

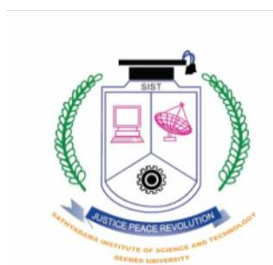
ONLINE CAMPUS RECUIRTMENT SYSTEM

Submitted in partial fulfilment of the requirements for the award of
Bachelor of science Degree in Computer Science

By

KAVIBHARATHI.S (Reg No. 38290035)

RITHIKA.S (Reg No. 38290070)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI-600119



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119, Tamil Nadu, India.



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certificate that this project is the bonafide work of
KAVIBHARATHI.S (RegNo.39290035) and RITHIKA.S(Reg No.38290070)
who carried out the project entities “**ONLINE CAMPUS RECRUITMENT
SYSTEM**” under my supervision from November 2020 to March 2021.

Internal Guide

Dr. J. ALBERT MAYAN M.E., Ph.D.,

Head of the Department

Submitted for viva voice Examination held on

Internal Examiner

External Examiner

DECLARATION

I, KAVI BHARATHI.S (Reg No.38290070) and RITHIKA.S(Reg No.38290070) hereby declare that the project report entitled **ONLINE CAMPUS RECRUITMENT SYSTEM** done by me under the guidance of **Ms. ANANDHI** is submitted on partial fulfillment of the requirement for the Award of Bachelor of Science Degree in Computer Science.

DATE

PLACE: CHENNAI

SIGNATURE OF STUDENT

III

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind of encouragement in doing this Project and for completing it successfully. I am grateful to them.

I convey my thanks to Dr. T. SAAIKALA, M.E., Ph.D., Dean, school of computing and Dr. S. VIGNESHWARI, M.E., Ph.D., and Dr. L. LAKSHMANAN, M.E., Ph.D., Head of the Department, Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide Ms. ANANDHI for her valuable guidance, suggestion and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teachers and Non-teaching staff member of the Department of computer science and engineering who were helpful in many ways for the completion for the project.

TABLE OF CONTENT

CHAPTERNO	TITLE	PAGE NO
	ABSTRACT	1
	LIST OF FIGURES	
1	INTRODUCTION	2
	1.1 OVERVIEW	2
2	LITERATURE SURVEY	5
	2.1 GENERAL	5
3	AIM AND SCOPE OF PROJECT	8
	3.1 AIM OF THE PROJECT	8
	3.2 SCOPE	8
	3.3 SYSTEM REQUIRMENT	8
	3.3.1 SOFTWARE REQUIRMENT	9
	3.3.2 HARDWARE REQUIRMENT	9
	3.4 MICROSOFT .NET FRAMEWORK	9
	3.4.1 FEATURE OF.NET	9
	3.4.2 THE .NET FRAME WORK	10

	3.4.3 MANAGED CODE	11
	3.4.4 MANAGED DATA	11
	3.4.5 COMMAN TYPE SYSTEM	11
	3.4.6 COMMAN LANGUAGE PECIFICATION	12
	3.4.7 THE CLASS LIBRARY	12
	3.4.8 LANGUAGE SUPPORTED BY .NET	13
	3.4.9 CONSTRUCTOR & DISTRUCTOR	15
	3.4.10 GARBAGE COLLECTION	15
	3.4.11 OVERLOADING	16
	3.4.12 MULTITHREADING	16
	3.4.13 STRUCTURED EXCEPTIO HANDLING	16
	3.5 THE.NET FRAME	17
	3.5.1 OBJECTIVES OF.NET	17
4	DESIGN METHODOLOGY	18
	4.1 EXISTING SYSTEM	18
	4.1.1 DIADVANTAGES	18
	4.2 PROPOSED SYSTEM	19
	4.2.1 ADVANTAGES	20

	4.3 MODULES	20
20	4.3.1 MODULES IN THE PROJECT	
21	4.3 ARCHITECTURE	
21	4.4 USERCASE	
	4.5 SEQUENCE DIAGRAM	22
22	4.6 STATE CHART DIAGRAM	
23	4.7 FEASIBILTY STUDY	
	4.7.1 TEACHNICAL FEASIBILITY	24
	4.7.2 ECONOMIC FEASIBILITY	24
	4.8 DFD DIAGRAM	25
	4.8.1 ZERO LEVEL	25
	4.8.2 FIRST LEVL	25
5	TYPES OF TESTING	28
	5.1 UNIT TESTING	28

	5.2 INTERGRATION TESTING	28
	5.3 FUNCTIONAL TESTING	29
	5.4 SYSTEM TESTING	29
	5.5 WHITE BOX TESTING	30
	5.6 BLACK BOX TESTING	30
	5.7 UNIT TESTING	30
	5.8 TEST STRATEGY & APPROACH	30
6	RESULT	31
7	CONCLUSION & FUTURE ENHANCEMENT	32
	7.1 CONCLUSION	32
	7.2 FUTURE ENHANCEMENT	32
	REFERENCE	33
	APPENDIX	34
	A. SOURCE CODE	
	B. SCREENSHOTS	35

LIST OF FIGURES

FIGURE NO:	FIGURE NAME	PAGE NO
4.3	OV ERALL ARCHITECTURE	21
4.4	USERCASE	21
4.5	SEQUENCE DIAGRAM	22
4.6	STATE CHAR DIAGRAM	22
4.8.1	ZERO LEVEL	25
4.8.2	FIRST LEVL	26

STUDENT RECRUITMENT MANAGEMENT SYSTEM

ABSTRACT

This paper presents a Job Recruitment System that consists of a student login, company login and admin login. This is like a monitoring system where admin can monitor the profile of the students. This system allows the student to create their profiles and upload all their details to their job recruitment profiles where the admin only can check each student details and also can monitor their students job details. The software system allows students to view a list of companies who have posted frequently. The admin has overall rights over the system and can moderate and delete any details not pertaining to rules. The system handles student as well as company data and efficiently displays all this data to respective sides.

Objectives:

- To prepare students ready for campus.
- To provide company details and online aptitude for students.
- To provide industry institute interaction.

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The online job portal system is a platform between job seeker(student) and job provider(company). The Student or Job seeker can easily find and apply for job by login into system. The Employer or company can easily get expert employee from our job portal site by posting a job. The job portal website is a common place between job seeker and company. When job seeker can login in his account then he can get notification of company lists which match with his profile or education detail. so job seeker can easily apply to this job and he can also send message to company for other inquiry. When company login in account, company or employer can get alert of job seeker lists which match with all criteria for posted job.

This project Online Recruitment System is an online website in which job seekers can register themselves online and apply for job and attend the exam. Online Recruitment System provides online help to the users all over the world. Using web recruitment systems like recruitment websites or jobsites also play a role in simplifying the recruitment process.

Such websites have facilities where prospective candidates can upload their CV's and apply for jobs suited to them. Such sites also make it possible for recruiters and companies to post their staffing requirements and view profiles of interested candidates. Earlier recruitment was done manually and it was all at a time consuming work. Now it is all possible in a fraction of second. It is all done online without much time consuming. Today's recruitment applications are designed to do a whole lot more than just reduce paperwork. They can make a significant contribution to a company's marketing and sales activity. Recruitment websites and software make possible for managers to access information that is crucial to managing their staff, which they can use for promotion decisions, payroll considerations and succession planning.

Online Recruitment System enables the users to have the typical examination facilities and features at their disposal. It resolves typical issues of manual examination processes and activities into a controlled and closely monitored work flow in the architecture of the application. This multiplatform solution brings in by default, the basic intelligence and immense possibilities for further extension of the application as required by the user.

The system makes it friendly to distribute, share and manage the examination entities with higher efficiency and easiness. The objective of these websites is to serve as a common meeting ground for jobseekers and employers, both locally and globally, where the candidates find their dream jobs and recruiters find the right candidate to fulfill their needs. These sites are specifically designed for those who seek the most demanding and challenging positions in their chosen field, with the most dynamic employers. Thousands of websites compete for your attention- each has its own unique interface, URL and peculiarities.

A quick look at the overall trends in Online recruiting shows the rise in the importance of marketing the web site, online training, dawn of video interviews and emergence of professional Internet Recruiters. Online recruiting and online recruiting systems, with its emphasis on a more strategic decision making process is fast gaining ground as a popular outsourced function.

CHAPTER 2

LITERATURE SURVEY

2.1 GENERAL

Literature survey is a text of scholarly paper, which includes current knowledge including substantive findings as well as theoretical and methodological contributions about particular topics.

Our research method is mainly based on the standard papers and journals under the campus recruitments methods. Over the past years many campus system methods have been developed and is used. In our survey we have mainly focused about the different campus recruitment methods. Some of them are discussed in this following literature survey section.

1.SMS-based Mobile Recommendation System for Campus Recruitment in China

Author: Xiangpei Hu , Lirong Wu , Chao Li , Minfang Huang

Year: 2001

This paper presents a SMS-based recommendation system for campus recruitment in China, which can help college placement office to match the companies and students with higher precision and lower cost.

We are mainly focusing on profile matching and preference-list-based two-sided matching for further recommendation. With regard to profile matching, three kinds of matching methods (i.e., semantic matching, tree-based knowledge matching and SMS-based query matching) are integrated according to representations of attributes of students and companies, and then the profile similarity degree is acquired. Another focus is to provide two-sided matching from the perspective of central bureau (college placement office). Based on profile similarity degree, the preference lists of companies and students are calculated, which serves as the input of two-sided matching.

2.Research on Campus Recruitment management platform based on dynamic electronic commerce

Author: Lu Shumin, Rao Yuan

Year: 2010

Now, the employment for college graduates is becoming a focus-point problem for the society and livelihood in China. With the rapid development of information technology and electronic commerce applications, more and more people realized that only to promote the symmetries and transparency about information can provide more equal opportunities for various graduates. In order to solve the deficiency about existed employment management system, a core business scenario is proposed in the paper for the whole process of employment, which includes five roles, such as government, enterprise, university, student, and platform administrator.

