application Layer ­ Cloud application layer provides the most visible layer to the end users of the cloud. Recruitment application provides an user interface to interact with users.

2. Cloud Software Environment Layer ­ The cloud software environment layer(also dubbed the software platform layer). The users of this layer are cloud applications’ developers, implementing their applications for and deploying them on the cloud. In Recruitment application Force.com is an application development environment.

3. Cloud Software Infrastructure Layer ­ The cloud software infrastructure layer provides fundamental resources to other higher level layers. Cloud services offered in this layer can be categorized into: computational resources, data storage, and communications. In Recruitment application Salesforce manage in infrastructure of auto scaling and load balancing on basis of number of working users.

4. Software Kernel ­ This cloud layer provides the basic software management for the physical servers that compose the cloud. Software kernels at this level can be implemented as an OS kernel, hypervisor, and virtual machine monitor and/or clustering middleware.

5. Hardware and Firmware​ ­ The bottom layer of the cloud stack in our proposed is the actual physical hardware and switches that form the backbone of  the cloud. In this regard, users of this layer of the cloud are normally big enterprises with huge IT requirements in need of subleasing Hardware as a Service (HaaS).  Recruitment application is developed in Sales force cloud computing environment known as Force.com development environment. In Force.com environment you can develop and deploy applications. Salesforce provides an app store known as Apex change just like play store of Google Android Marketplace. Salesforce AppExchange is Salesforce.com's cloud computing marketplace through which end users can access, download and install software apps. You can download Recruitment App from Apex change. To download the  application first user need to register on salesforce. After Registration users are able to download  and   install  the  applications.

**CHAPTER 5**

**Form Designs(Screenshots)**

**5.1 Home Page**

Graphical user interface, website

Description automatically generated

**Figure 14 Home Page**

The First Page rendered when application open. Initially logged in as a guest. Guest has only permission to see all jobs but can’t apply.

**5.2Login Page**

Graphical user interface, website

Description automatically generated

**Figure 15 Login Page**

On clicking Top Right button(Login), Login Page rendered which Consists

* Username.
* Password
* Login Button
* Forget Your Password?
* Not a member?(Register)

**5.3Login Responsive**

Graphical user interface, application, website

Description automatically generated

**Figure 16 Login Responsive**

The Page is responsive also it can be rendered in Mobile.

**5.4Logged Page**

Graphical user interface, website

Description automatically generated

**Figure 17 Logged Page**

The Logged Page rendered when login credentials are correct.

This page consists Home page with Username, and Application tab.

User now can see all jobs and also apply.

**5.5Jobs Tab**

Graphical user interface, website

Description automatically generated

**Figure 18 Jobs Tab Page**

The Job Tab Page includes

* Job Title.
* Description.
* Start Date.
* End Date.
* Experience.
* Salary.
* Skills Required.
  1. **Job Description**

Graphical user interface, website

Description automatically generated

**Figure 19 Job Description**

On Clicking The particular job, full details of the job appears.

* 1. **Application**

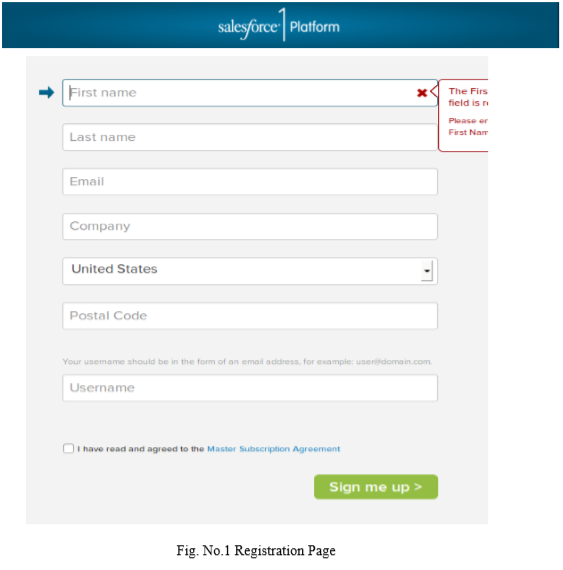
A screenshot of a computer

Description automatically generated

**Figure 20 Application**

On clicking New the user generate Application of a particular job.