Based on analyzing the five meta-models for employment in dynamic e-Commerce, i.e., B2C for student, B2B for enterprise and university, C2C for personal society network, mobile Commerce and collaborative e-Commerce for employment application, a new dynamic e-Commerce model, enterprise-University- Student Trading Model (DEC-EUST) about recruitment for college graduates is proposed in the paper. Furthermore, an independent management platform for Campus Recruitment, XiaoZhen website, is designed, which integrates the five meta-model for employment channel and provides comprehensive information services for every platform role.

3.Facilitating Job Recruitment Process Through Job Application Support

Author: Junalux Chalidabhongse, Nattapon Jirapokakul , Rata Chutivisarn

Year: 2006

Job recruitment and selection process is one of the most tedious tasks that human resource department in every organization has to perform. The recruitment and selection process consists of several steps. This paper focuses mainly on the step in which the applicants have to fill out application forms and undergo an initial screening interview. This step is very important because human resource recruiters must choose qualified candidates out of many applicants to be interviewed with the supervisors responsible for the job. Recruiters of many companies face a lot of problems performing this step especially when they have to perform it manually. This we were motivated to develop a more efficient way in filling out and processing application forms and develop job application support system to aid interviewers during the preliminary interview process.

Job application support system consists of three main components including electronic application form, electronic telephone interview and electronic interview.

The application module is constructed based on the competency matrix derived at each level of employment. Interview questions are determined based on the chosen competencies and are used to screen out candidates during a competency-based behavioral interview. A quantitative multi-objective decision model and job assignment model are developed for job-applicant matching process. The models are used to manage the data from the job application forms and evaluate the results from job interviewers. Job application support system is expected to bring about many positive changes. It can speed up the recruitment process by filtering out unwanted applicants and allow a better facilitation of the interview process by providing interviewers with guidelines and forms with which they could fill out. Cost can be saved because there are less paperwork. The quantitative multi-objective decision model and job assignment model that are developed can provide a more efficient way for job-applicant matching process

4. Matching Sem: Online recruitment system based on multiple semantic resources

Author: Aseel B. Kmail , Mohammed Maree , Mohammed Belkhatir

Year: 2015

The growth of online recruitment has spurred the need for more effective automated systems.

On the one hand, traditional approaches based on keyword-based matching techniques suffer from low precision, i.e. a large fraction of the systems suggestions are irrelevant. On the other hand, the newer semantics-based approaches are penalized by limitations of the exploited semantic resources, namely semantic knowledge incompleteness and limited domain coverage. In this paper, we present an automatic semantics-based online recruitment system that reuses knowledge captured in multiple existing semantic resources to match between candidate resumes and job posts. In addition, we use statistical-based concept-relatedness measures to alleviate the problem of semantic knowledge incompleteness in the exploited resources. An experimental instantiation of the proposed system has been installed to validate its effectiveness in matching job applicants to job posts.

5. The Design and Realization of Online Virtual Campus of NEU

Author: Yu Chun-Yang , Wang Hong

Year: 2008

In the 21st century, the technology of Web develops very fast and so does with the education way by Web. It is focus on the design and complement of virtual campus based on Torque Game Engine in this thesis. It uses Torque script as source code to do programming works. Then it uses constructor, 3D Studio Max and Photoshop to deal with modeling work of building and player and drawing of

texture. In the end it realizes the demand of multi-access by one PC as server and others as client.

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

3.1 AIM OF THE PROJECT

This paper presents a Campus Recruitment System that consists of a student login, company login and admin login. This is like a monitoring system where admin can monitoring Student profile. This system allows the student to create their profiles and upload all their details including their marksheets to their campus recruitment profiles where the admin only can check each student details and also can monitor their student's campus_ details.

3.2 SCOPE

To develop a system for admin who can monitor all the Campus Recruitment profiles and also monitor the students detail which has been already linked to their campus profile. Through this admin can monitor the student's location and activity. The system also consists of a company login where various companies visiting the college can view a list of students in that college and also their respective resumes. The software system allows students to view a list of companies who have posted frequently asked question etc. The admin has overall rights over the system and can moderate and delete any details not pertaining to college placement rules. The system handles student as well as company data and efficiently displays all this data to respective sides.

3.3 SYSTEM REQIRMENT

8

3.3.1. Software Requirements:

- Operating system : - Windows XP.
- Front End : - C#
- Back End : - MS Access

3.3.2. Hardware Requirements:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 160 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Color.
- Mouse : Logitech.
- RAM : 2GB.

3.4 MICROSOFT.NET FRAMEWORK

3.4.1 Features of .net

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral

platform for writing programs that can easily and securely interoperate. There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script.

9

The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

3.4.2 The .net framework

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are

- ◆ Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
- ◆ Memory management, notably including garbage collection.
- ◆ Checking and enforcing security restrictions on the running code.

- ◆ Loading and executing programs, with version control and other such features.
- ◆ The following features of the .NET framework are also worth description:

10

3.4.3 Managed Code

The code that targets .NET, and which contains certain extra Information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

3.4.4 Managed Data

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn’t get garbage collected but instead is looked after by unmanaged code.

3.4.5 Common Type System

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another

language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

3.4.6 Common Language Specification

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

3.4.7 THE CLASS LIBRARY

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity. The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

3.4.8 LANGUAGES SUPPORTED BY .NET

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services.

The .NET framework supports new versions of Microsoft's old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family.

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework. C# is Microsoft's new language. It's a C-style language that is essentially "C++ for Rapid Application Development". Unlike other languages, its specification is just

the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

13

Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages.

Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State's Perl Dev Kit.

Other languages for which .NET compilers are available include

- FORTRAN
- COBOL
- Eiffel

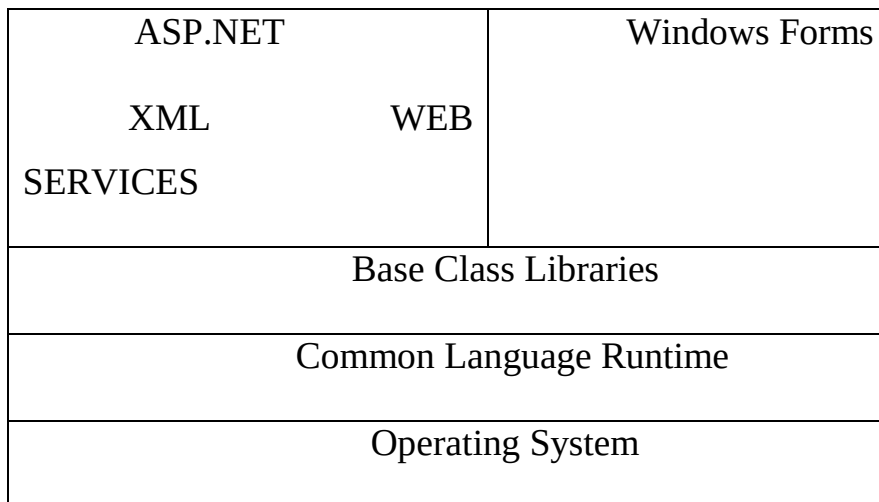


Figure. NET Framework

C#.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR (Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services C#.NET is a CLS-compliant language. Any objects, classes, or components that created in C#.NET can be used in any other CLS-compliant language. In addition, we can use objects, classes, and components created in other CLS-compliant languages in C#.NET .The use of CLS ensures complete interoperability among applications, regardless of the languages used to create the application.

3.4.9 Constructors and Destructors

Constructors are used to initialize objects, whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In C#.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

3.4.10 Garbage Collection

Garbage Collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by

destroying objects that are no longer in use. In C#.NET, the garbage collector checks for the objects that are not currently in use by applications.

15

When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

3.4.11 Overloading

Overloading is another feature in C#. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures, we can use it for constructors and properties in a class.

3.4.12 Multithreading

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, we can use multithreading to decrease the time taken by an application to respond to user interaction.

3.4.13 Structured Exception Handling

C#.NET supports structured handling, which enables us to detect and remove errors at runtime. In C#.NET, we need to use Try...Catch...Finally statements to create exception handlers. Using Try...Catch...Finally statements, we can create robust and effective exception handlers to improve the performance of our application.

3.5 The .Net Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

3.5.1 OBJECTIVES OF .NET FRAMEWORK

1. To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.
2. To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
3. Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications.

CHAPTER 4

EXPERRIMENTAL OR MATERIAL METHODS

4 Design Methodology

4.1Existing System

The job recruits graduates by maintaining manual records which involves many loop holes.

4.1.1 Disadvantages

1. Redundancy:

Manual records tend to contain data which are redundant. This is because Normalization followed is consistent data base is missing feature in manual records.

2. Inconsistency:

Because of redundant data, there is no consistency in the data.

3. Time Consuming:

Maintaining each students and employees data with much number of fields is a tedious process and takes more time to retrieve back.

4. Improper Validation:

There are many cases where validation is to be performed between the existing data and yet to be entered data. Also, a graduate who

appears for the interview and fails to succeed should not be reconsidered until a period of 3 months.

18

5. Tedious:

An organization data base goes through many numbers of updates each day which is hard to maintain in registers.

4.2 Proposed System

The developed system overcomes all the above listed drawbacks.

- All the Graduates, Employer and Administrator information is maintained in normalized database instead of manual records.
- This feature helps in maintaining database which is consistent, not redundant and easily maintainable.
- This system helps in restricting any graduate to register itself multiple times.
- The two main users are
 1. Student
 2. Administrator
- student is the one who is looking for a job in the same organization who can be a fresher or an experienced.

- After entering all the details, the students can view a page containing all the vacancies available.

19

- students can now apply to one or any number of jobs whose interview details are later intimated by the Administrator.

4.2.1 Advantages

Online Recruitment System enables the users to have the typical examination facilities and features at their disposal. It resolves typical issues of manual examination processes and activities into a controlled and closely monitored work flow in the architecture of the application. This multi platform solution brings in by default, the basic intelligence and immense possibilities for further extension of the application as required by the user. The system makes it friendly to distribute, share and manage the examination entities with higher efficiency and easiness. It is a comprehensive resource for finding a job online.