* 1. **Salesforce Login**

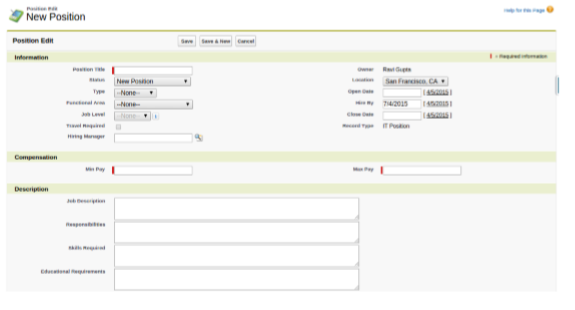
****

**Figure 21 Salesforce Login**

The Registration Page where you can register yourself to the application.

Salesforce Registration Page is linked to the application.

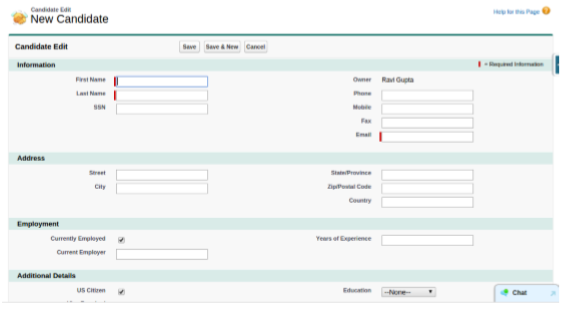
**4.9 New Position**

****

**Figure 22 New Position Page**

Admin can create New Position by clicking on New Position Tab.

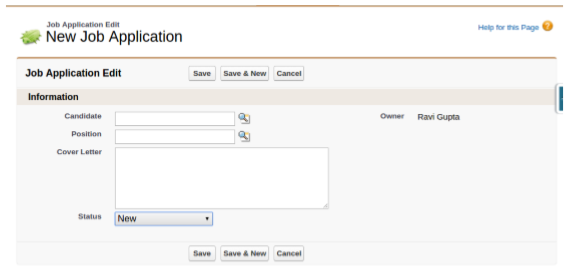
**5.10 New Candidate**

****

**Figure 23 New Candidate**

User can create profile on clicking New Candidate.

**5.11 New Job Application**

****

**Figure 24 New Job Application**

Admin can Edit Job Application status of a particular candidate.

**CHAPTER 6**

**Coding**

* **Login Class**

public class CustomLoginController {

public String username{get; set;}

public String password {get; set;}

public pageReference doLogin()

{

return Site.login(username,password,null);

}

}

* **Login Page**

<apex:page sidebar="false" controller="CustomLoginController" showheader="false" standardStylesheets="false">

<apex:form >

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<title>Untitled Document</title>

<style>

/\* Basics \*/

html, body {

width: 100%;

height: 100%;

font-family: "Helvetica Neue", Helvetica, sans-serif;

color: #444;

-webkit-font-smoothing: antialiased;

background: #f0f0f0;

}

#container {

position: fixed;

width: 340px;

height: 280px;

top: 50%;

left: 50%;

margin-top: -140px;

margin-left: -170px;

background: #fff;

border-radius: 3px;

border: 1px solid #ccc;

box-shadow: 0 1px 2px rgba(0, 0, 0, .1);

}

form {

margin: 0 auto;

margin-top: 20px;

}

label {

color: #555;

display: inline-block;

margin-left: 18px;

padding-top: 10px;

font-size: 14px;

}

p a {

font-size: 11px;

color: #aaa;

float: right;

margin-top: -13px;

margin-right: 20px;

-webkit-transition: all .4s ease;

-moz-transition: all .4s ease;

transition: all .4s ease;

}

p a:hover {

color: #555;

}

input {

font-family: "Helvetica Neue", Helvetica, sans-serif;

font-size: 12px;

outline: none;

}

input[type=text],

input[type=password] {

color: #777;

padding-left: 10px;

margin: 10px;

margin-top: 12px;

margin-left: 18px;

width: 290px;

height: 35px;

border: 1px solid #c7d0d2;

border-radius: 2px;

box-shadow: inset 0 1.5px 3px rgba(190, 190, 190, .4), 0 0 0 5px #f5f7f8;

-webkit-transition: all .4s ease;

-moz-transition: all .4s ease;

transition: all .4s ease;

}

input[type=text]:hover,

input[type=password]:hover {

border: 1px solid #b6bfc0;

box-shadow: inset 0 1.5px 3px rgba(190, 190, 190, .7), 0 0 0 5px #f5f7f8;

}

input[type=text]:focus,

input[type=password]:focus {

border: 1px solid #a8c9e4;

box-shadow: inset 0 1.5px 3px rgba(190, 190, 190, .4), 0 0 0 5px #e6f2f9;

}

#lower {

background: #ecf2f5;

width: 100%;

height: 69px;

margin-top: 20px;

box-shadow: inset 0 1px 1px #fff;

border-top: 1px solid #ccc;

border-bottom-right-radius: 3px;

border-bottom-left-radius: 3px;

}

input[type=checkbox] {

margin-left: 20px;

margin-top: 30px;

}

.check {

margin-left: 3px;

font-size: 11px;

color: #444;

text-shadow: 0 1px 0 #fff;

}

input[type=submit] {

float: right;

margin-right: 20px;

margin-top: 20px;

width: 80px;

height: 30px;

font-size: 14px;

font-weight: bold;

color: #fff;

background-color: #acd6ef; /\*IE fallback\*/

background-image: -webkit-gradient(linear, left top, left bottom, from(#acd6ef), to(#6ec2e8));

background-image: -moz-linear-gradient(top left 90deg, #acd6ef 0%, #6ec2e8 100%);

background-image: linear-gradient(top left 90deg, #acd6ef 0%, #6ec2e8 100%);

border-radius: 30px;

border: 1px solid #66add6;

box-shadow: 0 1px 2px rgba(0, 0, 0, .3), inset 0 1px 0 rgba(255, 255, 255, .5);

cursor: pointer;

}

input[type=submit]:hover {

background-image: -webkit-gradient(linear, left top, left bottom, from(#b6e2ff), to(#6ec2e8));

background-image: -moz-linear-gradient(top left 90deg, #b6e2ff 0%, #6ec2e8 100%);

background-image: linear-gradient(top left 90deg, #b6e2ff 0%, #6ec2e8 100%);

}

input[type=submit]:active {

background-image: -webkit-gradient(linear, left top, left bottom, from(#6ec2e8), to(#b6e2ff));

background-image: -moz-linear-gradient(top left 90deg, #6ec2e8 0%, #b6e2ff 100%);

background-image: linear-gradient(top left 90deg, #6ec2e8 0%, #b6e2ff 100%);

}

</style>

</head>

<body>

<!-- Begin Page Content -->

<div id="container">

<form>

<label for="username">Username:</label>

<apex:inputtext value="{!username}"/>

<label for="password">Password:</label>

<apex:inputsecret value="{!password}"/>

<div id="lower">

<apex:commandbutton value="Login" action="{!doLogin}"/>

</div><!--/ lower-->

</form>

</div><!--/ container-->

<!-- End Page Content -->

</body>

</apex:form>

</apex:page>

* **Community Login**

global with sharing class CommunitiesLoginController {

global CommunitiesLoginController () {}

// Code we will invoke on page load.

global PageReference forwardToAuthPage() {

String startUrl = System.currentPageReference().getParameters().get('startURL');

String displayType = System.currentPageReference().getParameters().get('display');

return Network.forwardToAuthPage(startUrl, displayType);

}

// Code we will invoke on page load.

global PageReference forwardToCustomAuthPage() {

return new PageReference( '/CustomLoginPage');

}

}

* **Visual Force Page**

<apex:page controller="LoginSignUpController">  
    <apex:form >  
    <apex:pageBlock >  
    <apex:pageBlockSection >  
         <apex:inputField id="name" value="{!object.User\_Name\_\_c}" required="true"/>  
        </apex:pageBlockSection>  
         <apex:pageBlockSection >  
        <apex:inputField id="age" value="{!object.Age\_\_c}"/>  
            </apex:pageBlockSection>  
          