4.3 Modules

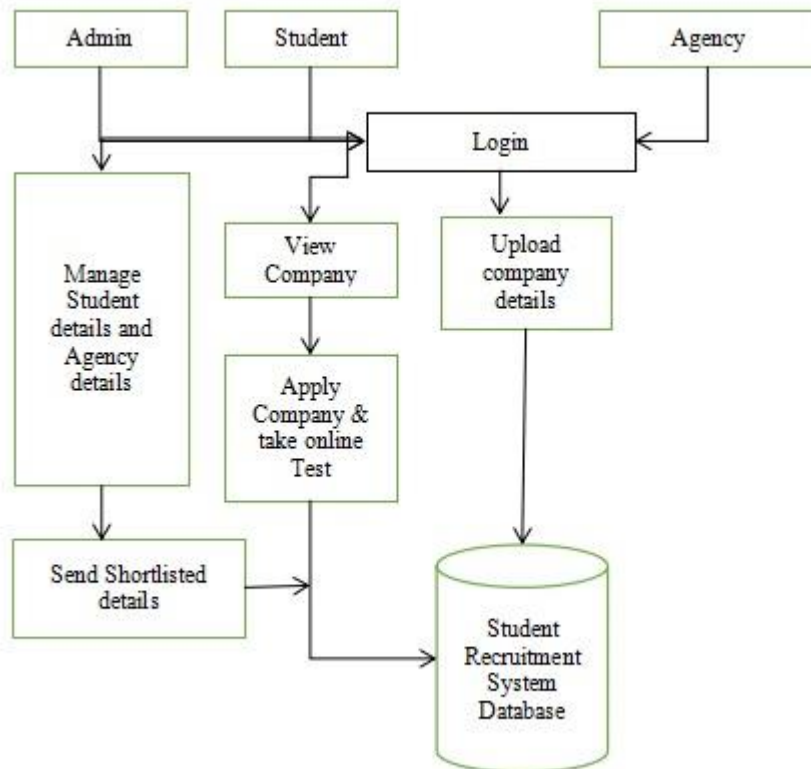
4.3.1 Module in this project:

1. Administrator
2. Jobseekers
3. Company

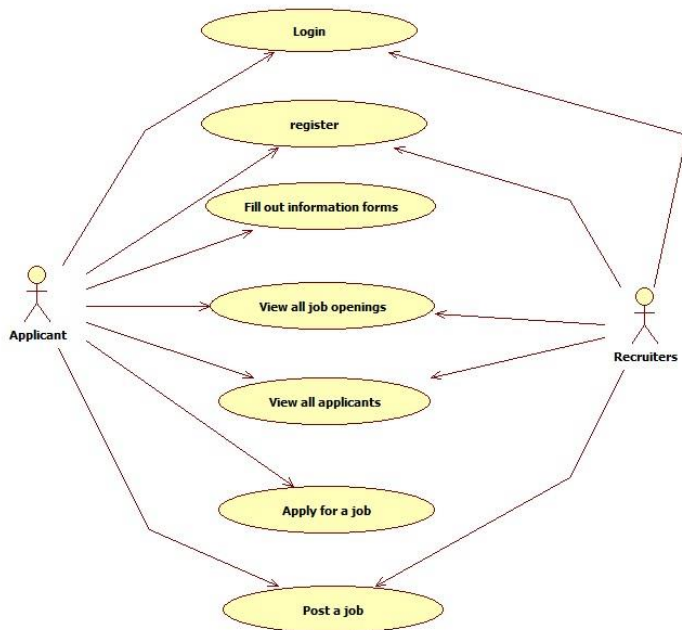
Brief description on the modules:

1. **Administrator:** Administrator has the full authority over the website. He can view all the registered users and have the power to delete them. He can edit the web pages and update them. He can view all the company details also.
2. **Jobseeker:** A jobseeker can register himself. After registration, he will be directed to his homepage. Here he can update his profile, change password and see the examination details and all.
3. **Company:** A company can register itself, conduct online examination, approve or disapprove candidates attending examination and provides results about the selected candidates.

4.3 Architecture

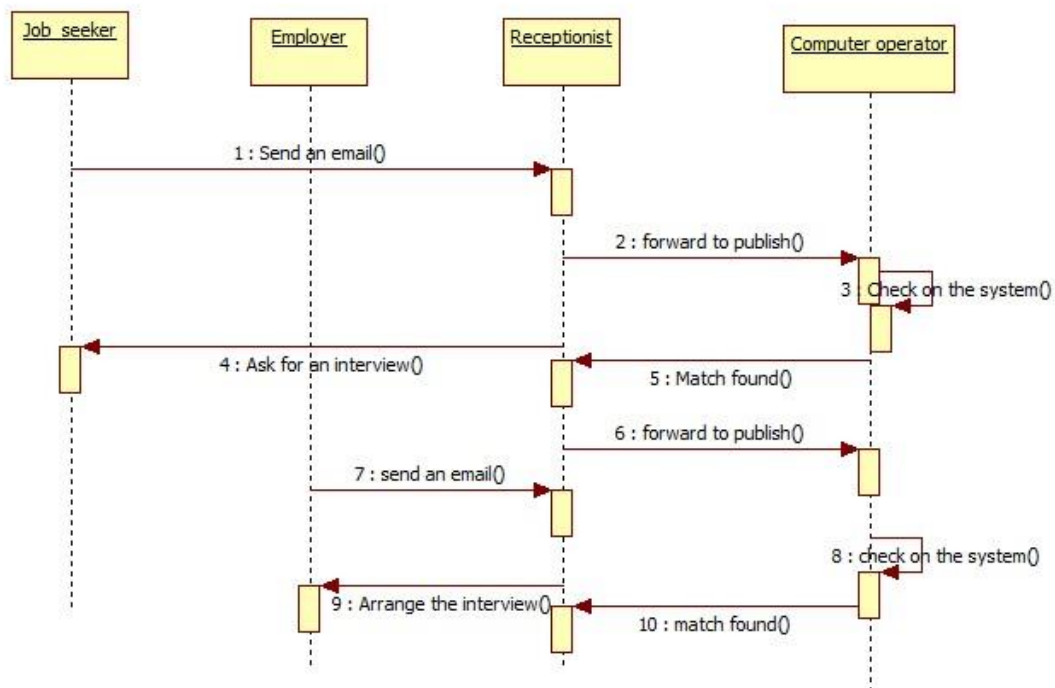


4.4 Usecase



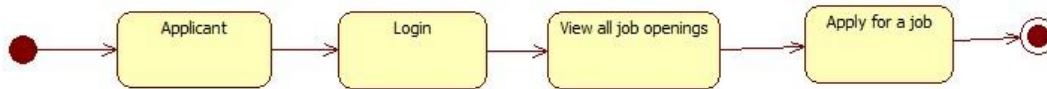
21

4.5 Sequence Diagram



4.6 State chart Diagram

Applicant:



Recruiter:



4.7 FEASIBILITY STUDY

A feasibility analysis usually involves a through assessment of the operational (need), financial and technical aspects of a proposal. Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints. A feasibility study should be relatively cheap and done at the earliest possible time. Depending on the study, the decision is made whether to go head with a more detailed analysis.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis were

Technical Feasibility

Economic Feasibility

Behavioral Feasibility

4.7.1 Technical Feasibility

Technical feasibility includes whether the technology is available in the market for development and its availability. Assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc, in order to give an introduction of technical system. Considering our project it is technical feasible. Online Recruiting and Online Recruitment Systems, with its emphasis on a more strategic decision making process is fast gaining ground as a popular outsourced function.

4.7.2 Economic Feasibility

This feasibility study presents tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service. Thus feasibility study should center along the following points:

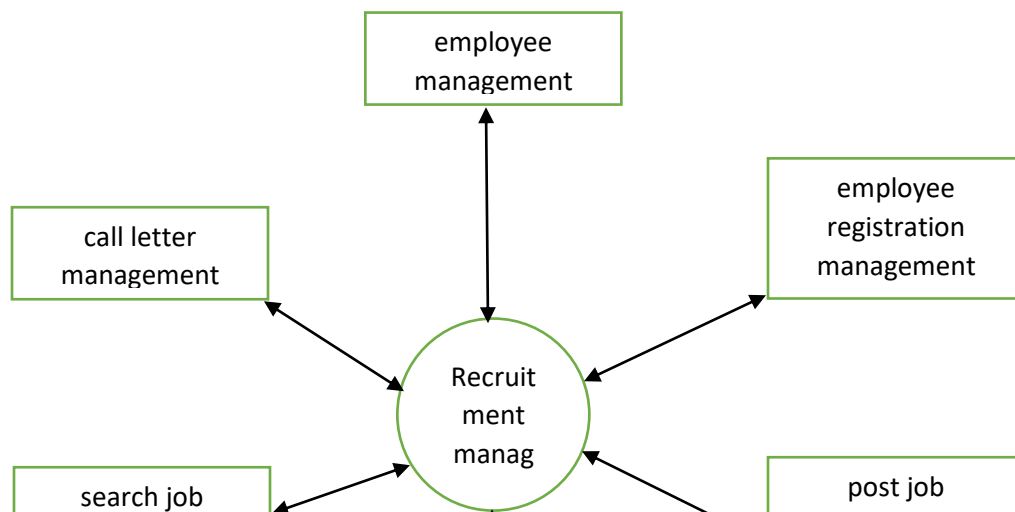
Improvement resulting over the existing method in terms of accuracy, timeliness.

24

4.8 DFD Diagram

4.8.1 Zero Level

This is a zero level patient information system where we have elaborated the high level process of job recruitment system. It is designed to be an at-a-glance and login showing the system as a single high level process, with its relationship to external entities of employee, employee registration and post jobs.

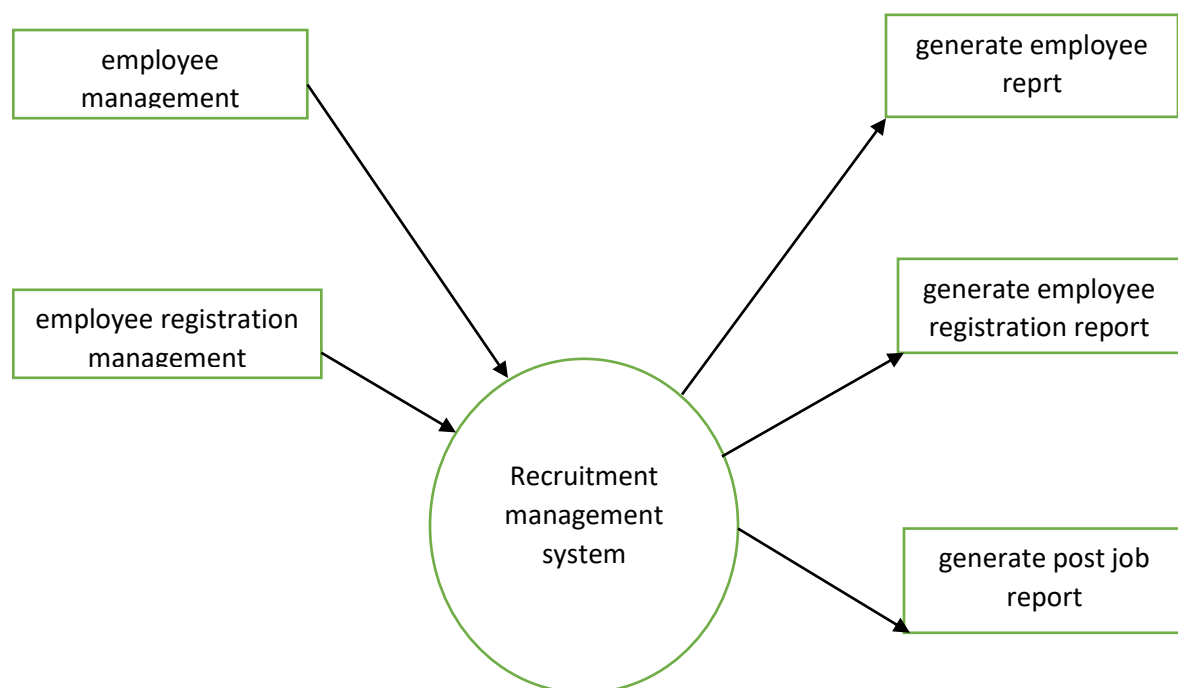


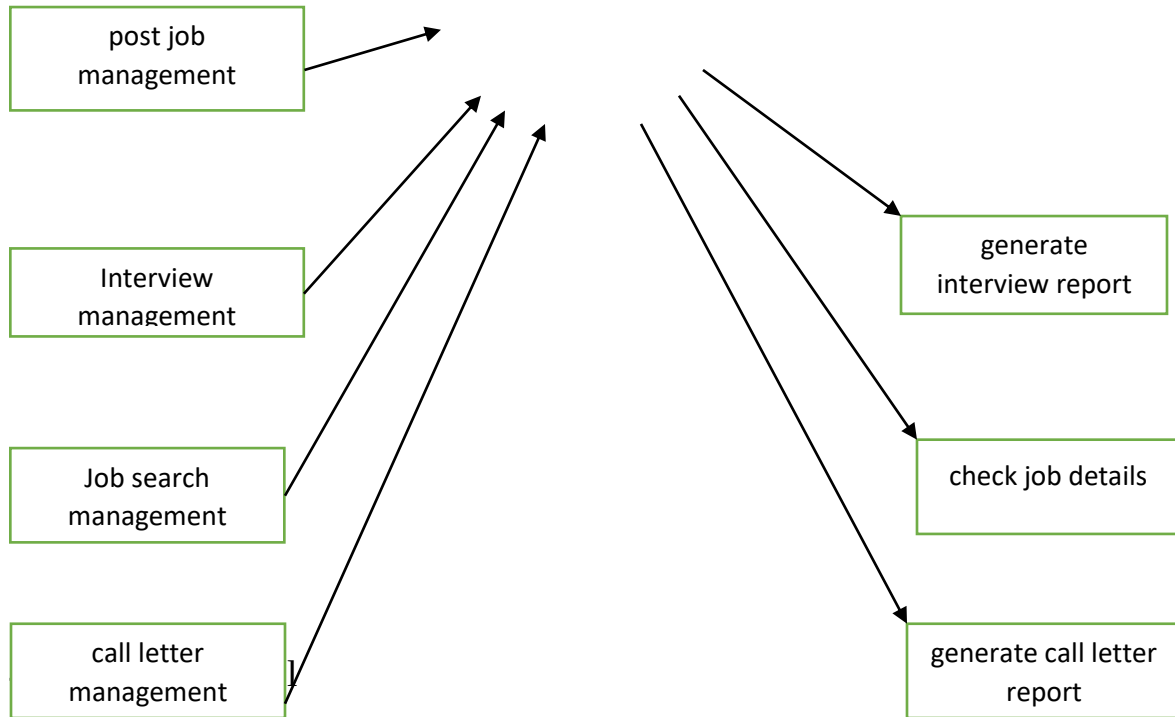
4.8.2 First Level

First level DFD of the patient information system diagram shows how the system is divided into sub system each of deals with one or more of the data flows to or from an external agent, and which together provide all the functionality of the job recruitment system.

25

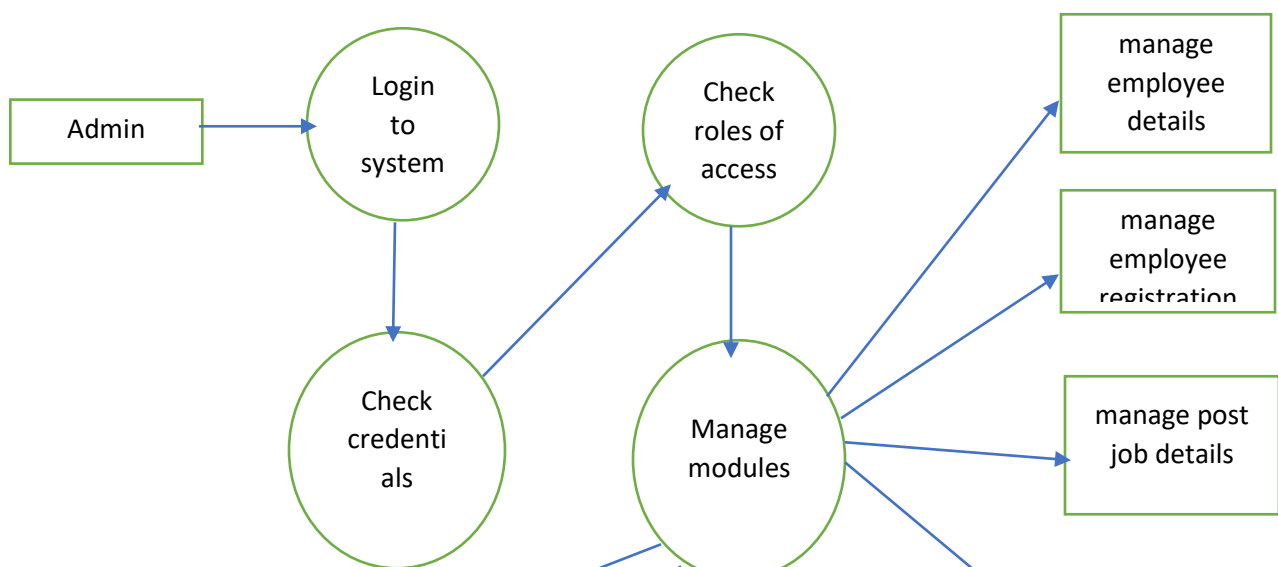
It also identifies internal data stores of login, call letter, search jobs, interview, post jobs that must be the present in order for job recruitment system and it shows the flow of data to various parts of job recruitment system.





26

It may require more functionality of job recruitment system to reach the necessary level of patient information system. It shows how the system is divided into sub modules. It contains more details of login, call letter, search job, interview, post job, employee registration.



CHAPTER 5

TYPES OF TESTING

5.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process

performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is Focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

5.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements

document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

5.7 UNIT TESTING

Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

5.8 TEST STATERGY & APPROACH

Field testing will be performed manually and functional tests will be written in detail.

E-recruitment otherwise known as online recruitment is a time and money saving platform which companies can use in order to reduce cost of recruitment for themselves. Through this research project, the stress of job search can be eradicated and reduced to a minimal level. This project is targeted towards the easy and efficient job search. This system helps in maintaining the information

of potential candidates at one place. It can be easily accessed by both applicant and employers.

It is kept safe for a long period of time without any changes or omission. It reduces the time employer take to make few recruitments in their firm. This system helps the job seekers to get every information required for the process.

Job seek E-recruitment can be improved further by creating modules or functionalities which support the testing of candidate and also referral of such candidates to companies based on their scores and their ranking on the recruitment platform.

CHAPTER 7

CONCLUSION & FUTRE ENHANCEMENT

7.1 CONCLUSION

Online Recruitment Management System can be used by employers to recruit the candidates based on their online test and the further interviews easily. Achieving this objective is difficult using the manual system as the information is scattered, can be redundant and collecting relevant information may be very time-consuming.

All these problems are solved by this project. This system helps in maintaining the information of potential candidates at one place. It can be easily accessed by both applicant and employers. It is kept safe for a long period of time without any changes or omission. It reduces the time employer take to make few recruitments in their firm. This system helps the job seekers to get every information required for the process.

7.2 FUTURE ENHANCEMENT

In proposed online job recruitment system there is scope for improvement of the system. System is not providing the SMS integration. Hence, it can be modified to give the SMS integration. Apart from these there is scope for generating many more features. In the future we can place the system on the cloud so the maintenance of the data can be reduced. The Exam system will integrate with the online placement system so the student result can get directly. There can be many more future Enhancement & improvement in the Online Placement System.