    <apex:pageBlockSection >  
        <apex:pageBlockSectionItem >  
          
            <apex:outputLabel value="password" for="pass"/>  
         <apex:inputSecret id="pass" value="{!object.Password\_\_c}" required="true"/>       
            </apex:pageBlockSectionItem>  
      </apex:pageBlockSection>  
         
         <apex:pageBlockSection >  
        <apex:inputField id="gender" value="{!object.gender\_\_c}"/>  
         </apex:pageBlockSection>  
          
        <apex:pageBlockSection >  
        <apex:inputField id="Email" value="{!object.EmailId\_\_c}"/>  
         </apex:pageBlockSection>  
        
         
        <apex:pageBlockSection >  
        <apex:inputField id="birthday" value="{!object.Birthday\_\_c}"/>  
         </apex:pageBlockSection>  
          
        
        <apex:commandButton value="signup" action="{!signUp}"/>  
            
  
    </apex:pageBlock>  
        </apex:form>  
</apex:page>

* **Controller Class**

:  
public class LoginSignUpController {  
  public Object\_\_c object{get;set;} //Assume it to be ur custom user  
  
public LoginSignUpController()  
{  
    object=new Object\_\_c();  
      
}  
    public PageReference login()  
    {  
       if( (object.EmailId\_\_c == null) || (object.Password\_\_c == null))  
       {  
           return null;  
       }  
             
           List<Object\_\_c> objects= [select Id,\_\_EmailId\_\_c,\_\_Password\_\_c from Object\_\_c];  
        for(Object\_\_c loginObject:objects)  
        {  
            if((loginObject.\_\_EmailId\_\_c == object.\_\_EmailId\_\_c) && (loginObject.\_\_Password\_\_c == object.\_\_Password\_\_c))  
            {  
                PageReference page = new PageReference('/apex/UserStart');  
                page.getParameters().put('objectId', loginObject.Id);  
                page.setRedirect(true);  
                return page;  
            }  
        }                                 
        return null;  
    }  
      
    public PageReference signUp()  
    {  
       insert object;  
      PageReference pageRef = new PageReference(ApexPages.currentPage().getUrl());  
      pageRef.setRedirect(true);  
      return pageRef;  
   }  
}

* **Position Page**

<apex:page controller="PositionsController" showHeader="false" standardStylesheets="true">

<apex:composition template="JavelinTemplate" >

<apex:define name="body">

<apex:form >

<apex:sectionHeader title="Positions" />

<apex:pageBlock title="" id="pageBlock">

<apex:pageBlockButtons >

<apex:commandButton value="New" action="{!URLFOR($Action.Position\_\_c.New)}"/>

</apex:pageBlockButtons>

<apex:pageBlockTable value="{!positions}" var="pos"

rendered="{!NOT(ISNULL(positions))}">

<apex:column value="{!pos.Name}" />

<apex:column value="{!pos.Min\_Pay\_\_c}" />

<apex:column value="{!pos.Max\_Pay\_\_c}" />

<apex:column value="{!pos.Open\_Date\_\_c}" />

<apex:column >

<apex:facet name="header">Status</apex:facet>

<div style="background-color:{!IF(pos.End\_Date\_\_c > TODAY(),'#7CFC00', '#FFA07A')}">

{!IF(pos.End\_Date\_\_c > TODAY(),'Open', 'Closed')}

</div>

</apex:column>

</apex:pageBlockTable>

</apex:pageBlock>

</apex:form>

</apex:define>

</apex:composition>

</apex:page>

* **Template Home Page**

<apex:page showHeader="false" standardStylesheets="false">

<link rel="stylesheet" type="text/css" href="http://ev-qa.zs.local/JAM-Upgrade/0008-Cadmium/Web/so-styles.css?version=8569" />

<link rel="stylesheet" type="text/css" href="http://ev-qa.zs.local/JAM-Upgrade/IdentityManager/app/ApplicationRegistry/WebResources/styles/launchbar.css?v=4286" />