REFERENCES

1. SMS-based Mobile Recommendation System for Campus Recruitment in China, Xiangpei Hu , Lirong Wu , Chao Li , Minfang Huang, 2001

2. Research on Campus Recruitment management platform based on dynamic electronic commerce, Lu Shumin, Rao Yuan, 2010
3. Facilitating Job Recruitment Process Through Job Application Support , Junalux Chalidabhongse, Nattapon Jirapokakul , Rata Chutivisarn, 2006
4. MatchingSem: Online recruitment system based on multiple semantic resources, Aseel B. Kmail , Mohammed Maree , Mohammed Belkhatir, 2015
5. The Design and Realization of Online Virtual Campus of NEU, Yu Chun-Yang , Wang Hong, 2008

APPENDIX

A. SOURCE CODE

Admin.aspx

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Net.Mail;


namespace Job
{

    public partial class admin : System.Web.UI.Page
    {

        //string mail_id = "dotnet.retech@gmail.com,shahana.retech@gmail.com";

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {

            Response.Redirect("viewcomadmin.aspx");

        }

    }

}
```

```
protected void Button2_Click(object sender, EventArgs e)
{
    Response.Redirect("viewmark.aspx");
}
```

```
protected void Button3_Click(object sender, EventArgs e)
{
    Response.Redirect("viewapplyjob.aspx");
}
```

```
protected void Button4_Click(object sender, EventArgs e)
{
    Response.Redirect("mail.aspx");
}
}
```

Agent.aspx

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.OleDb;

using System.Data.SqlClient;

namespace Job

{

    public partial class agent : System.Web.UI.Page

    {

        OleDbConnection con = new

OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data

Source=C:\Users\RECS_02\Documents\onlinejob.mdb");

        protected void Page_Load(object sender, EventArgs e)

        {

        }

    }

    protected void Button2_Click(object sender, EventArgs e)

    {
```

```

        Response.Redirect("index.aspx");
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        con.Open();

        OleDbCommand cmd = con.CreateCommand();

        cmd.CommandType = CommandType.Text;

        cmd.CommandText = "insert into company
values('"+TextBox1.Text+"','"+TextBox2.Text+"','"+TextBox3.Text+"','"+TextBo
x4.Text+"','"+TextBox5.Text+"')";

        cmd.ExecuteNonQuery();

        con.Close();

        Response.Write("<script>alert('Details Uploaded')</script>");

        TextBox1.Text = " ";

        TextBox2.Text = " ";

        TextBox3.Text = " ";

        TextBox4.Text = " ";

        TextBox5.Text = " ";

    }
}

```

```
}
```

Applyjob.aspx

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Web;
```

```
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
```

```
using System.Data;
```

```
using System.Data.OleDb;
```

```
using System.IO;
```

```
namespace Job
```

```
{
```

```
    public partial class applyjob : System.Web.UI.Page
```

```
    {
```

```
        OleDbConnection con = new
```

```
OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=C:\Users\RECS_02\Documents\onlinejob.mdb");
```

```
        byte[] bytes;
```

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Page.IsPostBack == false)
    {
        string sql = "select distinct Id from company";

        OleDbDataAdapter sda = new OleDbDataAdapter(sql, con);

        DataSet ds = new DataSet();

        sda.Fill(ds, "s");

        DataTable dt = ds.Tables["s"];

        foreach (DataRow r in dt.Rows)
        {
            DropDownList1.Items.Add(r["Id"].ToString());
        }
    }
}
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    string filename = Path.GetFileName(FileUpload1.PostedFile.FileName);

    string filecont = FileUpload1.PostedFile.ContentType;
```

```

using(Stream fs = FileUpload1.PostedFile.InputStream)
{
    using(BinaryReader dr = new BinaryReader(fs))
    {
        bytes = dr.ReadBytes((Int32)fs.Length);
    }
}

con.Open();

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into applyjob values('" +
DropDownList1.SelectedItem.ToString() + "','" + TextBox2.Text + "','" +
TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','"
+TextBox6.Text+ "','" +filename+ "');"

cmd.ExecuteNonQuery();

con.Close();

Response.Write("<script>alert('Thanks for Applying!!!')</script>");

}

protected void Button2_Click(object sender, EventArgs e)

```

```
{  
    Response.Redirect("student.aspx");  
}  
}  
}
```

Index.aspx

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Web;  
  
using System.Web.UI;  
  
using System.Web.UI.WebControls;  
  
namespace Job  
{  
    public partial class index : System.Web.UI.Page  
    {  
        protected void Page_Load(object sender, EventArgs e)  
        {
```

```
}
```

```
protected void Button3_Click(object sender, EventArgs e)
```

```
{
```

```
    Response.Redirect("signup.aspx");
```

```
}
```

```
protected void Button2_Click(object sender, EventArgs e)
```

```
{
```

```
    Response.Redirect("Login.aspx");
```

```
}
```

```
}
```

```
}
```

Login.aspx

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.OleDb;

namespace Job
{

    public partial class Login : System.Web.UI.Page
    {

        OleDbConnection con = new
OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\RECS_02\Documents\onlinejob.mdb");

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {

            Response.Redirect("index.aspx");

        }

    }

}
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    OleDbDataAdapter sda = new OleDbDataAdapter("select
Profile,Username,Password from signup where Profile ='" +
DropDownList1.SelectedItem.ToString() + "'and Username ='" + TextBox1.Text +
"' and Password = '" + TextBox2.Text + "'", con);

    DataTable dt = new DataTable();

    sda.Fill(dt);

    if (dt.Rows.Count > 0)
    {
        if (DropDownList1.SelectedIndex == 3)
        {
            Response.Redirect("student.aspx");
        }

        else if (DropDownList1.SelectedIndex == 2)
        {
            Response.Redirect("agent.aspx");
        }
    }
}
```

```
        else if ((DropDownList1.SelectedIndex == 1) && (TextBox1.Text ==  
"Admin") && (TextBox2.Text == "Admin"))  
  
        {  
  
            Response.Redirect("admin.aspx");  
  
        }  
  
        else  
  
        {  
  
            Response.Write("<script>alert('Invalid Username or  
Password')</script>");  
  
        }  
  
    }  
  
}
```

Mail.aspx

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Web;  
  
using System.Web.UI;  
  
using System.Web.UI.WebControls;
```

```
using System.Net.Mail;
```

```
using System.Data;
```

```
using System.Data.OleDb;
```

```
namespace Job
```

```
{
```

```
    public partial class mail : System.Web.UI.Page
```

```
    {
```

```
        OleDbConnection con = new
```

```
OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=C:\Users\RECS_02\Documents\onlinejob.mdb");
```

```
        string mail_id;
```

```
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
```

```
            if (Page.IsPostBack == false)
```

```
            {
```

```
                getmailaddress();
```

```
            }
```

```
}
```

```
protected void Button1_Click(object sender, EventArgs e)
```

```
{
```

```
    getmailaddress();
```

```
    mail_id=DropDownList1.SelectedItem.ToString();
```

```
    try
```

```
    {
```

```
        MailMessage msg = new MailMessage();
```

```
        msg.From = new MailAddress("cloudDotnet1234@gmail.com");
```

```
        msg.To.Add(mail_id);
```

```
        msg.Body = "Congratulation You have been Shortlisted for next Round";
```

```
        msg.IsBodyHtml = true;
```

```
        SmtpClient smtp = new SmtpClient();
```

```
        smtp.Host = "smtp.gmail.com";
```

```
        System.Net.NetworkCredential networkcred = new  
System.Net.NetworkCredential();
```

```
        networkcred.UserName = "cloudDotnet1234@gmail.com";
```

```
        networkcred.Password = "cloud@qwerty";
```

```
        smtp.UseDefaultCredentials = true;
```

```

smtp.Credentials = networkcred;

smtp.Port = 587;

smtp.EnableSsl = true;

smtp.Send(msg);

//Response.Write("<script>alert('Mail Sent!')</script>");

}

catch (Exception ex)

{

    Response.Write("Exception in sendEmail:" + ex.Message);

}

}

public void getmailaddress()

{

    string sql = "select distinct Mail from mark";

    OleDbDataAdapter sda = new OleDbDataAdapter(sql, con);

    DataSet ds = new DataSet();

    sda.Fill(ds, "s");

    DataTable dt = ds.Tables["s"];

    foreach (DataRow r in dt.Rows)

```

```
{  
    DropDownList1.Items.Add(r["Mail"].ToString());  
    mail_id = r["Mail"].ToString();  
  
}  
  
}  
  
}
```

Signup.aspx

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Web;  
  
using System.Web.UI;  
  
using System.Web.UI.WebControls;  
  
using System.Data;  
  
using System.Data.OleDb;  
  
using System.Text.RegularExpressions;  
  
using System.Configuration;
```

```
namespace Job
```

```
{
```

```
    public partial class signup : System.Web.UI.Page
```

```
    {
```

```
        OleDbConnection con = new
```

```
OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=C:\Users\RECS_02\Documents\onlinejob.mdb");
```

```
        protected void Page_Load(object sender, EventArgs e)
```

```
        {
```

```
        }
```

```
        protected void Button2_Click(object sender, EventArgs e)
```

```
        {
```

```
            Response.Redirect("index.aspx");
```

```
        }
```

```
        protected void Button1_Click(object sender, EventArgs e)
```

```
        {
```

```
            con.Open();
```

```

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into signup values('" + TextBox1.Text +
"', '"+TextBox2.Text+"', '"+TextBox3.Text+"', '"+TextBox4.Text+"', '"+TextBox5.Te
xt+"', '"+DropDownList1.SelectedItem.ToString()+"'");

cmd.ExecuteNonQuery();

con.Close();

Response.Write("<script>alert('Submitted')</script>");

TextBox1.Text = " ";

TextBox2.Text = " ";

TextBox3.Text = " ";

TextBox4.Text = " ";

TextBox5.Text = " ";

DropDownList1.Items.Clear();

}

}

}

```

Student.aspx

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;


namespace Job
{

    public partial class student : System.Web.UI.Page
    {

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button3_Click(object sender, EventArgs e)
        {

            Response.Redirect("index.aspx");

        }

        protected void Button1_Click(object sender, EventArgs e)
```

```
{  
    Response.Redirect("viewcompany.aspx");  
}  
  
protected void Button2_Click(object sender, EventArgs e)  
{  
    Response.Redirect("applyjob.aspx");  
}  
  
protected void Button4_Click(object sender, EventArgs e)  
{  
    Response.Redirect("testlogin.aspx");  
}  
}  
}
```

Test.aspx

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.OleDb;


namespace Job
{

    public partial class test : System.Web.UI.Page
    {

        OleDbConnection con = new
OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\RECS_02\Documents\onlinejob.mdb");

        string str;

        OleDbCommand cmd;

        int count = 0;

        string A;

        string B;

        string ab;

        protected void Page_Load(object sender, EventArgs e)
        {
```

```
if(Page.IsPostBack==false)

{

    catget();

}

}

protected void Button1_Click1(object sender, EventArgs e)

{

    if (RadioButton2.Checked == true)

    {

        count = count + 1;

    }

    if (RadioButton5.Checked == true)

    {

        count = count + 1;

    }

    if (RadioButton11.Checked == true)

    {

        count = count + 1;

    }

}
```

```
if (RadioButton16.Checked == true)

{

    count = count + 1;

}

if (RadioButton18.Checked == true)

{

    count = count + 1;

}

if (!String.IsNullOrEmpty(Session["Username"].ToString()))

{

    //Read values from session

    A = Session["Username"].ToString();

}

if (!String.IsNullOrEmpty(Session["Mail"].ToString()))

{

    //Read values from session
```

```
B = Session["Mail"].ToString();

}

con.Open();

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into mark
values('"+A+"','"+B+"','"+count+"')";

cmd.ExecuteNonQuery();

con.Close();

Response.Write("<script>alert('"+count+"');</script>");

RadioButton1.Checked = false;

RadioButton2.Checked = false;

RadioButton3.Checked = false;

RadioButton4.Checked = false;

RadioButton5.Checked = false;

RadioButton6.Checked = false;

RadioButton7.Checked = false;

RadioButton8.Checked = false;

RadioButton9.Checked = false;
```

```
RadioButton10.Checked = false;

RadioButton11.Checked = false;

RadioButton12.Checked = false;

RadioButton13.Checked = false;

RadioButton14.Checked = false;

RadioButton15.Checked = false;

RadioButton16.Checked = false;

RadioButton17.Checked = false;

RadioButton18.Checked = false;

RadioButton19.Checked = false;

RadioButton20.Checked = false;

}

protected void Button2_Click(object sender, EventArgs e)
{
    Response.Redirect("student.aspx");
}

public void catget()
{
```

```

con.Open();

string sql = "select distinct category from test2";

OleDbCommand cmd = new OleDbCommand(sql, con);

OleDbDataReader dr = cmd.ExecuteReader();

if(dr.HasRows)
{
    while(dr.Read())
    {
        DropDownList1.Items.Add(dr[0].ToString());
    }

    dr.NextResult();
}

con.Close();
}

```

```

protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
{
    con.Open();

    //str = "select Question,opt1,opt2,opt3,crtans from test2 where
category='Logical'";

```

```
str = "select Question,opt1,opt2,opt3,crtans from test2 where category='" +  
DropDownList1.SelectedItem.ToString() + "'";
```

```
cmd = new OleDbCommand(str, con);
```

```
OleDbDataReader reader = cmd.ExecuteReader();
```

```
reader.Read();
```

```
Label2.Text = reader["Question"].ToString();
```

```
RadioButton1.Text = reader["opt1"].ToString();
```

```
RadioButton2.Text = reader["crtans"].ToString();
```

```
RadioButton3.Text = reader["opt3"].ToString();
```

```
RadioButton4.Text = reader["opt2"].ToString();
```

```
reader.Read();
```

```
Label4.Text = reader["Question"].ToString();
```

```
RadioButton5.Text = reader["crtans"].ToString();
```

```
RadioButton6.Text = reader["opt2"].ToString();
```

```
RadioButton7.Text = reader["opt3"].ToString();
```

```
RadioButton8.Text = reader["opt1"].ToString();
```

```
reader.Read();
```

```
Label7.Text = reader["Question"].ToString();
```

```
RadioButton9.Text = reader["opt1"].ToString();  
RadioButton10.Text = reader["opt2"].ToString();  
RadioButton11.Text = reader["crtans"].ToString();  
RadioButton12.Text = reader["opt3"].ToString();  
reader.Read();  
Label10.Text = reader["Question"].ToString();
```

```
RadioButton13.Text = reader["opt1"].ToString();  
RadioButton14.Text = reader["opt2"].ToString();  
RadioButton15.Text = reader["opt3"].ToString();  
RadioButton16.Text = reader["crtans"].ToString();  
reader.Read();
```

```
Label8.Text = reader["Question"].ToString();  
RadioButton17.Text = reader["opt1"].ToString();  
RadioButton18.Text = reader["crtans"].ToString();  
RadioButton19.Text = reader["opt3"].ToString();  
RadioButton20.Text = reader["opt2"].ToString();  
reader.Close();  
con.Close();
```

```
    }  
  
    }  
}
```

Testlogin.aspx

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Web;  
  
using System.Web.UI;  
  
using System.Web.UI.WebControls;  
  
using System.Data;  
  
using System.Data.OleDb;  
  
namespace Job  
{  
  
    public partial class testlogin : System.Web.UI.Page  
    {
```

```

        OleDbConnection con = new
OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Users\RECS_02\Documents\onlinejob.mdb");

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {

            OleDbDataAdapter sda = new OleDbDataAdapter("select
Id,Username,Mail,Password from signup where Id='"+TextBox1.Text+"'and
Username='"+ TextBox2.Text + "' and Mail='"+TextBox3.Text+"' and Password
= '" + TextBox4.Text + "'", con);

            DataTable dt = new DataTable();

            sda.Fill(dt);

            if (dt.Rows.Count > 0)
            {

                Session["Username"] = TextBox2.Text;

                Session["Mail"] = TextBox3.Text;

```

```
if(TextBox1.Text!=""&&TextBox2.Text!=""&&TextBox3.Text!=""&&TextBox4.
Text!="")
    {
        Response.Redirect("test.aspx");
    }
}

else
{
    Response.Write("<script>alert('Invalid Username or
Password')</script>");
}

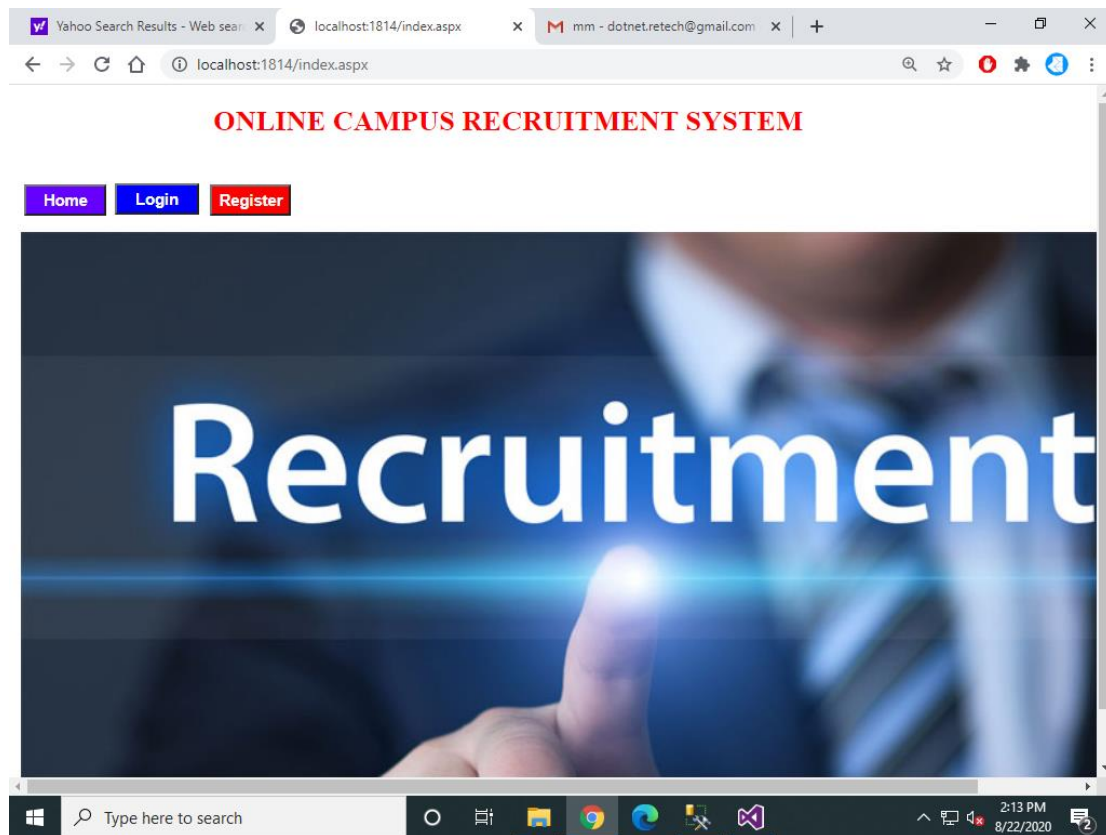
}

protected void Button2_Click(object sender, EventArgs e)
{
    Response.Redirect("student.aspx");
}