<div name="frameMenu" id="frameMenu">

<div id="launch-bar">

<div id="2\_LaunchBar">

<div id="lbr">

<div id="lbar-logout">

<a class="lbar-link" href="/secur/logout.jsp" id="lbl\_so">Sign Out</a>';

</div>

<div id="lbar-user" class="lbar-menu-off">

<a id="" class="" href="javascript:void(0);"><span class="user-label">User: </span><span class="user-name">{!$User.LastName}, {!$User.FirstName}</span></a>

</div>

</div>

</div>

</div>

<div id="mnuMain"><div id="header"></div>

<div id="motifCurve" style="">

<apex:tabpanel >

<apex:tab label="Tab1"></apex:tab>

<apex:tab label="tab2"></apex:tab>

</apex:tabpanel>

</div>

<apex:toolbar id="theToolbar" rendered="true">

<apex:outputText value="Sample Toolbar"/>

<apex:toolbarGroup itemSeparator="line" id="toobarGroupLinks">

<apex:outputLink value="http://www.salesforce.com">

salesforce

</apex:outputLink>

<apex:outputLink value="http://developer.salesforce.com">

apex developer network

</apex:outputLink>

</apex:toolbarGroup>

<apex:toolbarGroup itemSeparator="line" location="right" id="toobarGroupForm">

<apex:form id="theForm">

<apex:inputText id="theInputText">Enter Text</apex:inputText>

<apex:commandLink value="search" id="theCommandLink"/>

</apex:form>

</apex:toolbarGroup>

</apex:toolbar>

</div>

</div>

<apex:insert name="body"></apex:insert>

</apex:page>

* **Template table Page**

<apex:page showHeader="false" standardStylesheets="false" controller="PositionsController">

<apex:composition template="siteTemplate">

<apex:define name="body">

<apex:form >

<apex:sectionHeader title="Positions" />

<apex:pageBlock title="" id="pageBlock">

<apex:pageBlockTable value="{!positions}" var="pos"

rendered="{!NOT(ISNULL(positions))}">

<apex:column value="{!pos.Name}" />

<apex:column value="{!pos.Min\_Pay\_\_c}" />

<apex:column value="{!pos.Max\_Pay\_\_c}" />

<apex:column value="{!pos.Open\_Date\_\_c}" />

<apex:column >

<apex:facet name="header">Status</apex:facet>

<div style="background-color:{!IF(pos.End\_Date\_\_c > TODAY(),'#7CFC00', '#FFA07A')}">

{!IF(pos.End\_Date\_\_c > TODAY(),'Open', 'Closed')}

</div>

</apex:column>

</apex:pageBlockTable>

<apex:pageBlockButtons >

<apex:commandButton value="New" action="{!URLFOR($Action.Position\_\_c.New)}"/>

</apex:pageBlockButtons>

</apex:pageBlock>

</apex:form>

<apex:pageBlock title="Viewing Positions">

<apex:form id="theForm">

<apex:pageBlockSection >

<apex:dataList var="a" value="{!positions}" type="1">

{!a.name}

</apex:dataList>

</apex:pageBlockSection>

<apex:panelGrid columns="2">

<apex:commandLink action="{!previous}">Previous</apex:commandlink>

<apex:commandLink action="{!next}">Next</apex:commandlink>

</apex:panelGrid>

</apex:form>

</apex:pageBlock>

</apex:define>

</apex:composition>

</apex:page>

**->List of selected**

<apex:page controller="recruitsController">

<apex:messages />

<apex:pageBlock title="A Small Recruitment App">

<apex:pageBlockSection title="Add a new Candidate">

<apex:form >

<p><apex:outputLabel value="First Name" /> &nbsp;<apex:inputText value="{!FirstName}"/></p>

<p><apex:outputLabel value="Last Name" /> &nbsp;<apex:inputText value="{!LastName}"/></p>

<p><apex:outputLabel value="Position" /> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<apex:inputText value="{!Title}"/></p>

<p><apex:outputLabel value="Company" /> &nbsp;&nbsp;&nbsp;<apex:inputText value="{!Company}"/></p>