}
```

}

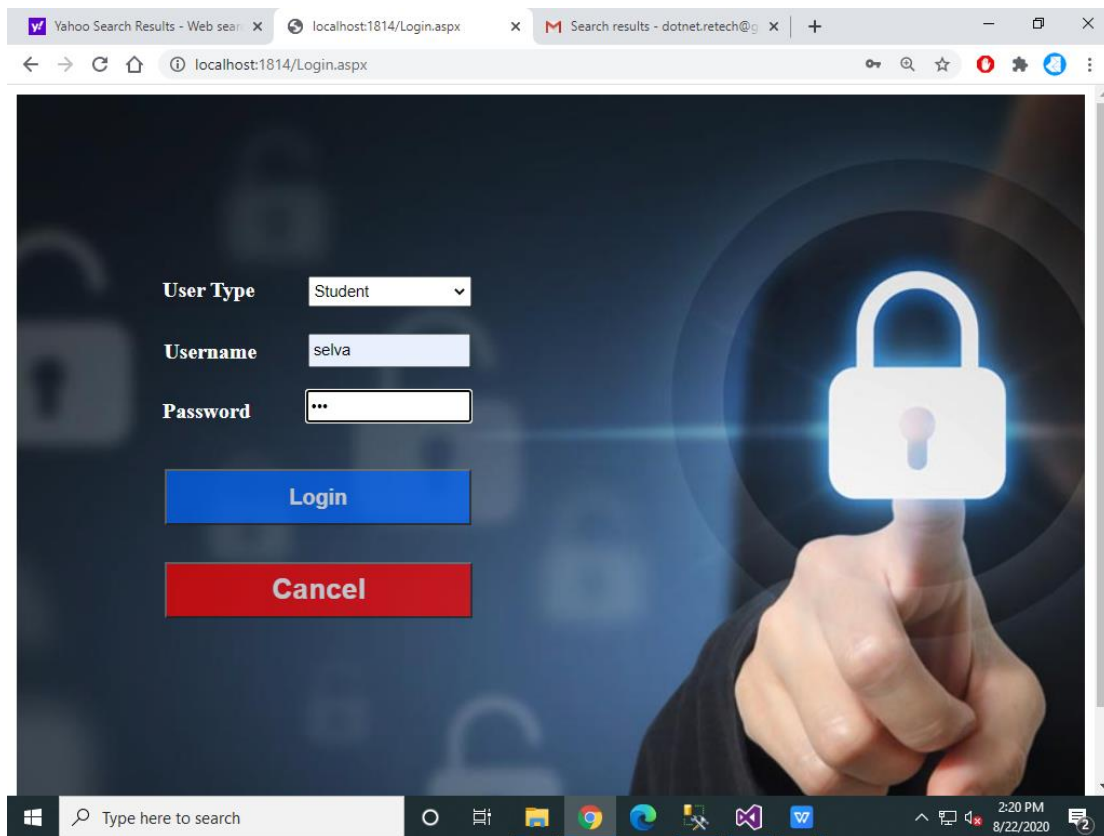
Index.aspx:



Login:



REGISTER:



LOGIN:

Yahoo Search Results - Web search x localhost:1814/viewcompany.aspx x Search results - dotnet.retech@ x +

localhost:1814/viewcompany.aspx

Job Details

[Logout](#)

Id	Companyname	Location	Role	Salary
Go01	Go Tech	Tambaram	IOS Developer	15,000
RE01	Retech Solution	West Tambaram	Developer Java/Python	10,000



Windows taskbar: Type here to search, 2:21 PM, 8/22/2020

VIEW COMPANY DETAILS:

Yahoo Search Results - Web search x localhost:1814/applyjob.aspx x Search results - dotnet.retech@ x +

localhost:1814/applyjob.aspx

APPLY NOW

**COMPUTERIZED APPLICATION FOR PAPER LESS
BASED EXAMINATION SYSTEM RECRUITMENT**