<apex:commandButton action="{!add}" value="Add Canditate" />

</apex:form>

</apex:pageBlockSection>

<apex:pageBlockSection title="List of Leads">

<apex:form >

<apex:pageBlockTable value="{!allContacts}" var="c" >

<apex:column value="{!c.id}" headervalue="ID"/>

<apex:column value="{!c.FirstName}" headervalue="First Name"/>

<apex:column value="{!c.LastName}" headervalue="Last Name" />

<apex:column value="{!c.Title}" headervalue="Title"/>

<apex:column value="{!c.Company}" headervalue="Company"/>

<apex:column >

<apex:commandButton action="{!addToRecruits}" value="Recruit" reRender="all">

<apex:param assignTo="{!leadID}" name="leadID" value="{!c.id}" />

</apex:commandButton>

</apex:column>

</apex:pageBlockTable>

</apex:form>

</apex:pageBlockSection>

<apex:pageBlockSection title="List of Recruits">

<apex:form >

<apex:pageBlockTable value="{!allRecruits}" var="r" >

<apex:column value="{!r.FirstName\_\_c}" headervalue="First Name"/>

<apex:column value="{!r.LastName\_\_c}" headervalue="Last Name"/>

</apex:pageBlockTable>

</apex:form>

</apex:pageBlockSection>

</apex:pageBlock>

</apex:page>

**CHAPTER 7**

**Testing**

**7.1 Test Case**

The Force.com platform requires that at least 75% of the Apex Code in an org be executed via unit tests in order to deploy the code to production. You shouldn’t consider 75% code coverage to be an end ­goal though. Instead, you should strive to increase the state coverage of your unit tests. Code has many more possible states than it has lines of code. For example, the following method has 4,294,967,296 different states: System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

* **BLACK BOX TESTING** :-

The technique of testing without having any knowledge of the interior workings of the application is called black­ box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black ­box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

* **WHITE BOX TESTING:-**

White­box testing is the detailed investigation of internal logic and structure of the code. White­box testing is also called glass testing or open­box testing. In order to perform white­box testing on an application, a tester needs to know the internal workings of the code.

* **GREY BOX TESTING**:-

Grey­box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

* **UNIT TESTING:-**

Unit Testing contains the testing of each unit of Recruitment Application. We have tested each interface by input values and check whether it is working properly working or not we also tested database connectivity. We have entered value in interface and check that the values are properly goes to corresponding tuples or not.

* **INTEGRATION TESTING:-**

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom­up integration testing and Top­down integration testing.

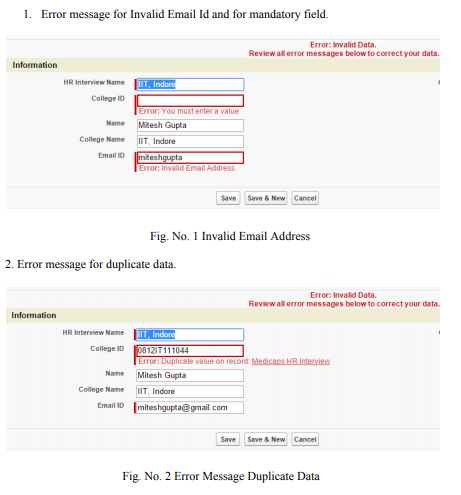
* **SYSTEM TESTING**:-

System testing tests the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards. This type of testing is performed by a specialized testing team.

* **ACCEPTANCE TESTING:-**

This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client’s requirement. The QA team will have a set of pre­written scenarios and test cases that will be used to test the application. In System Testing we have tested entire Recruitment Application. We have run all programs as a single system and inputs various test cases and analyse that all are going correctly or not. In system testing we have tested various test cases. According to which, Application showed the corresponding error message

**7.2 Cases**

****

**CHAPTER 8**

**Conclusion and Future Scope**

**8.1 Conclusion**

The essence of recruitment can be summed up as "the philosophy of attracting as many applicants as possible for given jobs". The face value of this definition is what guided recruitment activities in the past. These days, however, the emphasis is on aligning the organization's objectives with that of the individuals. By making this a priority, an organization safeguards its interests and standing. After all, a satisfied workforce is a stable workforce which also ensures that an organization has credible and reliable performance. In a bid to underscore this subtle point, the project examines the various processes and nuances one of the most critical activities of an organization

**8.2 Limitation and Future Scope**

**8.2.1 LIMITATIONS:-**

Recruitment application project have some limitation as this is first application. In this application the list of selected object in each round of interview is still done manually HR select the list and create a email manually. The another limitation is there must required a Internet connection and Browser application. Only attract job seekers and not actual difference makers. The best candidates aren’t looking for a job they’re happy doing what they’re doing.

Miss out on great hires because your recruiter is tunneled in on only interviewing candidates with the right keywords in their resume.

Recruiters waste a lot of time with unqualified candidates.

The barrier to entry is too low and anybody can apply to your job.

**8.2.2 FUTURE SCOPE:-**

As this is the combination of existing and non existing recruitment system it has huge scope. It fulfils the requirement of every organization. This is a common platform both for applicant and HR to check the vacancies and fill those vacancies with best suitable candidate

In future, job portals will become the necessity of every job seeker. The candidates will be dependent and these job portals to provide them jobs.

**CHAPTER 9**

**REFERENCES**

The following references have been used by me, during all the phases of the project:

* Sharma, Arun, Deva Rangarajan, and Bert Paes brugghe. "Increasing resilience by creating an adaptive salesforce." *Industrial Marketing Management* 88 (2020): 238-246.
* . Kim, Stephen K., and Yeon Sung Jung. "Regaining control of salesforce." *Industrial Marketing Management* 73 (2018): 84-98.
* Bailey, Ryan T., Ali Tasdighi, Seonggyu Park, Saman Tavakoli-Kivi, Tadesse Abitew, Jaehak Jeong, Colleen HM Green, and Abeyou W. Worqlul. "APEX-MODFLOW: A New Integrated Model to Simulate Hydrological Processes in Watershed Systems." *Environmental Modelling& Software* (2021): 105093.
* Leivaditis, Alexandros, Alexandros Singh, Giannos Stamoulis, Dimitrios M. Thilikos, and Konstantinos Tsatsanis. "Minor obstructions for apex-pseudoforests." *Discrete Mathematics* 344, no. 10 (2021): 112529.
* Fraenkel, Stefan, Darek M. Haftor, and Natallia Pashkevich. "Salesforce management factors for successful new product launch." *Journal of Business Research* 69, no. 11 (2016): 5053-5058.
* Egenhofer, Max J. "Spatial SQL: A query and presentation language." *IEEE Transactions on knowledge and data engineering* 6, no. 1 (1994): 86-95.
* Kießling, Werner, and Gerhard Köstler. "Preference SQL—design, implementation, experiences." In *VLDB'02: Proceedings of the 28th* International Conferenceon Very Large Databases, pp 990-1001. Morgan Kaufmann, 2002.

|  |  |
| --- | --- |
|  |  |

* Pull ig, Chris, James G. Max ham III, and Joseph F. Hair Jr. "Salesforce automation systems: An exploratory examination of organizational factors associated with effective implementation and salesforce productivity." *Journal of Business Research* 55, no. 5 (2002): 401-415
* Li, Yi, Caiping Tian, Keke Liu, Ying Zhou, Jing Yang, and Peng Zou. "A clickable APEX probe for proximity-dependent proteomic profiling in yeast." *Cell Chemical Biology* 27, no. 7 (2020): 858-865.
* Pitt, Leyland F., Michael T. Ewing, and Pierre R. Berthon. "Proactive behavior and industrial salesforce performance." *Industrial marketing management* 31, no. 8 (2002): 639-644.
* Archibald, Russell D. "Project management education and training in the USA." *International Journal of Project Management* 7, no. 4 (1989): 199-200.
* 12Cicmil, Svetlana, Terry Williams, Janice Thomas, and Damian Hodgson. "Rethinking project management: researching the actuality of projects." *International journal of project management* 24,

|  |  |
| --- | --- |
|  |  |