Company ID

Company Name

Username

Phone No.

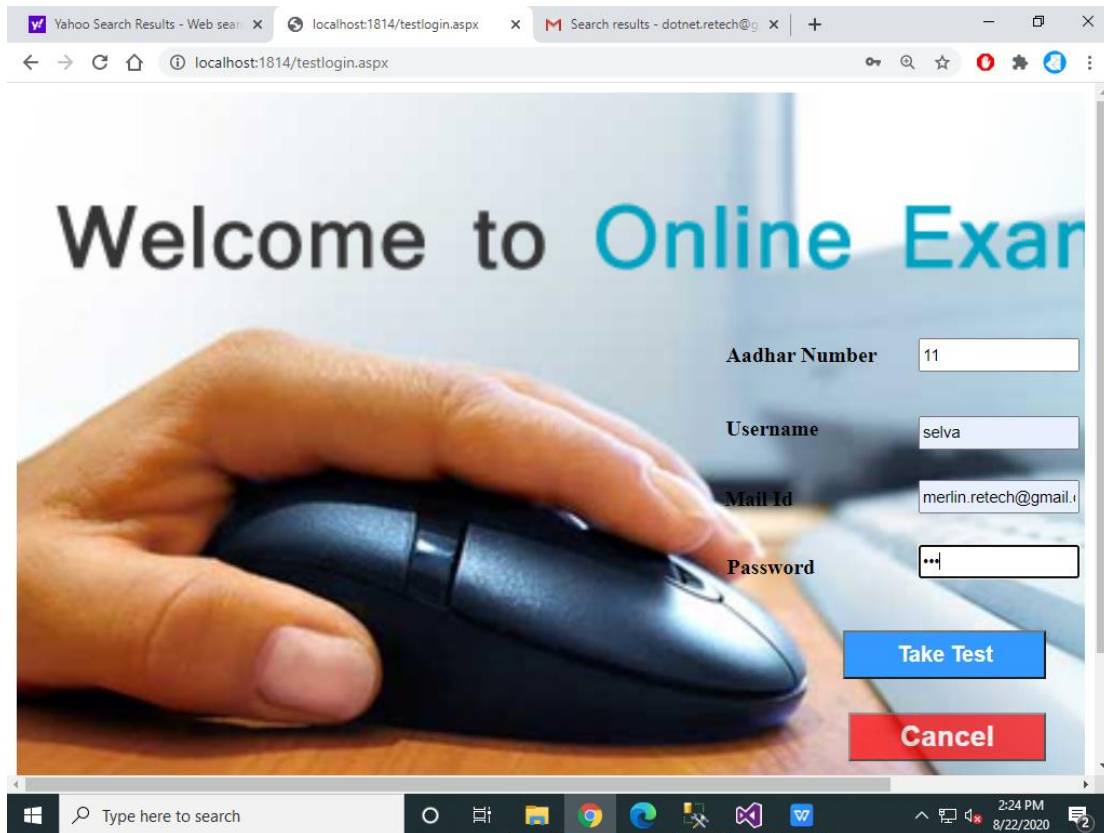
e-Mail

Role

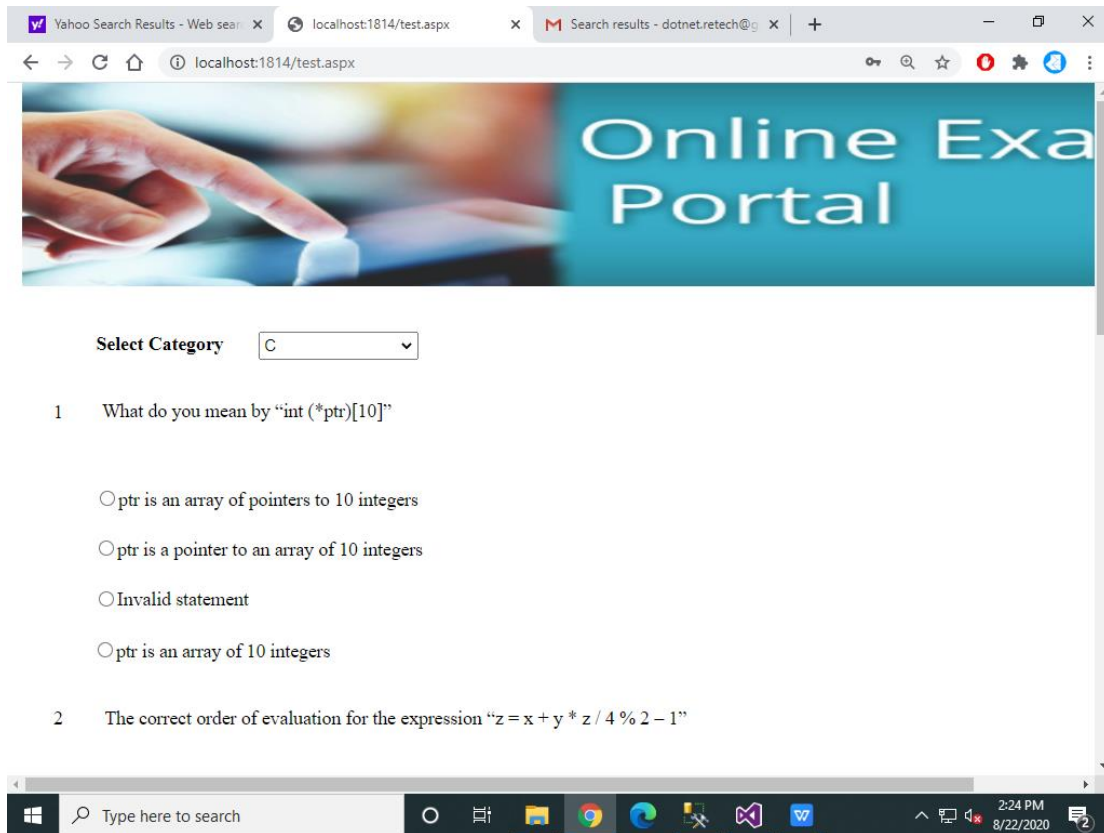
Resume No file chosen

Type here to search 2:23 PM 8/22/2020

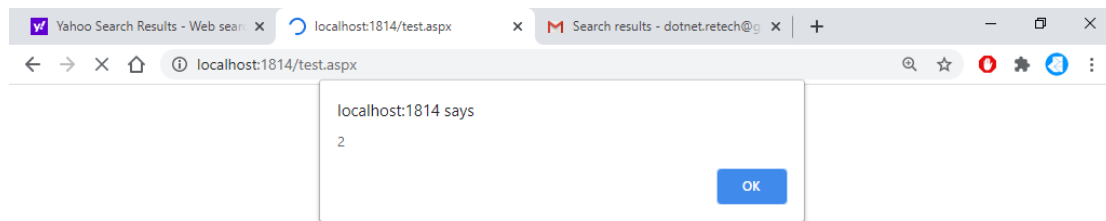
APPLY JOB:



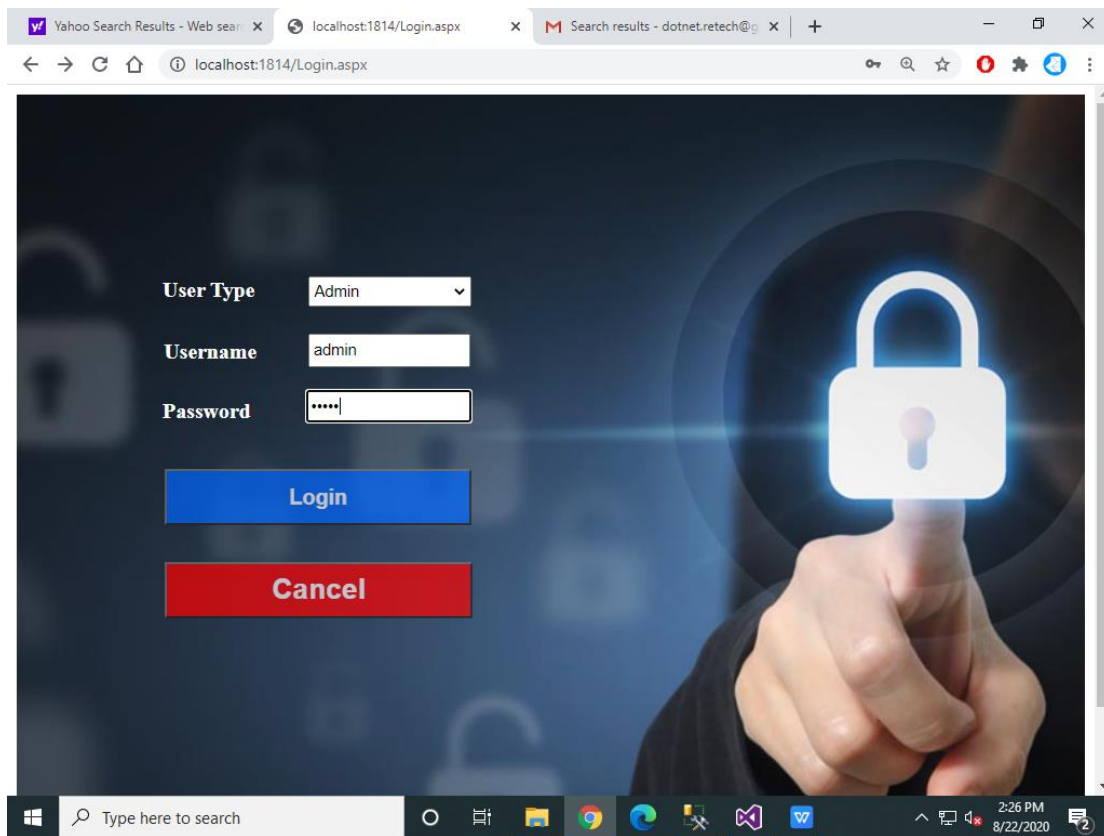
TEST LOGIN:



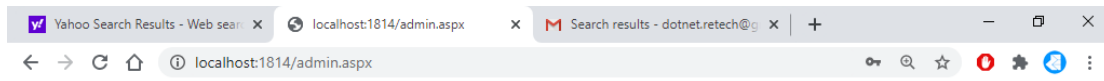
TEST ATTEMPT:



MARKS:



ADMIN LOGIN:



Admin Module

[View Company Details](#)

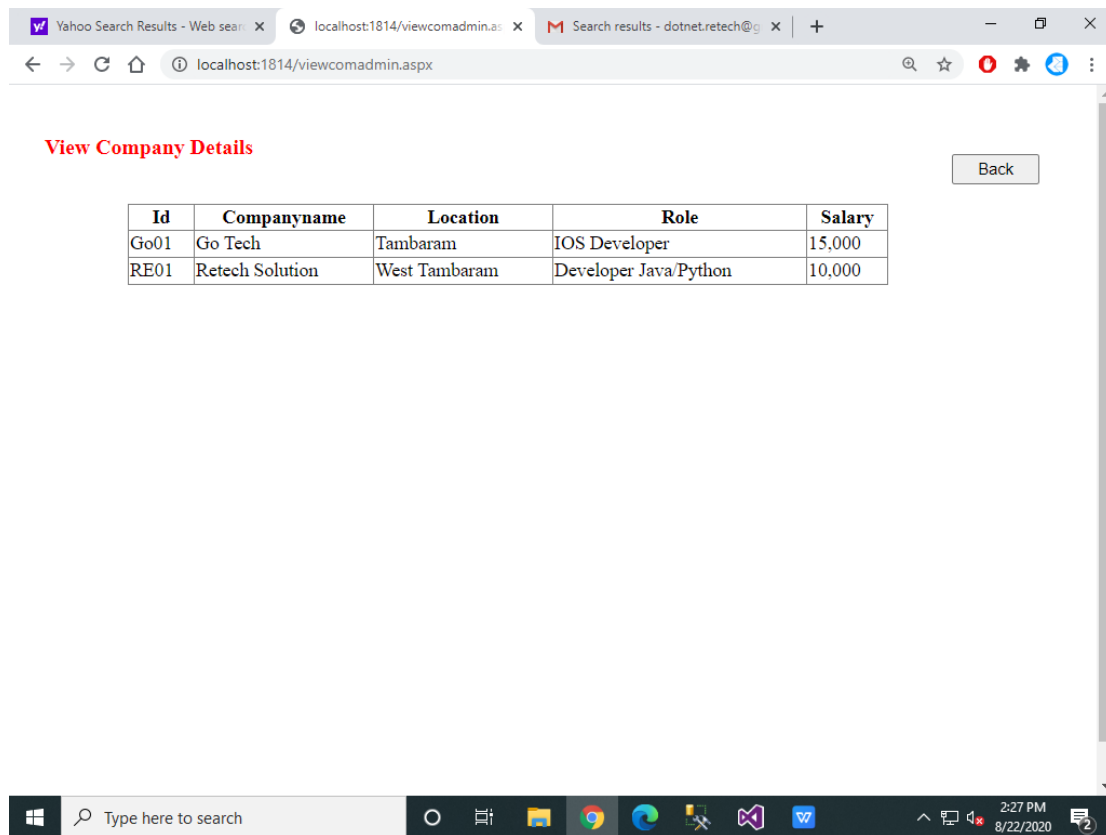
[View Score](#)

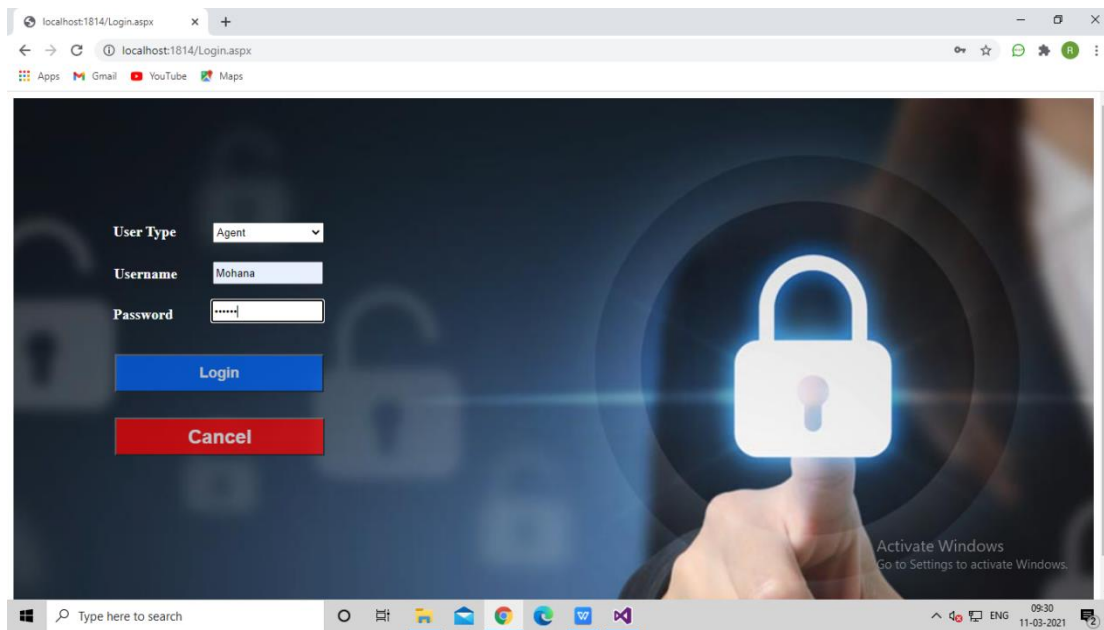
[View Job Required Details](#)

[Send Mail](#)

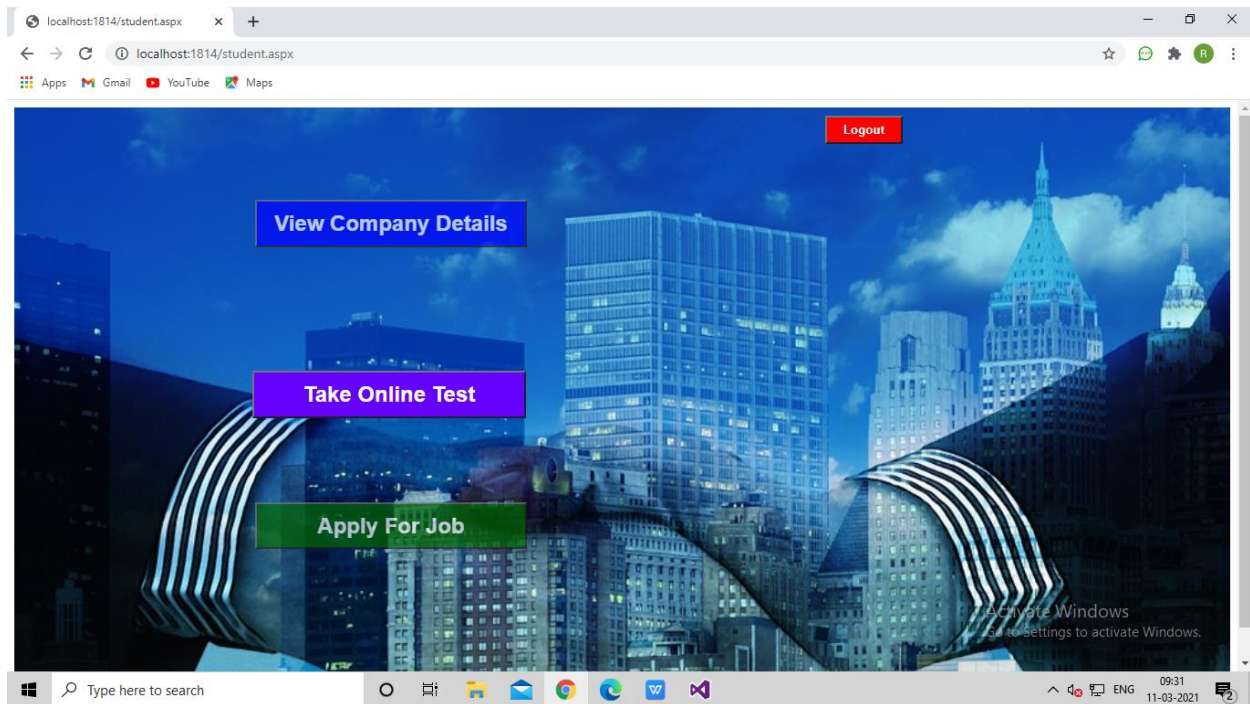


ADMIV VIEW:





AGENT LOGIN:



STUDENT PAGE: