INDIRA GANDHI NATIONAL OPEN UNIVERSITY

MCSP-060

"ONLINE APPLICATION FOR IT INSTITUTE "

By
PYAR MOHAMMAD
ENROLMENT NO: - 166527535

Under Guidance of
UPENDRA KUSIYAIT

Submitted to the School of Computer and Information Sciences, IGNOU
In partial fulfillment of the requirements
For the award of the degree
Master of Computer Applications (MCA)

2021

Indira Gandhi National Open University
Maidan Garhi
New Delhi – 110068.

SYNOPSIS ON

"**ONLINE APPLICATION** FOR IT INSTITUTE"

INDEX

ACKNOWLEDGEMENT

I am very glad to say that my project is completed after some difficulties and hard work. With Candor and Pleasure I take opportunity to express my sincere thanks and obligation to my esteemed guide **UPENDRA KUSIYAIT.** It is because of his able and mature guidance and co-operation without which it would not have been possible for me to complete my project.

It is my pleasant duty to thank all the staff member of the computer center who never hesitated me from time during the project development.

I gratefully acknowledge the support, encouragement & patience of my family and as always, nothing in my life would be possible without God.

Special thanks to **UPENDRA KUSIYAIT** for helping us satisfy our queries regarding project on time.

PYAR MOHAMMAD

INTRODUCTION

"IT Institute of Career Development" provides the basic information related to the Education Program Conducted for the Woman welfare, Providing IT Education to all the Sector of the Society such as for Minorities and Under Privileged people so that anyone who wants to get the information related to admission; courses, fees etc. can view online and search the details of their boys and girls online.

Our project is based on the "A Web application of the Vrinda Educare Institute of Career Development", and it perform all functioning of AICD that is issuing form, admission of student, Counseling, Course Enquiry, Center Allotment, Seat Management etc. are done using this.

In this online functioning all the function of AICD is done through on line .we can submit all the application forms, directly online. It is very less time consuming method and convent to use.

In the off line functioning student suffers from many problems they submitting all the form in the queue. That is very timing process and it is not sufficient for every student and also submitting all the examination forms in a queue but on line reduces this timing process and there is no need of queue for submitting all there.

On line process is helpful to performing easily in the University management; student directly acquired the Registration no, receipt number and the related information very quickly.

Today environment is on line based on which process is very helpful for all the students who lived in different place and submitting the form at a one place directly internet application web page. They directly logon on the web page and acquired all the information very quickly.

SURVEY OF TECHNOLOGY

Front End : C#.Net

Back End : SQL SERVER

Operating System : Windows XP

REASON FOR USING C# .Net: -

1. C#.Net 2.0 is the best front-end tool available for making Windows based application.
2. It is easy to use.
3. .Net is the world's most popular rapid application development tools for creating windows based application.
4. It provides GUI facilities.
5. Ready-made controls make project development easier and faster.
6. Connectivity with the database is easy with the help of ADO.NET.

REASONS FOR USING SQL SERVER: -

1. **It is powerful database.**
2. **It can store large amount of records.**
3. **It provides security features i.e. only authorized users can access the records.**
1. **SQL SERVER has included all standard RDBMS features.**

HTML :-

Hyper Text Markup language is the stander language that the web uses for creating and recognizing documents. Although not a subset of it is loosely related to the stander generalized markup language

(SGML). SGML is a method for representing document Formatting language.HTML is markup language that is used to create an HTML document. The instructions specify how a web page should be displayed in a browser. Using HTML tags and the element, you can-

- Control the appearance of the page and the content.
- Publish on-line documents and retrieve online information using the links inserted in the HTML document.
- Create on-line form.
- Insert object –audio clip, video clips, Active X component java applet etc.

SQL SERVER:-

A database is stored in a very structured manner. Each database requires some way for a user to interact with the information within. Such interaction is performed by a database management system (DBMS). SQL Server is a member of a large category of products known as database management systems (DBMS). The general purpose of a DBMS is to provide for the definition, storage, and management of data in a centralized area that can be shared by many users. SQL Server's database management system is patterned on the relational model.

Querying the database: -With each query of the database, we form a virtual table that contains the results of our query. Database queries are made with a specific language named SQL (structured query language).

SQL Server 2000 has many performance improvements and features which allows us to build and manage large databases, query them fast, insert data into them at high rates, partition them for fast loading and backup and store very large objects or whole files. Central, Shared, Accessible, Backed up, Versioned etc. SQL, as a relational data language supports certain basic functions to control, define and

manipulate data. SQL uses the term row to refer to a database record and the term column to refer to database field

ASP.NET:-

ASP.NET is a prodigal update of Microsoft's Active Server page (asp). ASP.NET is a powerful server based technology designed to create dynamic and interactive HTML pages on demand for your web-site or corporate intranet. Its design improves upon of code you need to write to giving you more power and flexibility.

ASP.NET is a key element in Microsoft powerful .NET development environment. It

allows us to create web application in a new, flexible way by placing commonly used code into reusable controls of various kinds that can fire event initiated by the users of a site.

ASP.NET branches out into many other technologies, such as web services ADO.NET, custom controls and security. The primary goal of ASP.NET is to enable you to build powerful, secure and dynamic application using the list possible amount of code.

Today's World Wide Web consists of a large number of individual web sites that do not co-operate. The .NET platform and its technologies development by Microsoft will enable such co-operation among web –site.

C# :-

C# (pronounced as 'C sharp') is computer programming language developed by Microsoft Corporation, USA. C# is fully object-oriented language like java and is the first component –oriented language. it ha be designed to support the key feature of .NET Framework , the new development platform of Microsoft for building component –based software solution . it is simple ,efficient, productive and

type-safe language derived from the family of C / C++. It is purely object-oriented, modern language suitable for developing web-based application. C# is designed for building

world application. Some major factor:-

- It is only language designed for the .NET Framework.
- It will become the language of choice for .NET programming
- Major parts of .NET Framework are actually coded in C#.
- It brand new language derived from the C / C++ family.
- It simplified and modernizes C++.
- It is the only component –oriented language designed for the .NET Framework.
- It is concise , lean and modern language ,
- It combines the best features of many commonly used languages the productivity of visual basic, the power of C++ and the elegance of java.

PROJECT CATEGORY (WEB-APPLICATION)

Web based systems involve a mixture between print publishing & S/W development between marketing and computing between internal communications and external relations and between art & technology. A web application has following attributes encountered in vast majority.

- **Network intensive:-** By its nature a web application is network intensive. It resides on a network and must serve the needs of diverse community of clients.
- **Content driven:-** In many cases, the primary function of a web application is to use hypermedia to present text, graphics, audio & video content to the end user.
- Continuous evolution:
- **Immediacy:-** Web based applications have on immediacy that is not found in other software that is the time to market for a complete website can be a matter of a few days or weeks.
- **Security:-** Web applications are applicable via networks access, it is difficult, if not impossible, to limit the population of end users who may access the application. In order to protect sensitive content and provide secure modes of data transmission, strong security measures must be implemented throughout the infrastructure that supports a web application & within the application itself.
- **Aesthetics:-** An undeniable part of the appeal of web applications is look &feel for success as technical design.

Feasibility Study:-

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resources.

The objective for this phase is not to solve the problem but to acquire a sense of scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. Information processing systems are capital investments because resources are being spent currently in order to achieve benefits to be received over a period of time following completion. There should be a careful assessment of each project before it is begun in terms of economic justification, technical feasibility, operational impact and adherence to the master development plan.

We started the project by listing the possible queries that the user might want to be satisfied. And on these lines we guided the project further.

The three main points, kept in mind at the time of project, are:

- **Possible (To build it with the given technology and resources)**
- **Affordable (given the time and cost constraints of the organization)**
- **Acceptable (for use by the eventual users of the system)**

The three major areas to consider while determining the feasibility of a project are: ---

i. **Technical Feasibility**
ii. **Economical Feasibility**
iii. **Operational Feasibility**

Technical Feasibility:-

This involves financial consideration to accommodate technical enhancements. If the budget is a serious constraint, then the project is judged not feasible.

The analyst thinks about the technical feasibility of the system. In the proposed system, the application has made in this way so that it is technically feasible for the sales promotion. There are some certain types of reports being used in the system to make the application more users friendly i.e. the Data Report is used. The system has made in such away that it is being used for a single user as well as multi-user environments. The operating systems are Windows NT 4.0/Windows 2000/Windows XP/2003. Thus we see that the system is more technically feasible regarding to operating system. I am using SQL Server as back ends for maintaining the database. SQL Server is a relational database management system of an object. SQL Server is a Windows based RDBMS. This is one of the powerful RDBMS due to its menu driven facility. This provides a better service to the user in the sense of taking the backup of data and then restores it. We can easily mirror the database in the SQL Server.

Economical Feasibility:-

An evaluation of development cost weighed against the ultimate income or benefit derived from the developed system. Today, software is the most expensive element of virtually all computer-based systems. A large cost estimation error can make the difference between profit and loss. Estimation of resources, cost, and schedule for a software engineering effort requires experience, access to good historical information, and the courage to commit to quantitative predictions. Estimation carries inherent risk .and this risk leads to uncertainty. Project complexity has a strong effect of the uncertainty inherent in planning. Complexity, however, is a relative measure that is affected by familiarity with past effort.

Project size is another important factor that can affect the accuracy and efficiency of estimates. As size increases, the inter dependency among various elements of the software grows rapidly. Software cost and effort estimation will never be an exact science. Too many variables-human, technical, environmental, political-can affect the ultimate cost of software and effort applied to develop it. However, software project estimation can be transformed from a black art to a series of systematic steps that provide estimates with acceptable risk. The system is not too costly according to the features of the application. The cost of the project is balanced. The cost of the project might be increase or decrease as according to requirement of the customer. The system has developed systematically.

Operational Feasibility:-

This application is very easy to operate as it is made user friendly. Main consideration is user's easy access to all the functionality of the application.

Documentation of The Feasibility Study:-

The findings of a feasibility study are generally documented in what is called a feasibility report. The degree of detail in such reports would be greatly dependent on the native of the project. The content of this project would be as given in:

I. Introduction
    - Statement of problem
    - Implementation Environment
    - Constraints

II. Management summery and recommendations
    - Important findings

- Comments
- Recommendations
- Impact

III. Alternatives
    - Alternative system configurations
    - Criteria used in selecting the final approach

IV. System description
    - Abbreviated statement of scope.
    - Feasibility of allocated elements

V. Cost Benefit Analysis

VI. Evaluation of technical Risk

VII. Legal ramification's

VIII. Others Project Specific Topics

Hardware Requirements

Processor: P4 or higher

RAM: 512MB

HDD: 1GB of free space

Graphic Card

Software Requirements

Front-end: ASP.NET

Back-end: Microsoft SQL Server

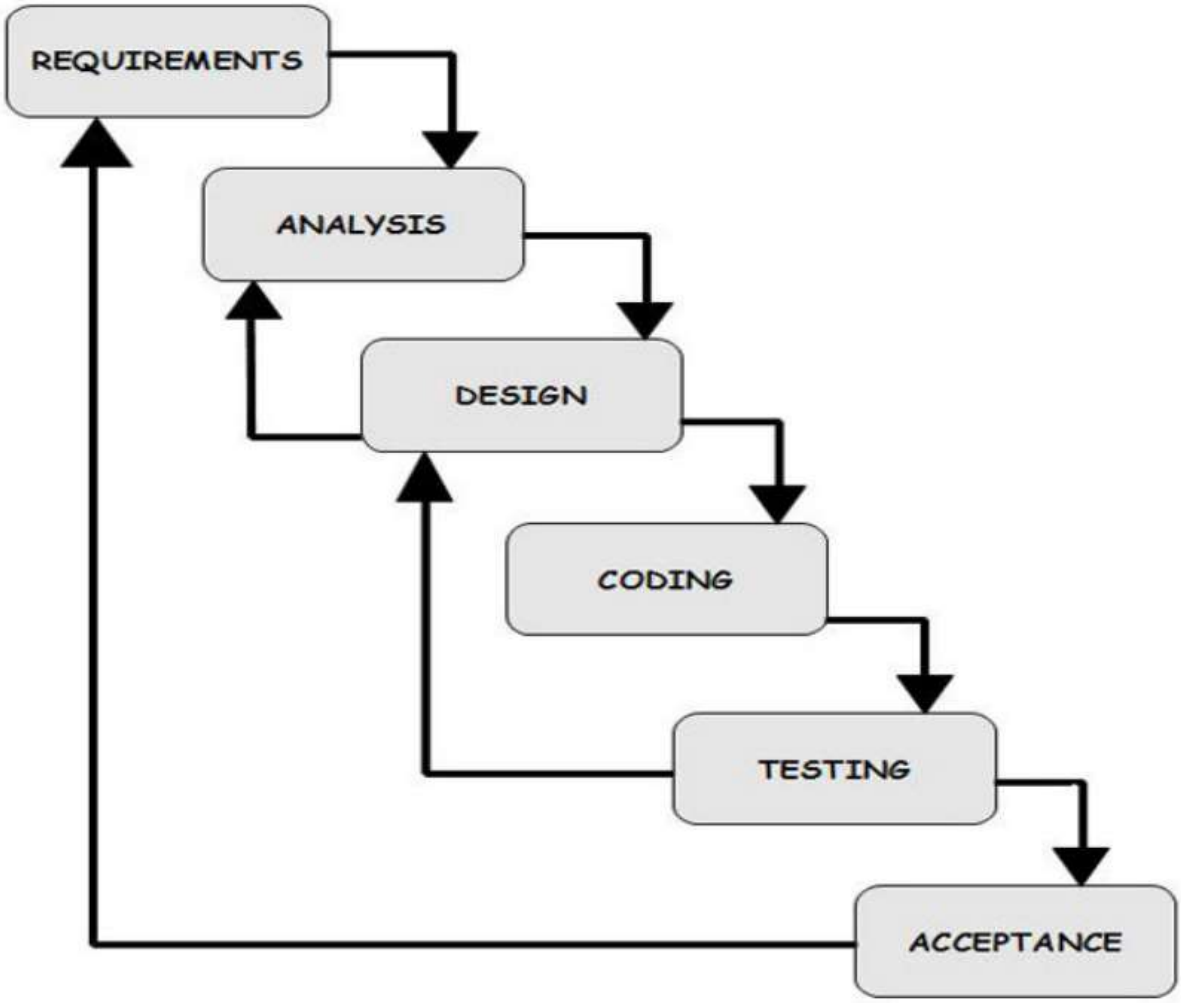
Platform: .NET platform

Operating system: Windows XP or Higher Version

LIFE CYCLE METHODOLOGY

WATERFALL MODEL: - The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin.

This type of model is basically used for the for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete. In waterfall model phases do not overlap.



Planning and scheduling

Planning of this project will include following things:

Topic understanding:-

- Modular break-up of the system
- Processor logic for each module
- Database requirement

It is vital that the field of application as introduced in the project may be totally a new field. So as soon as I took this project, I carefully went through the project to identify the requirements.

Modular break-up of the system:

It consists of following phases:

- Identify the various modules in the system
- List them in the right hierarchy
- Identify their priority of development

PERT Chart, GANTT Chart:-

PERT stands for program evaluation review technique. Unlike bar charts, PERT can be both a cost and a time management system. PERT is organized by events and activities or tasks. PERT has several advantages over bar charts and is likely to be used with more complex projects. One advantage of pert is that it is scheduling device that also shows graphically which task must be completed before others are begun. Also by displaying the various task paths, PERT enables the calculation of critical path. Each path consists of combinations of task, which must be completed. The time and cost associated with each

Task along a path are calculated and the path, which requires greatest amount of time, is critical path. Calculation of the critical path enables project managers to monitor this series of tasks more likely than others and to shift resources to it begins to fall behind schedule. Pert controls time and costs during the project and also facilitates finding the right balance between completing it within the budget. The appropriate way to manage a project is to define and control each task. Because project often falls behind schedule, pert is designed to facilitate getting a project back on schedule. Pert is based in part on the premise that subjective estimates of total completion time for a project usually greatly inferior to the sum of subjective estimates for each task. As with Gantt charts, to build a pert chart for a project, one must determine the dependence of the activities required for completion of the project and estimate how long each will take. Then one must determine the dependence of the activities on each other. In facts, the pert chart gives a graphical representation of this information. Clearly this technique does not help in deciding which activities are necessary or how long each will take, but it does force the manager to take the necessary planning steps to answer these questions. Pert chart are available in next page:

Some of the advantages of pert chart

It forces the manager to plan.

- ✓ It shows the interrelationships among the task in the project.
- ✓ It clearly identifies critical path of the project.
- ✓ It exposes all possible parallelism in the activities and thus helps in allocating .
- ✓ It allows scheduling and simulation of alternative schedules.
- ✓ It enables the manager to monitor and control the project.

Gantt Chart

Gantt chart is also known as Bar chart Hennery ***L. Gantt*** has developed Gantt chart. Gantt charts are a project control technique that can be used for several purposes, including scheduling, budgeting and resources planning. A Gantt chart is a bar chart, with each bar representing an activity. The bars are drawn against a time line. The length of each bar is proportional to the length of time planned for the activity. A Gantt charts help in scheduling the activities of a project, but it does not help in identifying them. One can begin with activities identified in work breakdown structure as we did below for quality management project during the scheduling activity, and also during implementation of the project quality management system, new activities may be identified that were not envisioned during the initial planning. The project manager must then go back and revise the breakdown structure and the schedules to deal with news activities. The Gantt chart in the figure is actually an enhanced version of standard Gantt charts. In below figure of Gantt chart, horizontal bars have been shown the duration of actions or tasks. The white bar shows the Slack time that is, latest time by which a task must be finished. Gantt chart, in which, each bar would be represented on of the engineers. Gantt charts are useful for resource, planning and scheduling. When a bar chart is used as a project control method, milestones or checkpoints usually are placed at the completion of each task. They indicate the completion of a particular task and are the basis for determining whether the task and the project are on schedule. Reviewer can ask whether resources allocated have been properly utilized and whether the task has been satisfactorily completed. However, because the bar chart incorporates only the scheduling dimension of a project, it gives little indication of which tasks must be completed before others are begun, and projects cost must be accumulated and evaluated using other method

Phase	Work Task	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
1	Feasibility study										
	Problem Definition										
	System specification										
	S/W Specification										
2	System Design										
	Data Design										
	Input Output Design										
	Functional Design										
	Procedural Design										
3	Database Creation										
	Main Interface & Menu										
	Forms										
	Report										
3	Unit Testing										
	Integration testing										
	Validation Testing										
	System testing										
5	Documentation										

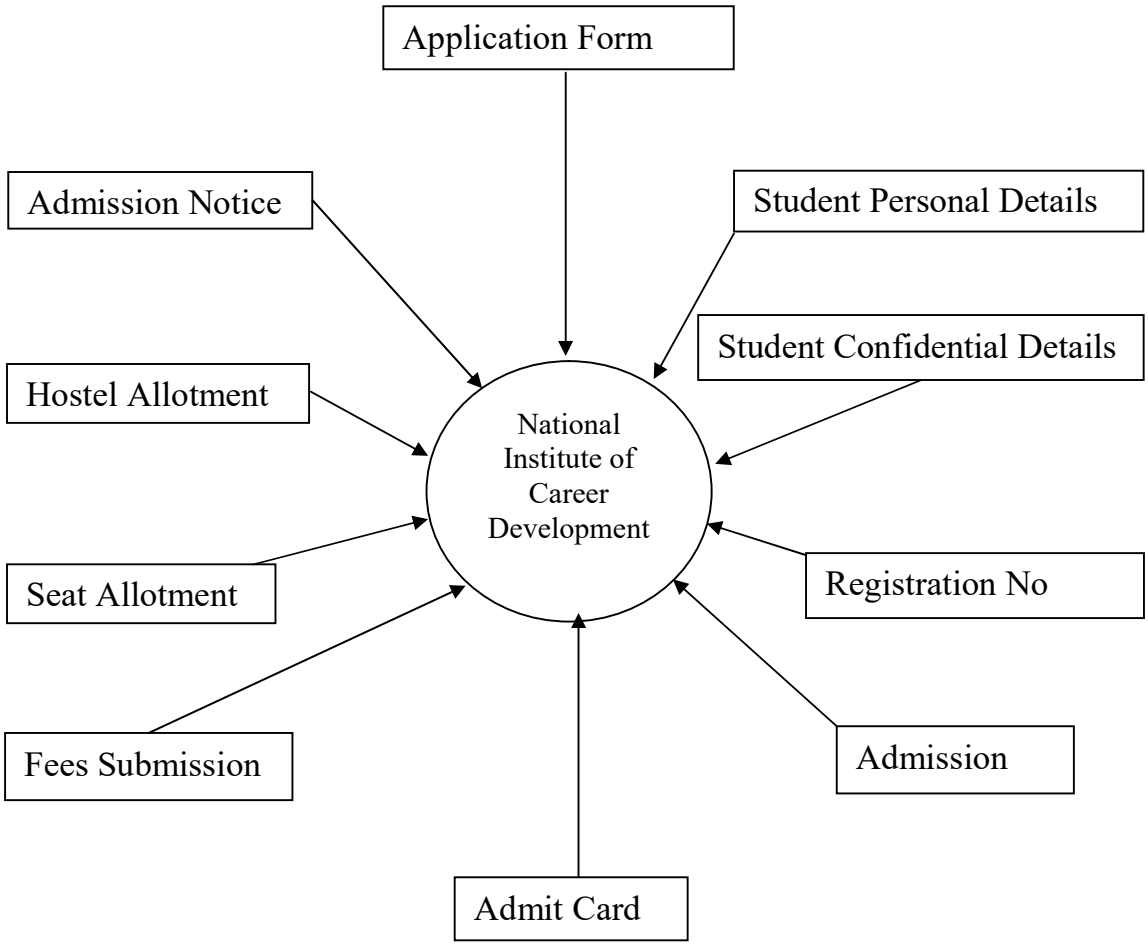
Module of the project:-

There are eight modules in this project

- **Sales of application form** - Firstly user fill all details about him like name, father name, mother name, dob etc and also fill up details of payment like DD No., DD Date, Bank Name from where he make his Drafter this a user name and a password is provided by admin to the user. Where User Name is the Registration No and User Password is the DOB of the user. Through this User Name and Password user can enter in this website.
- **Check the feasibility or minimum criteria** -In this module admin Check status.
- **Issuing the admit card-** After checking application form , admin will provide admit card to the user. This admit card will be printed by user to enter in examination hall. Without this they cannot give examination.
- **On line merit list** – After completed examination, user can see their marks and also their rank through subject wise.
- **On line Counseling –** In this module online counseling was done by admin. In this user has choice to select minimum five colleges.
- **Allotment of the college** – In this module on the basis of vacant seat preferred college was allotted to the user.
- **College Fee Submission –** After this user/student go to that allotted college and submit their fees.
- **Hostel Fees Submission –** In this module student submit their hostel fees.

Data Flow Diagram

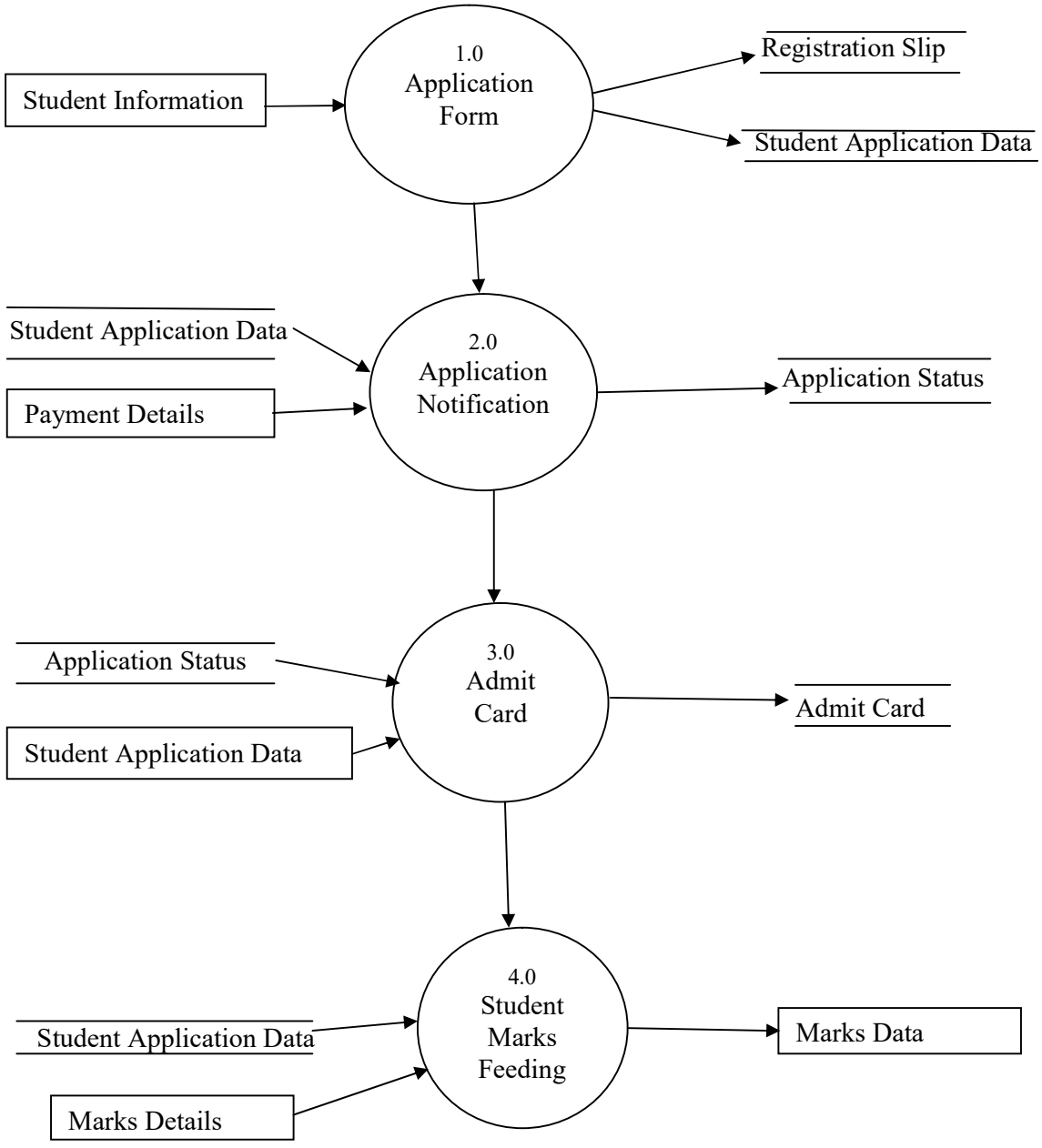
0 Level



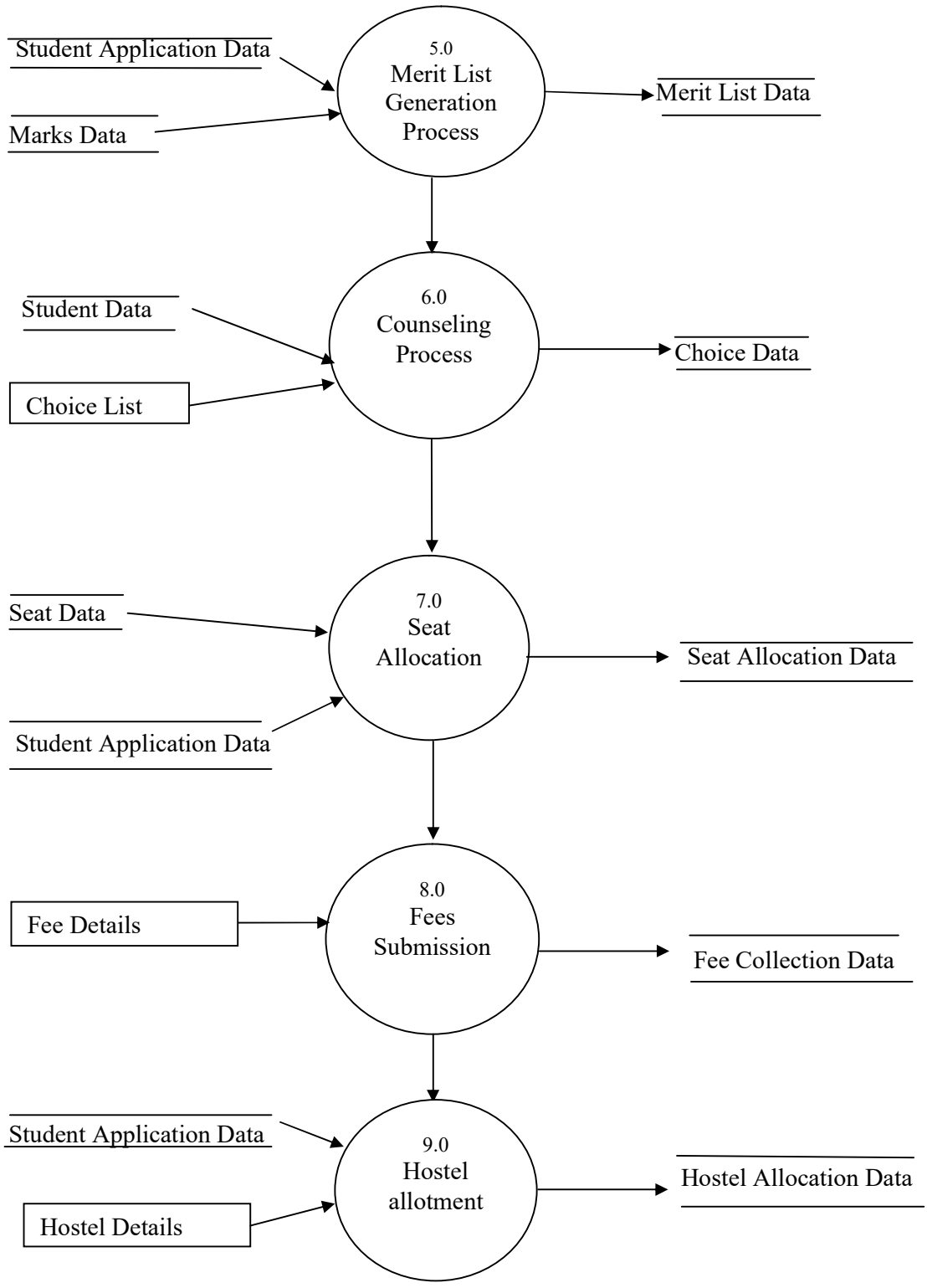
\

\

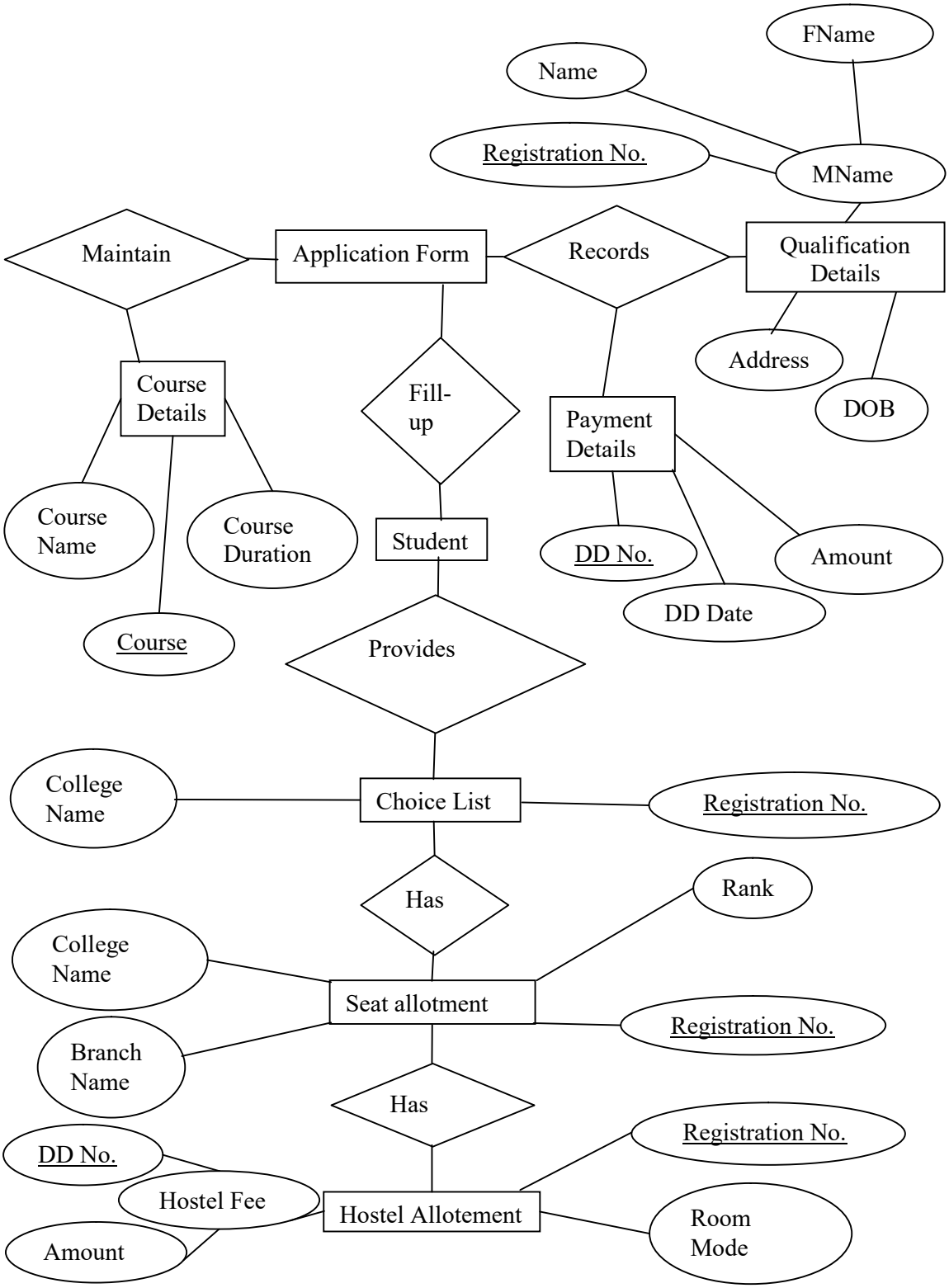
First Level



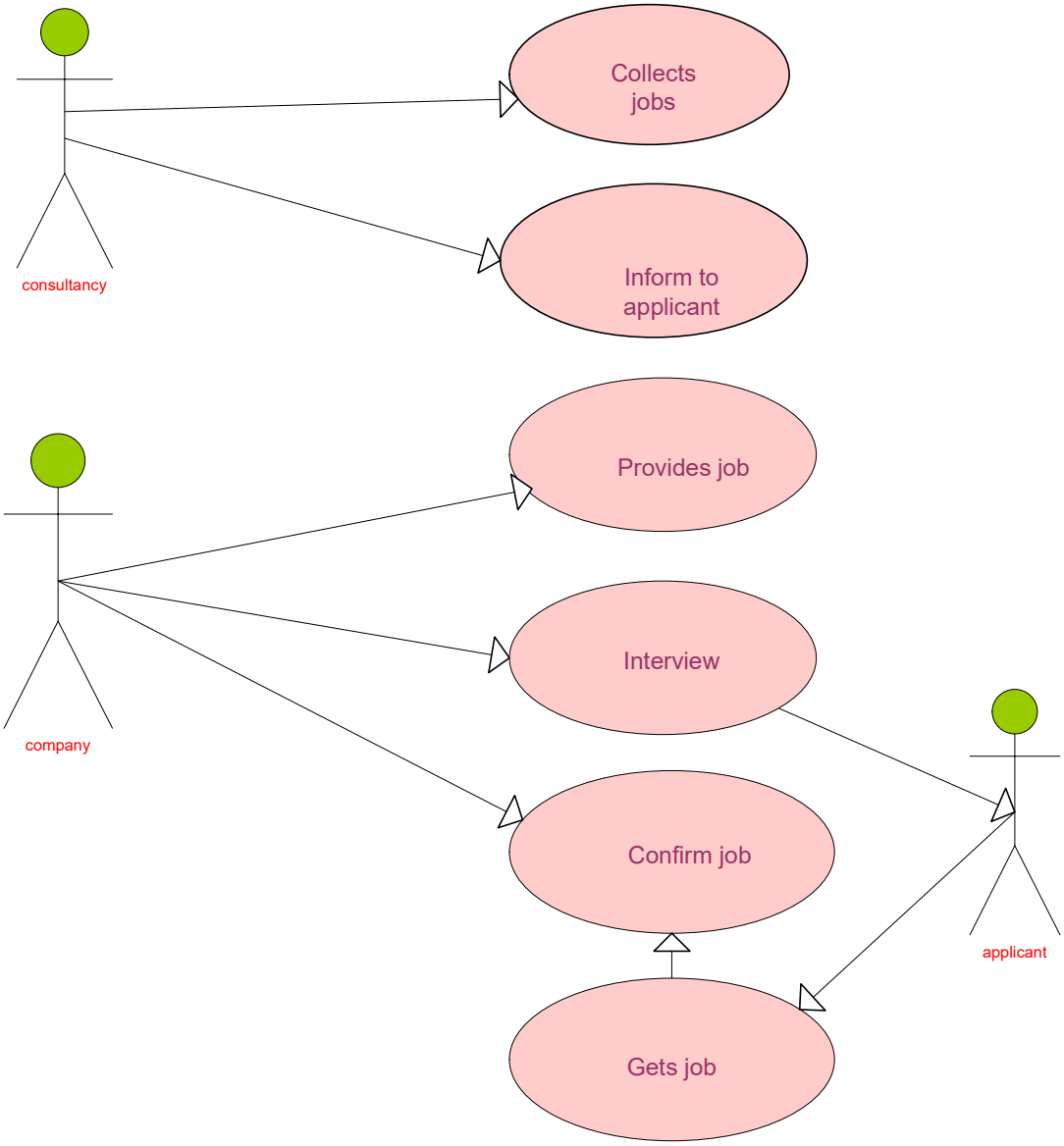
-- Continued --


Student Application Data
Marks Data
5.0
Merit List Generation Process
Merit List Data
Student Data
Choice List
6.0
Counseling Process
Choice Data
Seat Data
Student Application Data
7.0
Seat Allocation
Seat Allocation Data
Fee Details
8.0
Fees Submission
Fee Collection Data
Student Application Data
Hostel Details
9.0
Hostel allotment
Hostel Allocation Data

ER Diagram


FName
Name
Registration No.
MName
Maintain
Application Form
Records
Qualification Details
Address
DOB
Course Details
Fill-up
Payment Details
Course Name
Course Duration
Course
Student
DD No.
Amount
DD Date
Provides
College Name
Choice List
Registration No.
Has
Rank
College Name
Seat allotment
Registration No.
Branch Name
Has
Registration No.
DD No.
Hostel Fee
Hostel Allotement
Room Mode
Amount

USE CASE DIAGRAM



USE CASE DIAGRAM

System Design

Software design is a multi step process, which focuses on distinct attributes of program: data structure, software architecture, interface representation, and procedural detail. The design process translates requirements into a representation of the software that can be assessed for quality before coding begins. According to requirement, the design is documented.

The design must be translated into a machine-readable form. The code generation step perform this task .If design is performed in a detailed manner.

Design phase is the first step while moving from the problem domain to solution domain. This phase begins when the requirement specification document for the newly developing system is available. The aim of this phase is to produce a model or a representation of the system, which is useful to have an overall look of the system without even developing the entire system. This model of the system is a plan for developing the system. There are three characteristics for the evaluation of a good design.

1. The design should implement all the requirements of the user, which are specified in the SRS and it must accommodate all the requirements that may not be specified by the client in the analysis phase.
2. The design should be readable and understandable for the people who involve in developing and testing the system.
3. The design should provide a complete picture of the system, addressing the data functional domains from the implementation perspective.

Design Methodology :-

When solving a small problem, the entire problem can be tackled at once. The complexity of large problems and limitations of human minds do not allow large problems to be treated as huge monoliths. For software design, the goal is to divide the problem into manageably small phases that can be solved separately. However, the different phases cannot be entirely independent of each other, as they together form the system. The different phases have to co-operate and communicate in order to solve the larger problem.

Database Design:-

A database management system (DBMS) consists of a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information about one particular enterprise. The primary goal of a DBMS is to provide an environment that is both convenient and efficient to use in retrieving and sorting database information. Database systems are designed to manage large bodies of information. The management of data involves both the definition of storage of information and the provision of mechanisms for the manipulation of information.

We used relational database management system (RDBMS) for developing this system.

The goal of a relational- database management system (RDBMS) design is to generate a set of relation schema that allows us to store information without unnecessary redundancy. It also allows us to retrieve information easily. **Redundancy** means repetition of information i.e., same information may be written or stored in many places (files). This redundancy may lead to data inconsistency i.e., the various copies of the same data. When we access this inconsistent data, system may give wrong information. To reduce the data redundancy we use the concept of normalization.

Normalization of data is a process in which unsatisfactory relation schema are decomposed by breaking up their attribute into smaller relation schema that possess desirable properties. Normal forms provide a formal framework for analyzing relation schema based on their keys and the functional dependencies among attributes to database designers. System level issues are addressed using three activities: design, assembly, and use. The component dimension is addressed with two activities: design and realization. Concurrency is achieved in two ways:

1) System and activities occur simultaneously and can be modeled using the state oriented approach described previously.

2) A typical client server application is implemented with many components, each of which can be designed and realized concurrently.

Introduction to data dictionary:-

Data dictionaries are an integral component of structured analysis, since data flow diagrams by themselves do not fully describe the subject of the investigation/. The data flow diagrams provide the additional details about the project/system.

Data Dictionary (Definition):-

A data dictionary is a catalog- a repository- the elements in a system. These elements center on the data and the way they are structured to meet user requirements and organization needs. A data dictionary consists of a list of all the elements composing the data flowing through a system. The major elements are data flow, data stores, and processes. The data dictionary stores details and descriptions of these elements.

Describing Data Elements:-

Each entry in the data dictionary consists of a set of details describing the data used or produced in the system. Each item is identified by a data name, description, alias, and length and has specific values that are permissible for it in the system being studied.

DATA DICTIONARY

Tables

Application Form 1 :-

SN.	Field Name	Data Type	Size	Key
1	*FullName*	*varchar*	*50*	
2	*FatherName*	*varchar*	*50*	
3	*MotherName*	*varchar*	*50*	
4	*PemanentAddress*	*varchar*	*50*	
5	*PinCode*	*int*	*4*	
6	*State*	*varchar*	*50*	
7	*City*	*varchar*	*50*	
8	*Mobile No*	*varchar*	*50*	
9	*DOB*	*varchar*	*50*	
10	*Email*	*varchar*	*50*	
11	*RegistrationNo*	*numeric*	*9*	*PK*
12	*ApplicationStatus*	*varchar*	*50*	

Application Form 2:-

SN.	Field Name	Data Type	Size	Key
1	*Qualifyingexam*	*varchar*	*10*	
2	*NameofUniversity*	*varchar*	*50*	
3	*YearofPassing*	*varchar*	*50*	

4	*DDNumber*	*varchar*	*50*	
5	*DDDate*	*varchar*	*50*	
6	*Per_Marks*	*int*	*9*	
7	*RegistrationNo*	*int*	*9*	
8	*amount*	*Int*	*4*	
9	*Center*	*varchar*	*50*	
10	*Course*	*varchar*	*50*	

Choice:-

SN.	Field Name	Data Type	Size	Key
1	*RegistrationNo*	*decimal*	*9*	*PK*
2	*College_name1*	*varchar*	*50*	
3	*College_name2*	*varchar*	*50*	
4	*College_name3*	*varchar*	*50*	
5	*College_name4*	*varchar*	*50*	
6	*College_name5*	*varchar*	*50*	

Colleges:-

SN.	Field Name	Data Type	Size	Key
1	*CollegeName*	*varchar*	*50*	
2	*Branch*	*varchar*	*50*	
3	*TotalSeats*	*Int*	*4*	

Fillup Form:-

SN.	Field Name	Data Type	Size	Key
1	*RegistrationNo*	*int*	*9*	*PK*
2	*Name*	*varchar*	*50*	

3	*FatherName*	*varchar*	*50*	
4	*Stream*	*varchar*	*50*	
5	*TotalMarks*	*int*	*4*	
6	*Rank*	*int*	*4*	

List of Branches:-

SN.	Field Name	Data Type	Size	Key
1	*[S.No.]*	*Int*	*4*	*PK*
2	*Bname*	*varchar*	*100*	*PK*
3	*Bcode*	*varchar*	*50*	*PK*

Seat Allotment:-

SN.	Field Name	Data Type	Size	Key
1	*RegistrationNo*	*int*	*9*	*PK*
2	*Stream*	*varchar*	*50*	
3	*College*	*varchar*	*50*	

LIMITATION OF THE PROJECT

1. It is an intranet application and only authorized user can access the sites.

2. People who are not familiar with computer can't use this software.

3. No user can login or access the sites without the valid user name and password.

4. Maintenance cost is high because of its distribution over large network.

5. Some of the tabs are not working because their no need for need for them at this stage.

FUTURE SCOPE OF SYSTEM

After sufficient volume of data is stored in the system, the data mining and data warehousing techniques should be employed for information analysis and decision making

- Demand forecasting based on the available data.
- Decision support oriented reports
- Any type of report that not been generated earlier for generating new hypothesis using the available data.
- Loan provisions and recovery

Manpower analysis and deployment. The further scope of the project is very good. With a few of more changes this project can be deployed in the real life environment. This Project will reduce the scope of error, the workload and time; Increase efficiency taken to generate reports. The automated system will also help management to keep its data up to date.

Further Enhancement:-

This project will have bright future in which payment procedure will be performed through credit card this leads to better money transaction.

There may be online career counseling; it will help the applicant to select the better admissions. There will be online interview facility it will save time and money of Student and College.

Advantages of the proposed system:-

The user can access the system from anywhere .If the user wants any query about the clients, applicants or requirement then a few keystrokes on a computer keyboard by a operator can avail all the details helping him/her not to wander department to department to access the details. The paper work is greatly minimized. No training is given to the users (operators). With the interface a layman can have pleasant experience working with the application.
A detailed analysis of software requirements would provide necessary information for estimates, but analysis often takes weeks or months to complete. Therefore, We must examine the product and the problem it is intended to solve at very beginning of the project, At a minimum, the scope of the product must be established and bounded. Scope is defined by answering the following questions-

Context- How does the software to be built into a larger system, product, or business context and what constraints are imposed as a result of the context?

Information Objectives– What customer–visible data objects are produced as output from the software? What data objects are required for input?

Function and Performance– What function does the software perform to transform input data into output? Is any special performance? In recent times in India and also other parts of the world record handling and maintaining the accuracy have become cumbersome processes. This software will assist the concerned client in maintaining their records and searching proper matching as per requirement only at mouse click. Manually find out the certain records in a very tedious. Cumbersome and risky job. This software can well assist in proper Real Estate Records Management.

Bibliography

system analysis and Design-Elias M. Awad

ASP.Net Bible- Mridula Parihar

Wrox-Professional SQL Server-Louis Davidson

HTML 4.0-E.Stephen Mack and Janan Platt

By Kevin Loney and George Koch

Roger S. Pressman, "Software Engineering (A Practitioner's Approach)",

McGraw-Hill 2000

MCS -032 Object Oriented Analysis and Design

MCS-034 Software Engineering

MCS-043 Advance Database Management System

PROJECT REPORT

TABLE OF CONTENTS

INTRODUCTION

"IT Institute of Career Development" provides the basic information related to the Education Program Conducted for the Woman welfare, Providing IT Education to all the Sector of the Society such as for Minorities and Under Privileged people so that anyone who wants to get the information related to admission; courses, fees etc. can view online and search the details of their boys and girls online.

Our project is based on the "A Web application of the Vrinda Educare Institute of Career Development", and it perform all functioning of AICD that is issuing form, admission of student, Counseling, Course Enquiry, Center Allotment, Seat Management etc. are done using this. In this online functioning all the function of AICD is done through on line .we can submit all the application forms, directly online. It is very less time consuming method and convent to use. In the off line functioning student suffers form many problems they submitting all the form in the queue. That is very timing process and it is not sufficient for every student and also submitting all the examination forms in a queue but on line reduces this timing process and there is no need of queue for submitting all there.

Online process is helpful to performing easily in the University management; student directly acquired the Registration no, receipt number and the related information very quickly.

Today environment is on line based on which process is very helpful for all the students who lived in different place and submitting the form at a one place directly internet application webpage

SYSTEM ANALYSIS

The questionnaires, interviews and informal discussion with the employee's allowed me to understand the needs of the users.

Reason For Analysis:-

The recent expansion has increased the frequency of book issue and return, which has in turn aided to the complexity of hosting them. The current manual system is not fulfilling the information needs of those, who are responsible for implementing the management. Various analysis tools such as DFD are used for analysis.

Scope of system Analysis:-

The system analysis work will determine the feasibility and direction of the information needs as outlined in the system plan and the information system services request form .The findings of system analysis will be subsequently recorded in the system analysis completion report. We will be responsible for the delivery of this report within weeks from beginning of the analysis. Responsibility for system design will be determined at the end of this system analysis phase.

SURVEY OF TECHNOLOGY

Front End : C#.Net

Back End : SQL SERVER

Operating System : Windows XP

REASON FOR USING C# .Net: -

7. C#.Net 2.0 is the best front-end tool available for making Windows based application.
8. It is easy to use.
9. .Net is the world's most popular rapid application development tools for creating windows based application.
10. It provides GUI facilities.
11. Ready-made controls make project development easier and faster.
12. Connectivity with the database is easy with the help of ADO.NET.

REASONS FOR USING SQL SERVER: -

1. It is powerful database.

2. It can store large amount of records.

3. It provides security features i.e. only authorized users can access the records.

2. SQL SERVER has included all standard RDBMS features.

HTML:-

Hyper Text Markup language is the stander language that the web uses for creating and recognizing documents. Although not a subset of it is loosely related to the stander generalized markup language (SGML). SGML is a method for representing document

Formatting language.HTML is markup language that is used to create an HTML document. The instructions specify how a web page should be displayed in a browser. Using HTML tags and the element, you can-

- Control the appearance of the page and the content.
- Publish on-line documents and retrieve online information using the links inserted in the HTML document.
- Create on-line form.
- Insert object –audio clip, video clips, Active X component java applet etc.

SQL SERVER:-

A database is stored in a very structured manner. Each database requires some way for a user to interact with the information within. Such interaction is performed by a database management system (DBMS). SQL Server is a member of a large category of products known as database management systems (DBMS). The general purpose of a DBMS is to provide for the definition, storage, and management of data in a centralized area that can be shared by many users. SQL Server's database management system is patterned on the relational model.

Querying the database: - With each query of the database, we form a virtual table that contains the results of our query. Database queries are made with a specific language named SQL (structured query language).

SQL Server 2000 has many performance improvements and features which allows us to build and manage large databases, query them fast, insert data into them at high rates, partition them for fast loading and backup and store very large objects or whole files. Central, Shared, Accessible, Backed up, Versioned etc. SQL, as a relational data language supports certain

basic functions to control, define and manipulate data. SQL uses the term row to refer to a database record and the term column to refer to database field

ASP.NET:-

ASP.NET is a prodigal update of Microsoft's Active Server page (asp). ASP.NET is a powerful server based technology designed to create dynamic and interactive HTML pages on demand for your web-site or corporate intranet. Its design improves upon of code you need to write to giving you more power and flexibility.

ASP.NET is a key element in Microsoft powerful .NET development environment. It allows us to create web application in a new, flexible way by placing commonly used code into reusable controls of various kinds that can fire event initiated by the users of a site.

ASP.NET branches out into many other technologies, such as web services ADO.NET, custom controls and security. The primary goal of ASP.NET is to enable you to build powerful, secure and dynamic application using the list possible amount of code.

Today's World Wide Web consists of a large number of individual web sites that do not co-operate. The .NET platform and its technologies development by Microsoft will enable such co-operation among web –site.

C# :-

C# (pronounced as 'C sharp') is computer programming language developed by Microsoft Corporation, USA. C# is fully object-oriented language like java and is the first component –oriented language. it ha be designed to support the key feature of .NET Framework , the new development platform of Microsoft for building component –based software solution . it is simple ,efficient, productive and type-safe language derived from the family of C / C++. It is purely object-oriented, modern language suitable for developing web-based application. C# is designed for building world application. Some major factor:-

- It is only language designed for the .NET Framework.
- It will become the language of choice for .NET programming
- Major parts of .NET Framework are actually coded in C#.
- It brand new language derived from the C / C++ family.
- It simplified and modernizes C++.
- It is the only component –oriented language designed for the .NET Framework.
- It is concise , lean and modern language ,
- It combines the best features of many commonly used languages the productivity of visual basic, the power of C++ and the elegance of java.

PROJECT CATEGORY (WEB-APPLICATION)

Web based systems involve a mixture between print publishing & S/W development between marketing and computing between internal communications and external relations and between art & technology. A web application has following attributes encountered in vast majority.

- **Network intensive:-** By its nature a web application is network intensive. It resides on a network and must serve the needs of diverse community of clients.
- **Content driven:-** In many cases, the primary function of a web application is to use hypermedia to present text, graphics, audio & video content to the end user.
- Continuous evolution:
- **Immediacy:-** Web based applications have on immediacy that is not found in other software that is the time to market for a complete website can be a matter of a few days or weeks.
- **Security:-** Web applications are applicable via networks access, it is difficult, if not impossible, to limit the population of end users who may access the application. In order to protect sensitive content and provide secure modes of data transmission, strong security measures must be implemented throughout the infrastructure that supports a web application & within the application itself.
- **Aesthetics:-** An undeniable part of the appeal of web applications is look &feel for success as technical design.

Feasibility Study:-

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. Feasibility study is a test of system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resources. The objective for this phase is not to solve the problem but to acquire a sense of scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. Information processing systems are capital investments because resources are being spent currently in order to achieve benefits to be received over a period of time following completion. There should be a careful assessment of each project before it is begun in terms of economic justification, technical feasibility, operational impact and adherence to the master development plan.

We started the project by listing the possible queries that the user might want to be satisfied. And on these lines we guided the project further.

The three main points, kept in mind at the time of project, are:

- **Possible (To build it with the given technology and resources)**
- **Affordable (given the time and cost constraints of the organization)**
- **Acceptable (for use by the eventual users of the system)**

The three major areas to consider while determining the feasibility of a project are: ---

iv. **Technical Feasibility**

v. **Economical Feasibility**

vi. **Operational Feasibility**

Technical Feasibility:-

This involves financial consideration to accommodate technical enhancements. If the budget is a serious constraint, then the project is judged not feasible.

The analyst thinks about the technical feasibility of the system. In the proposed system, the application has made in this way so that it is technically feasible for the sales promotion. There are some certain types of reports being used in the system to make the application more users friendly i.e. the Data Report is used. The system has made in such away that it is being used for a single user as well as multi-user environments. The operating systems are Windows NT 4.0/Windows 2000/Windows XP/2003. Thus we see that the system is more technically feasible regarding to operating system. I am using SQL Server as back ends for maintaining the database. SQL Server is a relational database management system of an object. SQL Server is a Windows based RDBMS. This is one of the powerful RDBMS due to its menu driven facility. This provides a better service to the user in the sense of taking the backup of data and then restores it. We can easily mirror the database in the SQL Server.

Economical Feasibility:-

An evaluation of development cost weighed against the ultimate income or benefit derived from the developed system. Today, software is the most expensive element of virtually all computer-based systems. A large cost estimation error can make the difference between profit and loss. Estimation of resources, cost, and schedule for a software engineering effort requires experience, access to good historical information, and the courage to commit to quantitative predictions. Estimation carries inherent risk .and this risk leads to uncertainty. Project complexity has a strong effect of the uncertainty inherent in planning. Complexity, however, is a relative measure that is affected by familiarity with past effort. Project size is another important factor that can affect the accuracy and efficiency of estimates. As size

increases, the inter dependency among various elements of the software grows rapidly. Software cost and effort estimation will never be an exact science. Too many variables-human, technical, environmental, political-can affect the ultimate cost of software and effort applied to develop it. However, software project estimation can be transformed from a black art to a series of systematic steps that provide estimates with acceptable risk. The system is not too costly according to the features of the application. The cost of the project is balanced. The cost of the project might be increase or decrease as according to requirement of the customer. The system has developed systematically.

Operational Feasibility:-

This application is very easy to operate as it is made user friendly. Main consideration is user's easy access to all the functionality of the application.

Documentation of The Feasibility Study:-

The findings of a feasibility study are generally documented in what is called a feasibility report. The degree of detail in such reports would be greatly dependent on the native of the project. The content of this project would be as given in:

IX. Introduction
    - Statement of problem
    - Implementation Environment
    - Constraints

X. Management summery and recommendations
    - Important findings
    - Comments
    - Recommendations

- Impact

XI. Alternatives

- Alternative system configurations
- Criteria used in selecting the final approach

XII. System description

- Abbreviated statement of scope.
- Feasibility of allocated elements

XIII. Cost Benefit Analysis

XIV. Evaluation of technical Risk

XV. Legal ramification's

XVI. Others Project Specific Topics

Hardware Requirements

Processor: P4 or higher

RAM: 512MB

HDD: 1GB of free space

Graphic Card

Software Requirements

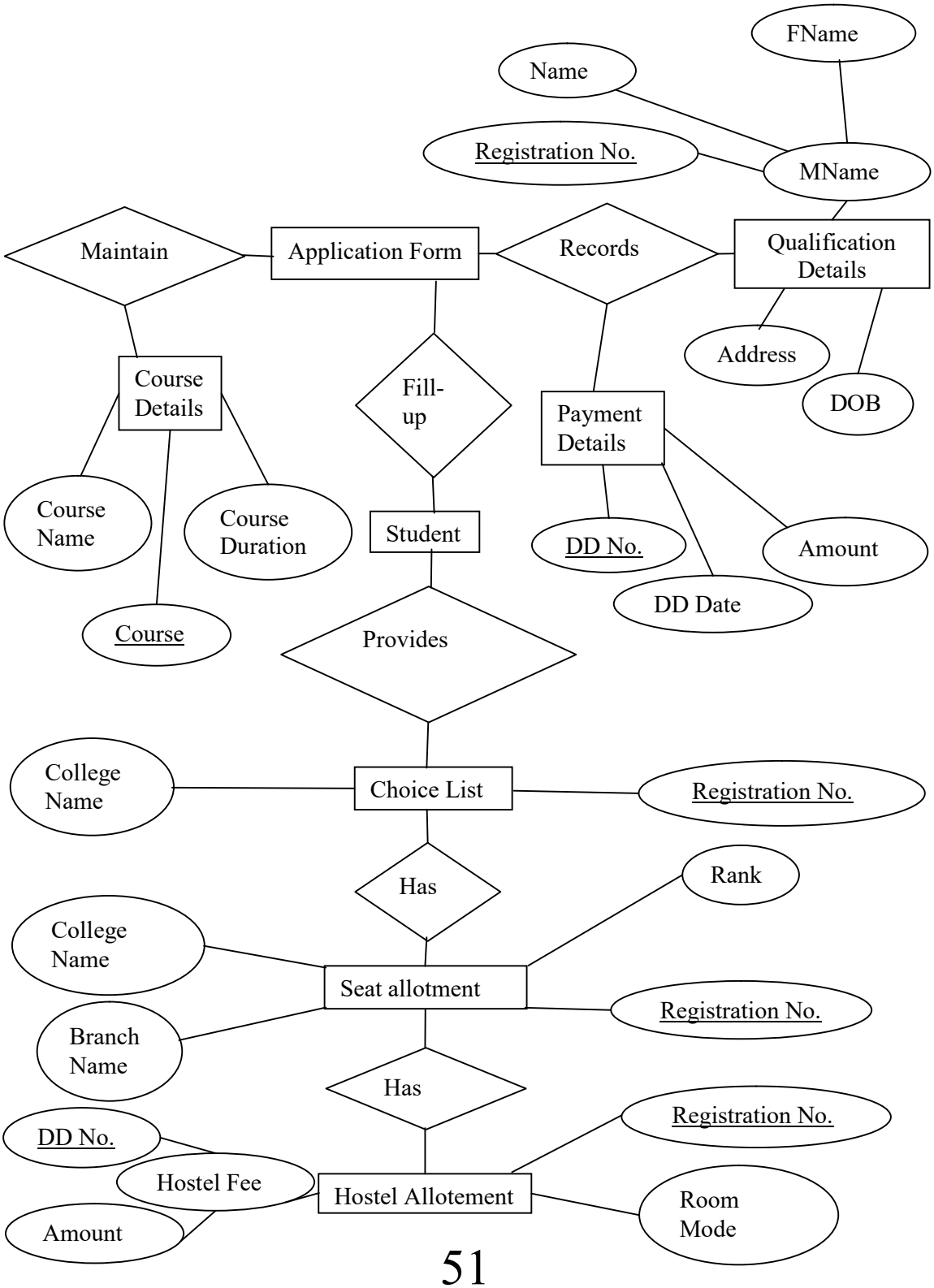
Front-end: ASP.NET

Back-end: Microsoft SQL Server

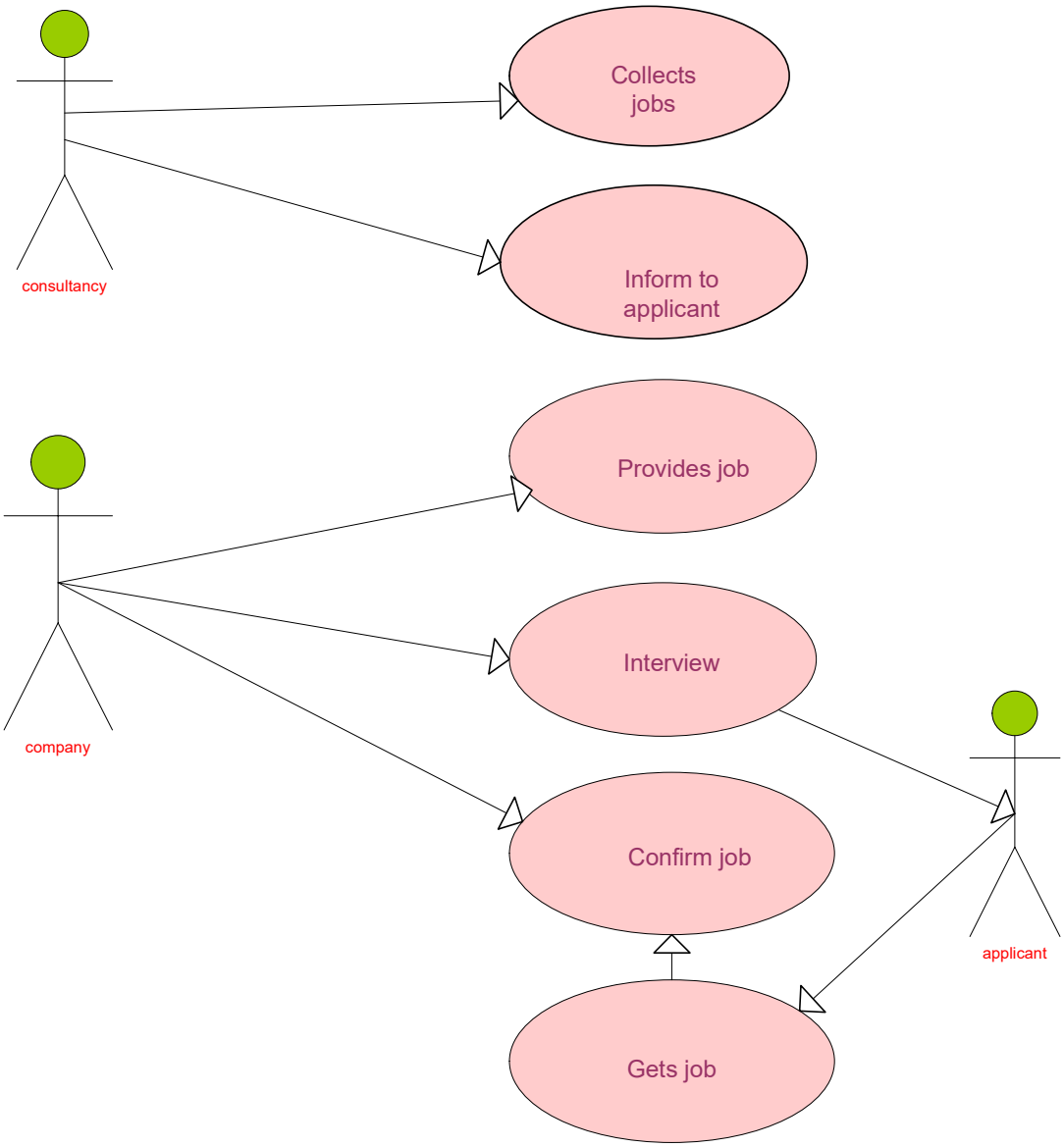
Platform: .NET platform

Operating system: Windows XP or Higher Version

ER Diagram


FName
Name
Registration No.
MName
Maintain
Application Form
Records
Qualification Details
Address
DOB
Course Details
Fill-up
Payment Details
Course Name
Course Duration
Course
Student
DD No.
Amount
DD Date
Provides
College Name
Choice List
Registration No.
Has
Rank
College Name
Seat allotment
Registration No.
Branch Name
Has
DD No.
Registration No.
Hostel Fee
Hostel Allotement
Room Mode
Amount

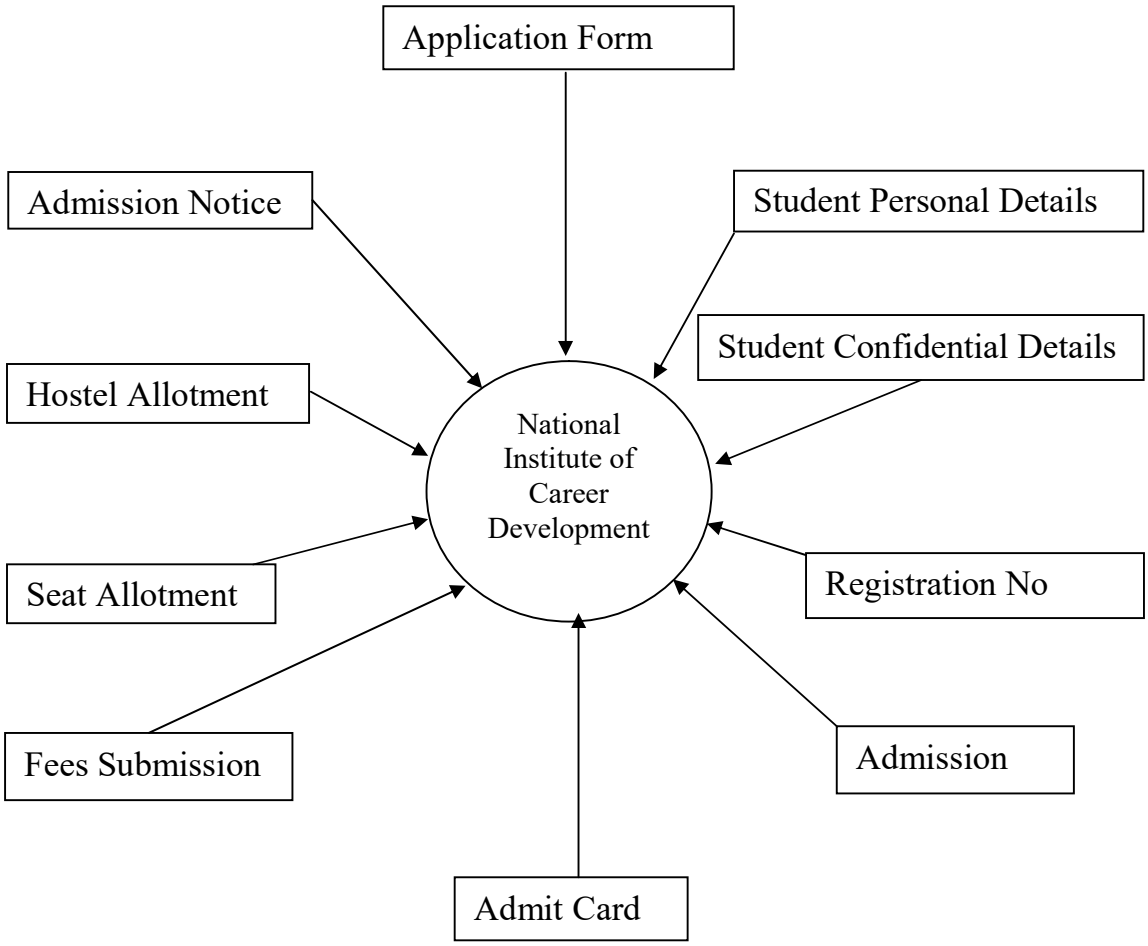
USE CASE DIAGRAM



USE CASE DIAGRAM

Data Flow Diagram

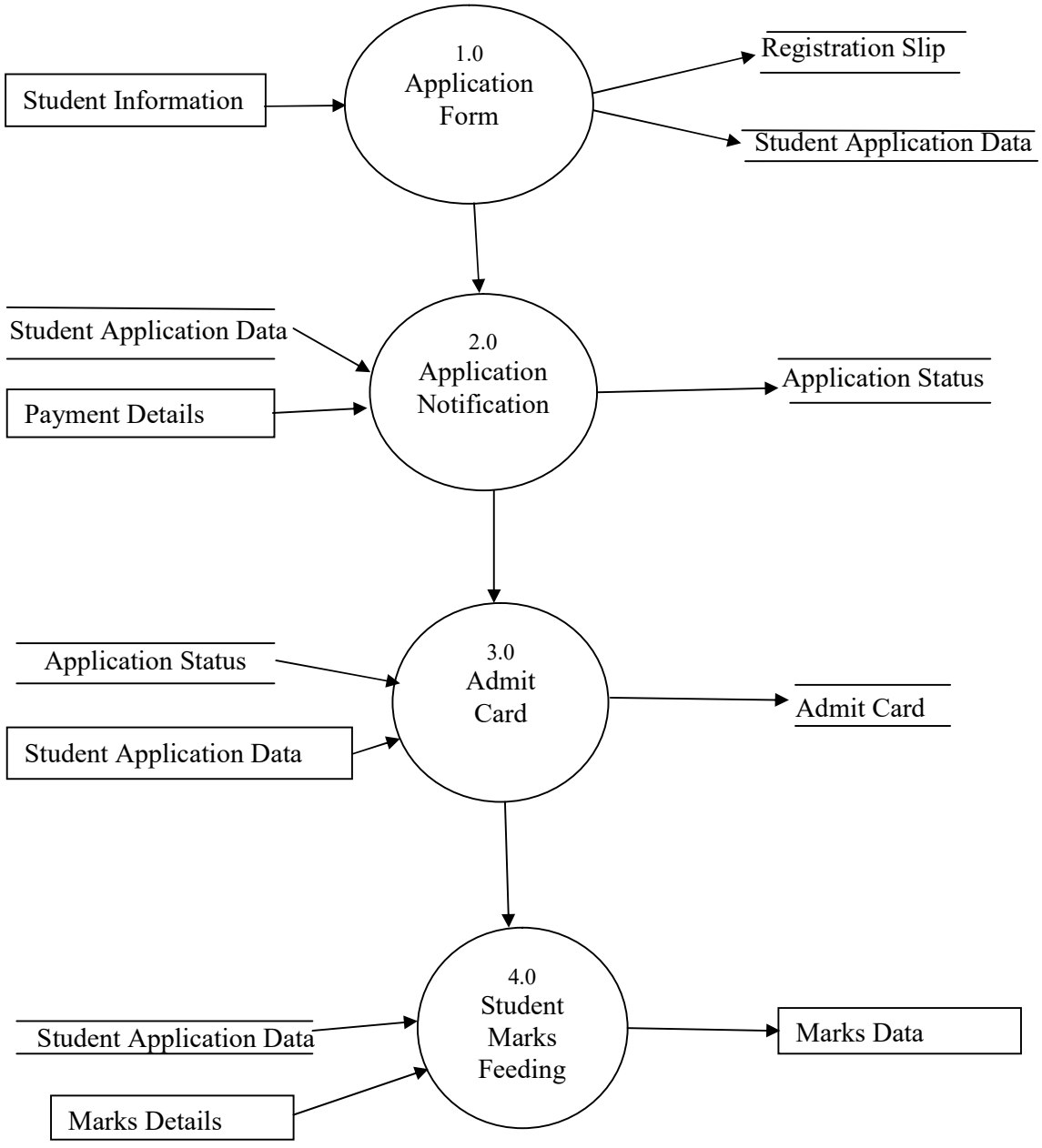
0 Level



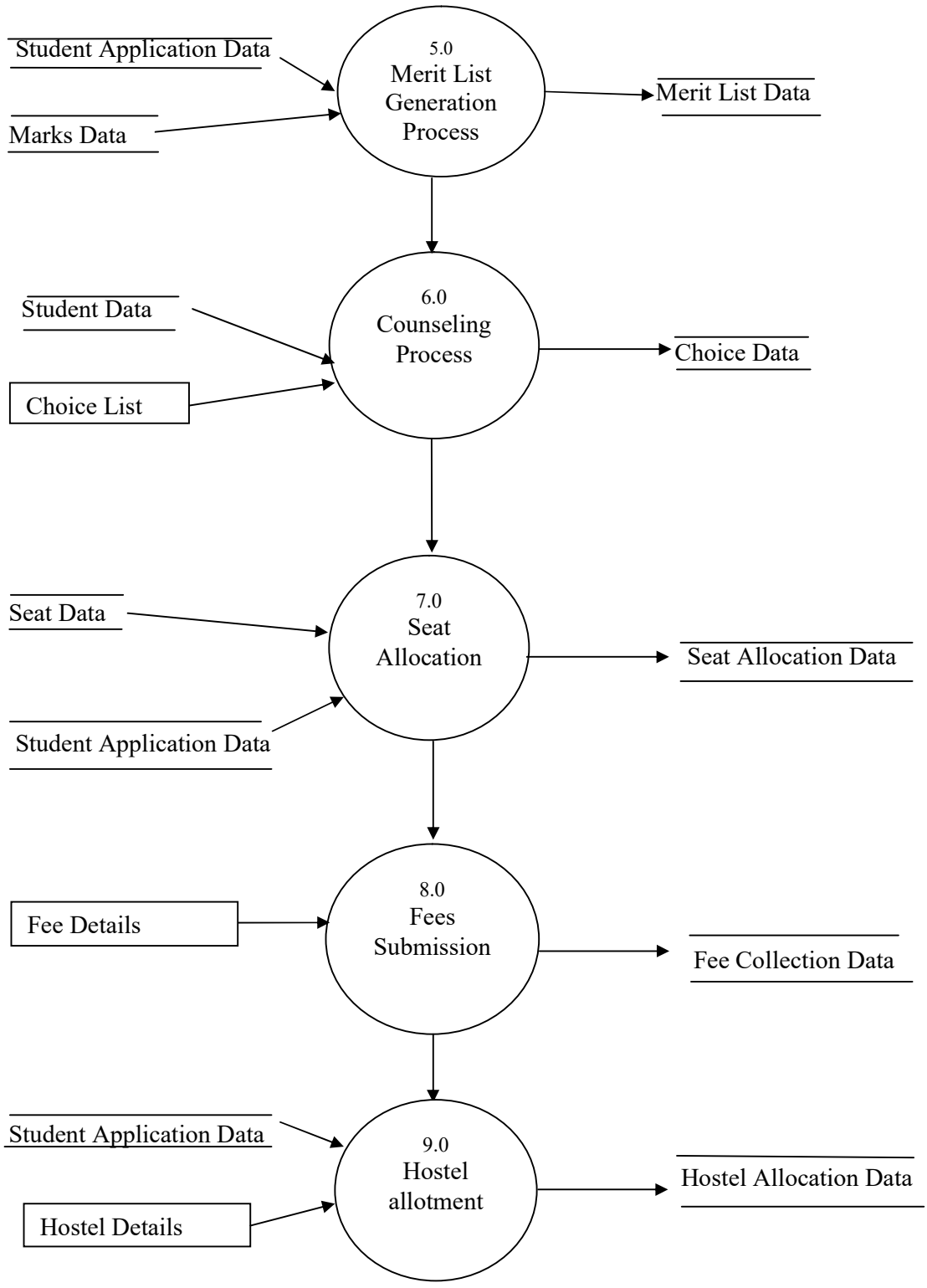
\

\

First Level



-- Continued --


Student Application Data
Marks Data
5.0
Merit List Generation Process
Merit List Data
Student Data
Choice List
6.0
Counseling Process
Choice Data
Seat Data
Student Application Data
7.0
Seat Allocation
Seat Allocation Data
Fee Details
8.0
Fees Submission
Fee Collection Data
Student Application Data
Hostel Details
9.0
Hostel allotment
Hostel Allocation Data

System Design

Software design is a multi step process, which focuses on distinct attributes of program: data structure, software architecture, interface representation, and procedural detail. The design process translates requirements into a representation of the software that can be assessed for quality before coding begins. According to requirement, the design is documented.

The design must be translated into a machine-readable form. The code generation step perform this task .If design is performed in a detailed manner.

Design phase is the first step while moving from the problem domain to solution domain. This phase begins when the requirement specification document for the newly developing system is available. The aim of this phase is to produce a model or a representation of the system, which is useful to have an overall look of the system without even developing the entire system. This model of the system is a plan for developing the system. There are three characteristics for the evaluation of a good design.

1. The design should implement all the requirements of the user, which are specified in the SRS and it must accommodate all the requirements that may not be specified by the client in the analysis phase.
2. The design should be readable and understandable for the people who involve in developing and testing the system.
3. The design should provide a complete picture of the system, addressing the data functional domains from the implementation perspective.

Design Methodology:-

When solving a small problem, the entire problem can be tackled at once. The complexity of large problems and limitations of human minds do not allow large problems to be treated as huge monoliths. For software design, the goal is to divide the problem into manageably small phases that can be solved separately. However, the different phases cannot be entirely independent of each other, as they together form the system. The different phases have to co-operate and communicate in order to solve the larger problem.

Database Design:-

A database management system (DBMS) consists of a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information about one particular enterprise. The primary goal of a DBMS is to provide an environment that is both convenient and efficient to use in retrieving and sorting database information. Database systems are designed to manage large bodies of information. The management of data involves both the definition of storage of information and the provision of mechanisms for the manipulation of information.

We used relational database management system (RDBMS) for developing this system.

The goal of a relational- database management system (RDBMS) design is to generate a set of relation schema that allows us to store information without unnecessary redundancy. It also allows us to retrieve information easily. **Redundancy** means repetition of information i.e., same information may be written or stored in many places (files). This redundancy may lead to data inconsistency i.e., the various copies of the same data. When we access this inconsistent data, system may give wrong information. To reduce the data redundancy we use the concept of normalization.

Normalization of data is a process in which unsatisfactory relation schema are decomposed by breaking up their attribute into smaller relation schema that possess desirable properties. Normal forms provide a formal framework for analyzing relation schema based on their keys and the functional dependencies among attributes to database designers. System level issues are addressed using three activities: design, assembly, and use. The component dimension is addressed with two activities: design and realization. Concurrency is achieved in two ways:

1) System and activities occur simultaneously and can be modeled using the state oriented approach described previously.

2) A typical client server application is implemented with many components, each of which can be designed and realized concurrently.

Introduction to data dictionary:-

Data dictionaries are an integral component of structured analysis, since data flow diagrams by themselves do not fully describe the subject of the investigation/. The data flow diagrams provide the additional details about the project/system.

Data Dictionary (Definition):-

A data dictionary is a catalog- a repository- the elements in a system. These elements center on the data and the way they are structured to meet user requirements and organization needs. A data dictionary consists of a list of all the elements composing the data flowing through a system. The major elements are data flow, data stores, and processes. The data dictionary stores details and descriptions of these elements.

Describing Data Elements:-

Each entry in the data dictionary consists of a set of details describing the data used or produced in the system. Each item is identified by a data name, description, alias, and length and has specific values that are permissible for it in the system being studied.

Tables Used

Application Form 1 :-

SN.	Field Name	Data Type	Size	Key
1	*FullName*	*varchar*	*50*	
2	*FatherName*	*varchar*	*50*	
3	*MotherName*	*varchar*	*50*	
4	*PemanentAddress*	*varchar*	*50*	
5	*PinCode*	*int*	*4*	
6	*State*	*varchar*	*50*	
7	*City*	*varchar*	*50*	
8	*Mobile No*	*varchar*	*50*	
9	*DOB*	*varchar*	*50*	
10	*Email*	*varchar*	*50*	
11	*RegistrationNo*	*numeric*	*9*	*PK*
12	*ApplicationStatus*	*varchar*	*50*	

Application Form 2:-

SN.	Field Name	Data Type	Size	Key
1	*Qualifyingexam*	*varchar*	*10*	
2	*NameofUniversity*	*varchar*	*50*	
3	*YearofPassing*	*varchar*	*50*	
4	*DDNumber*	*varchar*	*50*	
5	*DDDate*	*varchar*	*50*	
6	*Per_Marks*	*int*	*9*	

7	*RegistrationNo*	*int*	*9*	
8	*amount*	*Int*	*4*	
9	*Center*	*varchar*	*50*	
10	*Course*	*varchar*	*50*	

Choice:-

SN.	Field Name	Data Type	Size	Key
1	*RegistrationNo*	*decimal*	*9*	*PK*
2	*College_name1*	*varchar*	*50*	
3	*College_name2*	*varchar*	*50*	
4	*College_name3*	*varchar*	*50*	
5	*College_name4*	*varchar*	*50*	
6	*College_name5*	*varchar*	*50*	

Colleges:-

SN.	Field Name	Data Type	Size	Key
1	*CollegeName*	*varchar*	*50*	
2	*Branch*	*varchar*	*50*	
3	*TotalSeats*	*Int*	*4*	

Fillup Form:-

SN.	Field Name	Data Type	Size	Key
1	*RegistrationNo*	*int*	*9*	*PK*
2	*Name*	*varchar*	*50*	
3	*FatherName*	*varchar*	*50*	
4	*Stream*	*varchar*	*50*	
5	*TotalMarks*	*int*	*4*	

6	*Rank*	*int*	*4*	

List of Branches:-

SN.	Field Name	Data Type	Size	Key
1	*[S.No.]*	*Int*	*4*	*PK*
2	*Bname*	*varchar*	*100*	*PK*
3	*Bcode*	*varchar*	*50*	*PK*

Seat Allotment:-

SN.	Field Name	Data Type	Size	Key
1	*RegistrationNo*	*int*	*9*	*PK*
2	*Stream*	*varchar*	*50*	
3	*College*	*varchar*	*50*	


CODING

SQL:-

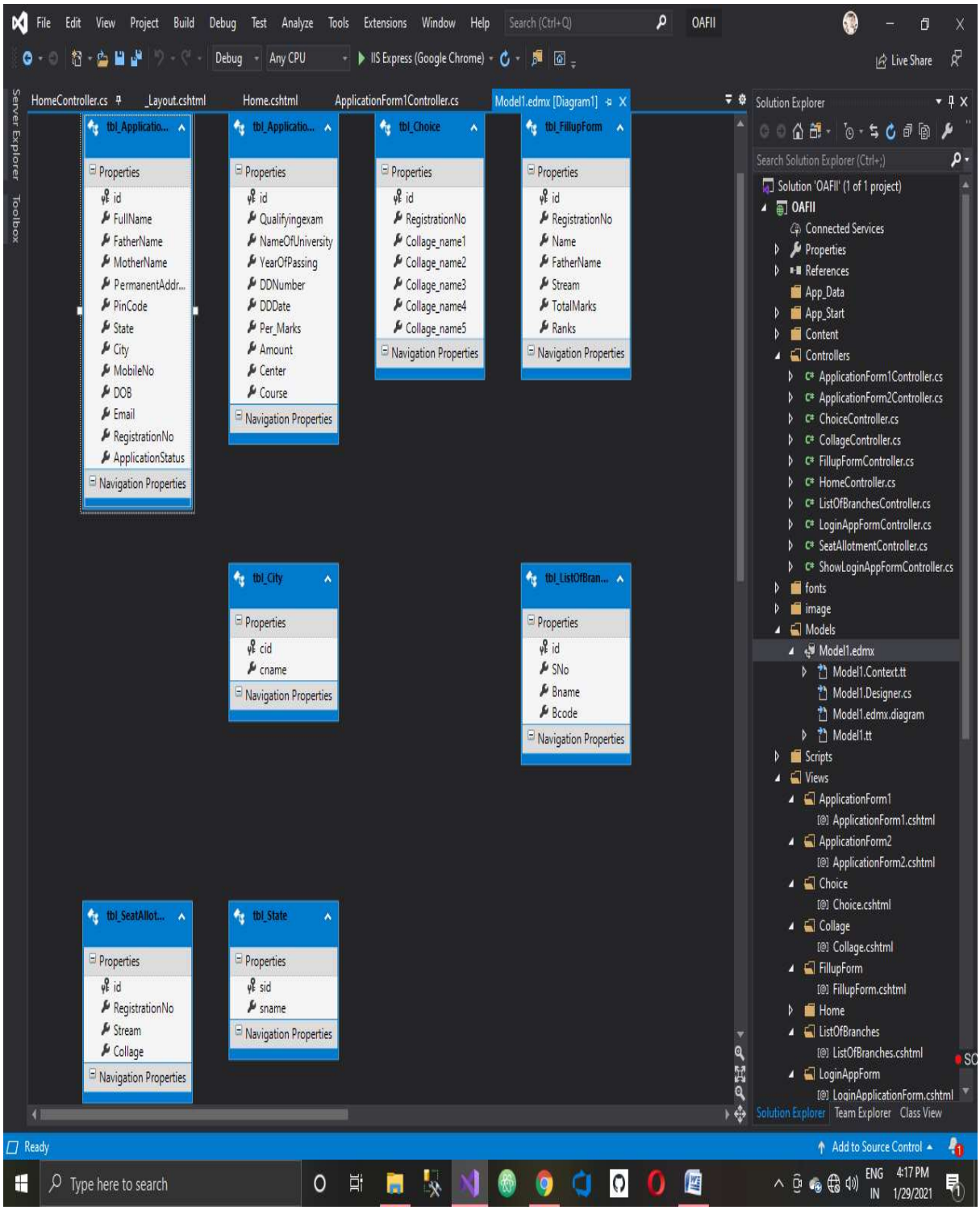
```
create database OAFII
use OAFII


create table tbl_ApplicationForm1
(
  id int primary key identity,
  FullName varchar(50),
  FatherName varchar(50),
  MotherName varchar(50),
  PermanentAddress varchar(50),
  PinCode int,
  State varchar(50),
  City varchar(50),
  MobileNo int,
  DOB varchar(50),
  Email varchar(50),
  RegistrationNo int,
  ApplicationStatus varchar(50)
)

create table tbl_State
(
sid int primary key identity,
sname varchar(50)
)

insert into tbl_State(sname)values('Andhra
Pradesh'),('Arunachal
```

```
Pradesh'),('Assam'),('Bihar'),('Chhattisgarh'),('Goa'),('Gujarat'),
('Haryana'),('Himachal
Pradesh'),('Jharkhand'),('Karnataka'),('Kerala'),('Madhya
Pradesh'),('Maharashtra'),('Manipur'),('Punjab'),('Rajasthan'),('
Uttar Pradesh')

select*from tbl_State

create table tbl_City
(
cid int primary key identity,
cname varchar(50)
)

insert into
tbl_City(cname)values('Mumbai'),('Delhi'),('Bangalore'),('Hydera
bad'),('Ahmedabad'),('Chennai'),('Kolkata'),('Surat'),('Pune'),('J
aipur'),('Lucknow'),('Bhopal'),('Visakhapatnam'),('Patna'),('Ran
chi'),('Srinagar'),('Raipur'),('Chandigarh')

select*from tbl_City

create table tbl_ApplicationForm2
(
  id int primary key identity,
  Qualifyingexam varchar(50),
  NameOfUniversity varchar(50),
  YearOfPassing int,
  DDNumber int,
  DDDate varchar(50),
  Per_Marks varchar(50),
  Amount int,
```

```
  Center varchar(50),
  Course varchar(50)
)

create table tbl_Choice
(
  id int primary key identity,
  RegistrationNo int,
  Collage_name1 varchar(50),
  Collage_name2 varchar(50),
  Collage_name3 varchar(50),
  Collage_name4 varchar(50),
  Collage_name5 varchar(50)
)

create table tbl_FillupForm
(
  id int primary key identity,
  RegistrationNo int,
  Name varchar(50),
  FatherName varchar(50),
  Stream varchar(50),
  TotalMarks int,
  Ranks int
)


create table tbl_SeatAllotment
(
  id int primary key identity,
  RegistrationNo int,
  Stream varchar(50),
```

```
  Collage varchar(50)
)
select *from tbl_ApplicationForm1
select *from tbl_ApplicationForm2
select *from tbl_Choice
select *from tbl_FillupForm
select *from tbl_SeatAllotment
select *from tbl_State
select *from tbl_City
```

Model:-


File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
OAFII
Debug
Any CPU
IIS Express (Google Chrome)
Live Share
HomeController.cs
_Layout.cshtml
Home.cshtml
ApplicationForm1Controller.cs
Model1.edmx [Diagram1]
tbl_Applicatio...
Properties
id
FullName
FatherName
MotherName
PermanentAddr...
PinCode
State
City
MobileNo
DOB
Email
RegistrationNo
ApplicationStatus
Navigation Properties
tbl_Applicatio...
id
Qualifyingexam
NameOfUniversity
YearOfPassing
DDNumber
DDDate
Per_Marks
Amount
Center
Course
tbl_Choice
RegistrationNo
Collage_name1
Collage_name2
Collage_name3
Collage_name4
Collage_name5
tbl_FillupForm
Name
FatherName
Stream
TotalMarks
Ranks
tbl_City
cid
cname
tbl_ListOfBran...
SNo
Bname
Bcode
tbl_SeatAllot...
Collage
tbl_State
sid
sname
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'OAFII' (1 of 1 project)
Connected Services
References
App_Data
App_Start
Content
Controllers
ApplicationForm1Controller.cs
ApplicationForm2Controller.cs
ChoiceController.cs
CollageController.cs
FillupFormController.cs
HomeController.cs
ListOfBranchesController.cs
LoginAppFormController.cs
SeatAllotmentController.cs
ShowLoginAppFormController.cs
fonts
image
Models
Model1.edmx
Model1.Context.tt
Model1.Designer.cs
Model1.edmx.diagram
Model1.tt
Scripts
Views
ApplicationForm1
ApplicationForm1.cshtml
ApplicationForm2
ApplicationForm2.cshtml
Choice
Choice.cshtml
Collage
Collage.cshtml
FillupForm
FillupForm.cshtml
Home
ListOfBranches
ListOfBranches.cshtml
LoginAppForm
LoginApplicationForm.cshtml
Solution Explorer
Team Explorer
Class View
Ready
Add to Source Control
Type here to search
ENG
IN
4:17 PM
1/29/2021

HomeController.cs:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace OAFII.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Home()
        {
            return View();
        }
    }
}
```

Home.cshtml

```
@{
    ViewBag.Title = "Home";
}

<img src="~/image/p1.jpg" width="1300px" height="515px" />
```

LoginAppFormController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using OAFII.Models;

namespace OAFII.Controllers
{
    public class LoginAppFormController : Controller
    {
        OAFIIEntities1 db = new OAFIIEntities1();
```

```
        public ActionResult LoginApplicationForm()
        {
            return View();
        }
        public JsonResult LoginAppForm(tbl_ApplicationForm1 obj)
        {
            var data = (from x in db.tbl_ApplicationForm1 where
x.RegistrationNo == obj.RegistrationNo && x.Email == obj.Email
select x).ToList();
            return Json(data, JsonRequestBehavior.AllowGet);
        }
    }
}
```

LoginApplicationForm.cshtml:-

```
@{
    ViewBag.Title = "LoginApplicationForm";
}

<script src="~/LoginAppForm.js"></script>
<script src="~/LoginAppFormvalidation.js"></script>

<table style="background-color:blue; color:white; text-align:center"
width="26.4%">
    <tr>
        <br />
        <br />
        <br />
        <td colspan="2">
            LoginApplicationForm
        </td>
    </tr>
</table>

<table style="background-color:maroon; color:white">
    <tr>
        <td>Registration Number :</td>
        <td>
            <input type="text" id="txtreg" placeholder="Registration
No." />
        </td>
    <tr>
    <tr>
        <td>Email Id :</td>
        <td>
```

```
            <input type="text" id="txtemail" placeholder="Email Id"
/>
        </td>
    </tr>
    <tr>
        <td></td>
        <td>
            <input type="button" id="btnsubmit" value="LOGIN"
onclick="return Validation()" />
            <!-- <a href="../Reg/RegIndex"><input type="button"
id="btnsignup" value="SignUp" onclick="Signup()" /></a> -->
        </td>
    </tr>
</table>

<br /><br /><br />
```

LoginAppForm.js:-

```
function SaveData() {
    $.ajax({
        url: '../LoginAppForm/LoginAppForm',
        data: { RegistrationNo: $("#txtreg").val(), Email:
$("#txtemail").val() },
        success: function (data) {
            if (data.length > 0) {
                window.location.href =
"../ShowLoginAppForm/ShowLoginAppForm?QS=" + data[0].id;
            }
            else {
                alert("LoginApplicationForm1 detail wrong");
            }
            Clear();
        },
        error: function () {
            alert("LoginApplicationForm1   fail!");
        }
    });
}


function Clear() {
    $("#txtreg").val("");
    $("#txtemail").val("");
```

```
    $("#btnsubmit").val("SUBMIT");
}
```

LoginAppFormvalidation.js:-

```
function Validation() {
    var Dabba = "";

    Dabba += checkreg();
    Dabba += checkemail();

    if (Dabba != "") {
        alert(Dabba);
        return false;
    }
    else {
        SaveData();
    }
}


function checkreg() {
    var TB = $("#txtreg");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter Registration Number\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical Registration
Number!!\n";
    }
    else {
        return "";
    }
}

function checkemail() {
    var TB = $("#txtemail");
    var Exp = /^([a-zA-Z0-9_.-])+@(([a-zA-Z0-9-])+.)+([a-zA-Z0-
9]{2,4})+$/;
    if (TB.val() == "") {
        return 'Please enter your email\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter valid email!!\n";
    }
```

```
    else {
        return "";
    }
}
```

ShowLoginAppFormController.cs:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using OAFII.Models;

namespace OAFII.Controllers
{
    public class ShowLoginAppFormController : Controller
    {
        OAFIIEntities1 db = new OAFIIEntities1();
        public ActionResult ShowLoginAppForm(string QS)
        {
            ViewBag.data = QS;
            return View();
        }
        public JsonResult GetDataById(int A)
        {
            var data = (from x in db.tbl_ApplicationForm1 where x.id
== A select x).ToList();
            return Json(data, JsonRequestBehavior.AllowGet);
        }
    }
}
```

ShowLoginAppForm.cshtml:-

```
@{
    ViewBag.Title = "ShowLoginAppForm";
}

<script src="~/jquery.min.js"></script>
```

```
<table id="tbl" border="1" style="background-
color:yellow;color:red">
    <tr style="background-color:red;color:white">
        <th>Full Name:</th>
        <th>Father Name:</th>
        <th>Mother Name:</th>
        <th>Parmanent Address:</th>
        <th>Pin Code:</th>
        <th>State:</th>
        <th>City:</th>
        <th>Mobile No.:</th>
        <th>DOB :</th>
        <th>Email :</th>
        <th>Registration No. :</th>
        <th>Application Status :</th>
    </tr>
</table>

<script type="text/javascript">

    $(document).ready(function () {
        BindSingleUser();
    });

    function BindSingleUser() {
        $.ajax({
            url: '../ShowLoginAppForm/GetDataById',
            data: {A: @ViewBag.data},
            success: function (data) {
                $("#tbl").append('<tr> <td>' + data[0].FullName +
'</td> <td>' + data[0].FatherName + '</td> <td>' +
data[0].MotherName + ' </td> <td>' + data[0].PermanentAddress + '
</td> <td>' + data[0].PinCode + ' </td> <td>' + data[0].State +
'</td> <td>' + data[0].City + ' </td> <td>' + data[0].MobileNo +
'</td> <td>' + data[0].DOB + '</td> <td>' + data[0].Email + '</td>
<td>' + data[0].RegistrationNo + '</td> <td>' +
data[0].ApplicationStatus + '</td> < tr >');
            },
            error: function () {
                alert("Data not found by ID!");
            }
        });
    }

</script>
```

ApplicationForm1Controller.cs:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using OAFII.Models;

namespace OAFII.Controllers
{
    public class ApplicationForm1Controller : Controller
    {
        OAFIIEntities1 db = new OAFIIEntities1();
        tbl_ApplicationForm1 obj = new tbl_ApplicationForm1();
        public ActionResult ApplicationForm1()
        {
            return View();
        }
        public void Insert(tbl_ApplicationForm1 obj)
        {
            db.tbl_ApplicationForm1.Add(obj);
            db.SaveChanges();
        }
        public JsonResult DisplayState()
        {
            var data = (from a in db.tbl_State select a).ToList();
            return Json(data, JsonRequestBehavior.AllowGet);
        }
        public JsonResult DisplayCity()
        {
            var data = (from a in db.tbl_City select a).ToList();
            return Json(data, JsonRequestBehavior.AllowGet);
        }
    }
}
```

ApplicationForm1.cshtml:-

```
@{
    ViewBag.Title = "ApplicationForm1";
}
```

```
<script src="~/ApplicationForm1Validation.js"></script>
<script src="~/ApplicationForm1.js"></script>
<br />
<br />
<br />

<table style="background-color:cyan; color:black; width:25.5%; text-
align:center">
    <tr>
        <td>
            <h2>ApplicationForm1 Form</h2>
        </td>
    </tr>
</table>

<table style="background-color:gray; color:yellow">
    <tr>
        <td>Full Name:</td>
        <td><input type="text" id="txtfname" /></td>
    </tr>
    <tr>
        <td>Father Name:</td>
        <td><input type="text" id="txtfaname" /></td>
    </tr>
    <tr>
        <td>Mother Name:</td>
        <td><input type="text" id="txtmname" /></td>
    </tr>
    <tr>
        <td>Parmanent Address:</td>
        <td><input type="text" id="txtaddress" /></td>
    </tr>
    <tr>
        <td>Pin Code:</td>
        <td><input type="text" id="txtpincode" /></td>
    </tr>
    <tr>
        <td>State:</td>
        <td>
            <select id="ddlstate">
                <option value="0">SELECT</option>
            </select>
        </td>
    </tr>
    <tr>
        <td>City:</td>
        <td>
```

```
            <select id="ddlcity">
                <option value="0">SELECT</option>
            </select>
        </td>
    </tr>
    <tr>
        <td>Mobile No.:</td>
        <td><input type="text" id="txtphone" /></td>
    </tr>
    <tr>
        <td>DOB :</td>
        <td>
            <input type="date" id="txtdob" />
        </td>
    </tr>
    <tr>
        <td>Email :</td>
        <td>
            <input type="text" id="txtemail" />
        </td>
    </tr>
    <tr>
        <td>Registration No. :</td>
        <td>
            <input type="text" id="txtreg" />
        </td>
    </tr>
    <tr>
        <td>Application Status :</td>
        <td>
            <input type="text" id="txtapp" />
        </td>
    </tr>
    <tr>
        <td></td>
        <td><input type="button" id="btnsubmit" value="SUBMIT"
onclick="return Validation()" /></td>
    </tr>
</table>

<br />
<br />
<br />
```

ApplicationForm1.js:-

```
$(document).ready(function () {
    BindState();
    BindCity();
});

function SaveData() {
    $.ajax({
        url: '../ApplicationForm1/Insert',
        data: { FullName: $("#txtfname").val(), FatherName:
$("#txtfaname").val(), MotherName: $("#txtmname").val(),
PermanentAddress: $("#txtaddress").val(), PinCode:
$("#txtpincode").val(), State: $("#ddlstate").val(), City:
$("#ddlcity").val(), MobileNo: $("#txtphone").val(), DOB:
$("#txtdob").val(), Email: $("#txtemail").val(), RegistrationNo:
$("#txtreg").val(), ApplicationStatus: $("#txtapp").val() },
        success: function () {
            alert("ApplicationForm1 Data successfull insert!");
            Clear();
        },
        error: function () {
            alert("ApplicationForm1 Data not insert");
        }
    });
}

function BindState() {
    $.ajax({
        url: '../ApplicationForm1/DisplayState',
        type: 'post',
        data: {},
        async: false,
        success: function (data) {
            for (var i = 0; i < data.length; i++) {
                $("#ddlstate").append($('<option/>').attr("value",
data[i].sid).text(data[i].sname));
            }
        },
        error: function () {
            alert("State not found!");
        }
    });
}

function BindCity() {
    $.ajax({
        url: '../ApplicationForm1/DisplayCity',
        type: 'post',
        data: {},
```

```
        async: false,
        success: function (data) {
            for (var i = 0; i < data.length; i++) {
                $("#ddlcity").append($('<option/>').attr("value",
data[i].cid).text(data[i].cname));
            }
        },
        error: function () {
            alert("City not found!");
        }
    });
}

function Clear() {
    $("#txtfname").val("");
    $("#txtfaname").val("");
    $("#txtmname").val("");
    $("#txtaddress").val("");
    $("#txtpincode").val("");
    $("#ddlstate").val("0");
    $("#ddlcity").val("0");
    $("#txtphone").val("");
    $("#txtdob").val("");
    $("#txtemail").val("");
    $("#txtreg").val("");
    $("#txtapp").val("");
    $("#btnsubmit").val("SUBMIT");
}
```

ApplicationForm1Validation.js

```
function Validation() {
    var Dabba = "";

    Dabba += checkfuname();
    Dabba += checkfaname();
    Dabba += checkmaname();
    Dabba += checkaddress();
    Dabba += checkpin();
    Dabba += checkstate();
    Dabba += checkcity();
    Dabba += checkmobile();
    Dabba += checkdob();
    Dabba += checkemail();
    Dabba += checkreg();
    Dabba += checkapp();
```

```
    if (Dabba != "") {
        alert(Dabba);
        return false;
    }
    else {
        SaveData();
    }
}


function checkfuname() {
    var TB = $("#txtfname");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  full name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical full name!!\n";
    }
    else {
        return "";
    }
}

function checkfaname() {
    var TB = $("#txtfaname");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  father name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical father name!!\n";
    }
    else {
        return "";
    }
}

function checkmaname() {
    var TB = $("#txtmname");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  mother name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical mother name!!\n";
    }
```

```
    else {
        return "";
    }
}

function checkaddress() {
    var TB = $("#txtaddress");
    if (TB.val() == "") {
        return 'Please enter  address\n'
    }
    else {
        return "";
    }
}

function checkpin() {
    var TB = $("#txtpincode");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter your pincode\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical pincode no!!\n";
    }
    else {
        return "";
    }
}

function checkstate() {
    var TB = $("#ddlstate");
    if (TB.val() == "0") {
        return 'Please select  state\n'
    }
    else {
        return "";
    }
}

function checkcity() {
    var TB = $("#ddlcity");
    if (TB.val() == "0") {
        return 'Please select  city\n'
    }
    else {
        return "";
    }
}
```

```
function checkmobile() {
    var TB = $("#txtphone");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter  mobile number\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical mobile no!!\n";
    }
    else {
        return "";
    }
}

function checkdob() {
    var TB = $("#txtdob");
    if (TB.val() == "") {
        return 'Please select  DOB\n'
    }
    else {
        return "";
    }
}

function checkemail() {
    var TB = $("#txtemail");
    var Exp = /^([a-zA-Z0-9_.-])+@(([a-zA-Z0-9-])+.)+([a-zA-Z0-
9]{2,4})+$/;
    if (TB.val() == "") {
        return 'Please enter email\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter valid  email!!\n";
    }
    else {
        return "";
    }
}

function checkreg() {
    var TB = $("#txtreg");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter Registration Number\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical registration no!!\n";
```

```
    }
    else {
        return "";
    }
}

function checkapp() {
    var TB = $("#txtapp");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter application status name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical application
status!!\n";
    }
    else {
        return "";
    }
}
```

ApplicationForm2Controller.cs:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using OAFII.Models;

namespace OAFII.Controllers
{
    public class ApplicationForm2Controller : Controller
    {
        OAFIIEntities1 db = new OAFIIEntities1();
        tbl_ApplicationForm2 obj = new tbl_ApplicationForm2();
        public ActionResult ApplicationForm2()
        {
            return View();
        }
        public void Insert(tbl_ApplicationForm2 obj)
        {
            db.tbl_ApplicationForm2.Add(obj);
            db.SaveChanges();
        }
```

```
    }
}
```

ApplicationForm2.cshtml:-

```
@{
    ViewBag.Title = "ApplicationForm2";
}

<script src="~/Application Form2.js"></script>
<script src="~/ApplicationForm2Validation.js"></script>

<br />
<br />
<br />

<table style="background-color:cyan; color:black; width:25.9%; text-
align:center">
    <tr>
        <td>
            <h2>ApplicationForm2 Form</h2>
        </td>
    </tr>
</table>

<table style="background-color:gray; color:yellow">
    <tr>
        <td>Qualify Exam:</td>
        <td><input type="text" id="txtq" /></td>
    </tr>
    <tr>
        <td>Name Of University:</td>
        <td><input type="text" id="txtname" /></td>
    </tr>
    <tr>
        <td>Year Of Passing:</td>
        <td><input type="text" id="txtpass" /></td>
    </tr>
    <tr>
        <td>DD Number:</td>
        <td><input type="text" id="txtnum" /></td>
    </tr>
    <tr>
        <td>DD Date :</td>
        <td>
```

```
            <input type="date" id="txtdob" />
        </td>
    </tr>
    <tr>
        <td>Per_Marks:</td>
        <td><input type="text" id="txtper" /></td>
    </tr>
    <tr>
        <td>Amount.:</td>
        <td><input type="text" id="txtamt" /></td>
    </tr>
    <tr>
        <td>Center :</td>
        <td>
            <input type="text" id="txtcen" />
        </td>
    </tr>
    <tr>
        <td>Course. :</td>
        <td>
            <input type="text" id="txtcou" />
        </td>
    </tr>
    <tr>
        <td></td>
        <td><input type="button" id="btnsubmit" value="SUBMIT"
onclick="return Validation()" /></td>
    </tr>
</table>

<br />
<br />
<br />
```

Application Form2.js:-

```
function SaveData() {
    $.ajax({
        url: '../ApplicationForm2/Insert',
        data: { Qualifyingexam: $("#txtq").val(), NameOfUniversity:
$("#txtname").val(), YearOfPassing: $("#txtpass").val(), DDNumber:
$("#txtnum").val(), DDDate: $("#txtdob").val(), Per_Marks:
$("#txtper").val(), Amount: $("#txtamt").val(), Center:
$("#txtcen").val(), Course: $("#txtcou").val() },
        success: function () {
            alert("ApplicationForm2 Data successfull insert!");
```

```
            Clear();
        },
        error: function () {
            alert("ApplicationForm2 Data not insert");
        }
    });
}


function Clear() {
    $("#txtq").val("");
    $("#txtname").val("");
    $("#txtpass").val("");
    $("#txtnum").val("");
    $("#txtdob").val("");
    $("#txtper").val("");
    $("#txtamt").val("");
    $("#txtcen").val("");
    $("#txtcou").val("");
    $("#btnsubmit").val("SUBMIT");
}
```

ApplicationForm2Validation.js:-

```
function Validation() {
    var Dabba = "";

    Dabba += checkquy();
    Dabba += checkuni();
    Dabba += checkyear();
    Dabba += checkddn();
    Dabba += checkddd();
    Dabba += checkpm();
    Dabba += checkamt();
    Dabba += checkcenter();
    Dabba += checkcourse();

    if (Dabba != "") {
        alert(Dabba);
        return false;
    }
    else {
        SaveData();
    }
}
```

```
function checkquy() {
    var TB = $("#txtq");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  Qualifying\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical Qualifying name!!\n";
    }
    else {
        return "";
    }
}

function checkuni() {
    var TB = $("#txtname");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  University name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical University name!!\n";
    }
    else {
        return "";
    }
}

function checkyear() {
    var TB = $("#txtpass");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return 'Please enter  Year Of Passing\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical Year Of Passing!!\n";
    }
    else {
        return "";
    }
}

function checkddn() {
    var TB = $("#txtnum");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter your DD Number\n";
```

```
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical DD Number!!\n";
    }
    else {
        return "";
    }
}

function checkddd() {
    var TB = $("#txtdob");
    if (TB.val() == "") {
        return 'Please select  DD Date\n'
    }
    else {
        return "";
    }
}

function checkpm() {
    var TB = $("#txtper");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter Percentage Marks\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical Percentage Marks!!\n";
    }
    else {
        return "";
    }
}

function checkamt() {
    var TB = $("#txtamt");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter  Amount\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical Amount!!\n";
    }
    else {
        return "";
    }
}

function checkcenter() {
```

```
    var TB = $("#txtcen");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter Center name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical Center name!!\n";
    }
    else {
        return "";
    }
}

function checkcourse() {
    var TB = $("#txtcou");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter Course name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical Course name!!\n";
    }
    else {
        return "";
    }
}
```

ChoiceController.cs:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using OAFII.Models;

namespace OAFII.Controllers
{
    public class ChoiceController : Controller
    {
        OAFIIEntities1 db = new OAFIIEntities1();
        tbl_Choice obj = new tbl_Choice();
        public ActionResult Choice()
        {
            return View();
        }
        public void Insert(tbl_Choice obj)
```

```
        {
            db.tbl_Choice.Add(obj);
            db.SaveChanges();
        }
    }
}
```

Choice.cshtml:-

```
@{
    ViewBag.Title = "Choice";
}
<script src="~/Choice.js"></script>
<script src="~/ChoiceValidation.js"></script>

<br />
<br />
<br />

<table style="background-color:cyan; color:black; width:24%; text-
align:center">
    <tr>
        <td>
            <h2>Choice</h2>
        </td>
    </tr>
</table>

<table style="background-color:gray; color:yellow">
    <tr>
        <td>Registration No.:</td>
        <td><input type="text" id="txtreg" /></td>
    </tr>
    <tr>
        <td>Collage Name1:</td>
        <td><input type="text" id="txtclg1" /></td>
    </tr>
    <tr>
        <td>Collage Name2:</td>
        <td><input type="text" id="txtclg2" /></td>
    </tr>
    <tr>
        <td>Collage Name3:</td>
        <td><input type="text" id="txtclg3" /></td>
    </tr>
```

```
    <tr>
        <td>Collage Name4:</td>
        <td><input type="text" id="txtclg4" /></td>
    </tr>
    <tr>
        <td>Collage Name5.:</td>
        <td><input type="text" id="txtclg5" /></td>
    </tr>
    <tr>
        <td></td>
        <td><input type="button" id="btnsubmit" value="SUBMIT"
onclick="return Validation()" /></td>
    </tr>
</table>

<br />
<br />
```

Choice.js:-

```
function SaveData() {
    $.ajax({
        url: '../Choice/Insert',
        data: { RegistrationNo: $("#txtreg").val(), Collage_name1:
$("#txtclg1").val(), Collage_name2: $("#txtclg2").val(),
Collage_name3: $("#txtclg3").val(), Collage_name4:
$("#txtclg4").val(), Collage_name5: $("#txtclg5").val() },
        success: function () {
            alert("Choice Data successfull insert!");
            Clear();
        },
        error: function () {
            alert("Choice Data not insert");
        }
    });
}

function Clear() {
    $("#txtreg").val("");
    $("#txtclg1").val("");
    $("#txtclg2").val("");
    $("#txtclg3").val("");
    $("#txtclg4").val("");
    $("#txtclg5").val("");
    $("#btnsubmit").val("SUBMIT");
}
```

ChoiceValidation.js:-

```
function Validation() {
    var Dabba = "";

    Dabba += checkreg();
    Dabba += checkclg1();
    Dabba += checkclg2();
    Dabba += checkclg3();
    Dabba += checkclg4();
    Dabba += checkclg5();

    if (Dabba != "") {
        alert(Dabba);
        return false;
    }
    else {
        SaveData();
    }
}


function checkreg() {
    var TB = $("#txtreg");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter Registration Number\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical Registration
Number!!\n";
    }
    else {
        return "";
    }
}

function checkclg1() {
    var TB = $("#txtclg1");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  Collage Name1\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical Collage Name1!!\n";
    }
```

```
    else {
        return "";
    }
}

function checkclg2() {
    var TB = $("#txtclg2");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  Collage Name2\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical Collage Name2!!\n";
    }
    else {
        return "";
    }
}

function checkclg3() {
    var TB = $("#txtclg3");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  Collage Name3\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical Collage Name3!!\n";
    }
    else {
        return "";
    }
}

function checkclg4() {
    var TB = $("#txtclg4");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  Collage Name4\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical Collage Name4!!\n";
    }
    else {
        return "";
    }
}

function checkclg5() {
```

```
    var TB = $("#txtclg5");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  Collage Name5\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical Collage Name5!!\n";
    }
    else {
        return "";
    }
}
```

CollageController.cs:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace OAFII.Controllers
{
    public class CollageController : Controller
    {
        public ActionResult Collage()
        {
            return View();
        }
    }
}
```

Collage.cshtml:-

```
@{
    ViewBag.Title = "Collage";
}

<br />
<br />
<br />

<table style="background-color:blue; color:white" width="58.4%">
    <tr>
```

```
<th></th>
<th></th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th> </th>
<th>
    Collage Name
</th>
<th></th>
<th></th>
<th> </th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th> </th>
<th></th>
<th></th>
<th> </th>
<th> </th>
<th></th>
<th></th>
```

```
        <th> </th>
        <th> </th>
        <th></th>
        <th></th>
        <th> </th>
        <th> </th>
        <th></th>
        <th> </th>
        <th> </th>
        <th></th>
        <th></th>
        <th></th>
        <th> </th>
        <th> </th>
        <th></th>
        <th></th>
        <th></th>
        <th> </th>
        <th> </th>
        <th></th>
        <th></th>
        <th>
            Branch
        </th>
        <th></th>
        <th> </th>
        <th> </th>
        <th></th>
        <th></th>
        <th>
            Seats
        </th>
    </tr>
</table>

<table border="1" style="background-color:black; color:white">
    <tr>
        <td>
            Indian Institute of Information Technology, Design &
Manufacturing
        </td>
        <td>
            Kanchipuram, Tamil Nadu
        </td>
        <td>
            280
        </td>
    </tr>
```

```
    <tr>
        <td>
            Atal Bihari Vajpayee Indian Institute of Information
Technology & Management
        </td>
        <td>
            Gwalior, Madhya Pradesh
        </td>
        <td>
            196
        </td>
    </tr>
    <tr>
        <td>
            Indian Institute of Information Techonology
        </td>
        <td>
            Kottayam
        </td>
        <td>
            90
        </td>
    </tr>
    <tr>
        <td>
            Indian Institute of Information Technology,
        </td>
        <td>
            Allahabad, Uttar Pradesh
        </td>
        <td>
            297
        </td>
    </tr>
    <tr>
        <td>
            Indian Institute of Information Technology, Kota
        </td>
        <td>
            Rajasthan
        </td>
        <td>
            120
        </td>
    </tr>
    <tr>
        <td>
            Indian Institute of Information Technology Srirangam
```

```
        </td>
        <td>
            Tiruchirapalli
        </td>
        <td>
            60
        </td>
    </tr>
    <tr>
        <td>

            Indian Institute of Information Technology, Design &
Manufacturing, Kurnool
        </td>
        <td>
            Andhra Pradesh
        </td>
        <td>
            120
        </td>
    </tr>
    <tr>
        <td>

            Indian Institute of Information Technology Manipur
Mantripukhri
        </td>
        <td>
            Imphal
        </td>
        <td>
            100
        </td>
    </tr>
    <tr>
        <td>
            Indian Institute of Information Technology Kilohrad,
Sonepat
        </td>
        <td>
            Haryana
        </td>
        <td>
            90
        </td>
    </tr>
    <tr>
        <td>
```

```
            Indian Institute of Information Technology Kalyani
        </td>
        <td>
            West Bengal
        </td>
        <td>
            130
        </td>
    </tr>
    <tr>
        <td>
            Indian Institute of Information Technology, Una
        </td>
        <td>
            Himachal Pradesh
        </td>
        <td>
            160
        </td>
    </tr>
    <tr>
        <td>
            Indian Institute of Information Technology, Vadodara
        </td>
        <td>
            Gujarat
        </td>
        <td>
            140
        </td>
    </tr>
    <tr>
        <td>
            Indian Institute of Information Technology Sri City,
Chittoor
        </td>
        <td>
            Andhra Pradesh
        </td>
        <td>
            270
        </td>
    </tr>
</table>

<br />
<br />
<br />
```

FillupFormController.cs:-

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using OAFII.Models;

namespace OAFII.Controllers
{
    public class FillupFormController : Controller
    {
        OAFIIEntities1 db = new OAFIIEntities1();
        tbl_FillupForm obj = new tbl_FillupForm();
        public ActionResult FillupForm()
        {
            return View();
        }
        public void Insert(tbl_FillupForm obj)
        {
            db.tbl_FillupForm.Add(obj);
            db.SaveChanges();
        }
    }
}
```

FillupForm.cshtml

```
@{
    ViewBag.Title = "FillupForm";
}

<script src="~/FillupForm.js"></script>
<script src="~/FillupFormValidation.js"></script>

<br />
<br />
<br />

<table style="background-color:cyan; color:black; width:24%; text-
align:center">
```

```
    <tr>
        <td>
            <h2>Fillup Form</h2>
        </td>
    </tr>
</table>

<table style="background-color:gray; color:yellow">
    <tr>
        <td>Registration No.:</td>
        <td><input type="text" id="txtreg" /></td>
    </tr>
    <tr>
        <td>Name:</td>
        <td><input type="text" id="txtname" /></td>
    </tr>
    <tr>
        <td>Father Name:</td>
        <td><input type="text" id="txtfaname" /></td>
    </tr>
    <tr>
        <td>Stream:</td>
        <td><input type="text" id="txtstr" /></td>
    </tr>
    <tr>
        <td>Total Marks:</td>
        <td><input type="text" id="txtmark" /></td>
    </tr>
    <tr>
        <td>Rank:</td>
        <td><input type="text" id="txtrank" /></td>
    </tr>
    <tr>
        <td></td>
        <td><input type="button" id="btnsubmit" value="SUBMIT"
onclick="return Validation()" /></td>
    </tr>
</table>

<br />
<br />
<br />
```

FillupForm.js:-

```
function SaveData() {
    $.ajax({
        url: '../FillupForm/Insert',
        data: { RegistrationNo: $("#txtreg").val(), Name:
$("#txtname").val(), FatherName: $("#txtfaname").val(), Stream:
$("#txtstr").val(), TotalMarks: $("#txtmark").val(), Ranks:
$("#txtrank").val() },
        success: function () {
            alert("Fillup Data successfull insert!");
            Clear();
        },
        error: function () {
            alert("Fillup Data not insert");
        }
    });
}

function Clear() {
    $("#txtreg").val("");
    $("#txtname").val("");
    $("#txtfaname").val("");
    $("#txtstr").val("");
    $("#txtmark").val("");
    $("#txtrank").val("");
    $("#btnsubmit").val("SUBMIT");
}
```

FillupFormValidation.js:-

```
function Validation() {
    var Dabba = "";

    Dabba += checkreg();
    Dabba += checkname();
    Dabba += checkfaname();
    Dabba += checkstream();
    Dabba += checktotalmarks();
    Dabba += checkranks();


    if (Dabba != "") {
        alert(Dabba);
        return false;
    }
    else {
        SaveData();
```

```
    }
}


function checkreg() {
    var TB = $("#txtreg");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter Registration Number\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical Registration
Number!!\n";
    }
    else {
        return "";
    }
}

function checkname() {
    var TB = $("#txtname");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical name!!\n";
    }
    else {
        return "";
    }
}

function checkfaname() {
    var TB = $("#txtfaname");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter  father name\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical father name!!\n";
    }
    else {
        return "";
    }
}

function checkstream() {
```

```
    var TB = $("#txtstr");
    var Exp = /^[a-z A-Z]+$/
    if (TB.val() == "") {
        return 'Please enter stream\n'
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only alphabetical stream!!\n";
    }
    else {
        return "";
    }
}

function checktotalmarks() {
    var TB = $("#txtmark");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter Total Marks\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical Total Marks!!\n";
    }
    else {
        return "";
    }
}

function checkranks() {
    var TB = $("#txtrank");
    var Exp = /^[0-9]+$/
    if (TB.val() == "") {
        return "Please enter Ranks\n";
    }
    else if (!Exp.test(TB.val())) {
        return "Please Enter Only numerical Ranks!!\n";
    }
    else {
        return "";
    }
}
```

ListOfBranchesController.cs:-

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

```
using System.Web;
using System.Web.Mvc;
using OAFII.Models;

namespace OAFII.Controllers
{
    public class ListOfBranchesController : Controller
    {
        public ActionResult ListOfBranches()
        {
            return View();
        }
    }
}
```

ListOfBranches.cshtml:-

```
@{
    ViewBag.Title = "ListOfBranches";
}

<br />
<br />
<br />

<table style="background-color:blue; color:white" width="20.8%">
    <tr>
        <th>
            S.No.
        </th>
        <th>
            Branch Name
        </th>
        <th>
            Branch Code
        </th>
    </tr>
</table>

<table border="1" style="background-color:black; color:white">
    <tr>
        <td>
            1.
        </td>
        <td>
```

```
        Kanchipuram, Tamil Nadu
    </td>
    <td>
        524
    </td>
</tr>
<tr>
    <td>
        2.
    </td>
    <td>
        Gwalior, Madhya Pradesh
    </td>
    <td>
        474001
    </td>
</tr>
<tr>
    <td>
        3.
    </td>
    <td>
        Kottayam
    </td>
    <td>
        686001
    </td>
</tr>
<tr>
    <td>
        4.
    </td>
    <td>
        Allahabad, Uttar Pradesh
    </td>
    <td>
        211001
    </td>
</tr>
<tr>
    <td>
        5.
    </td>
    <td>
        Rajasthan
    </td>
    <td>
        431
```

```
    </td>
</tr>
<tr>
    <td>
        6.
    </td>
    <td>
        Tiruchirapalli
    </td>
    <td>
        620001
    </td>
</tr>
<tr>
    <td>
        7.
    </td>
    <td>
        Andhra Pradesh
    </td>
    <td>
        931
    </td>
</tr>
<tr>
    <td>
        8.
    </td>
    <td>
        Imphal
    </td>
    <td>
        795001
    </td>
</tr>
<tr>
    <td>
        9.
    </td>
    <td>
        Haryana
    </td>
    <td>
        244413
    </td>
</tr>
<tr>
    <td>
```

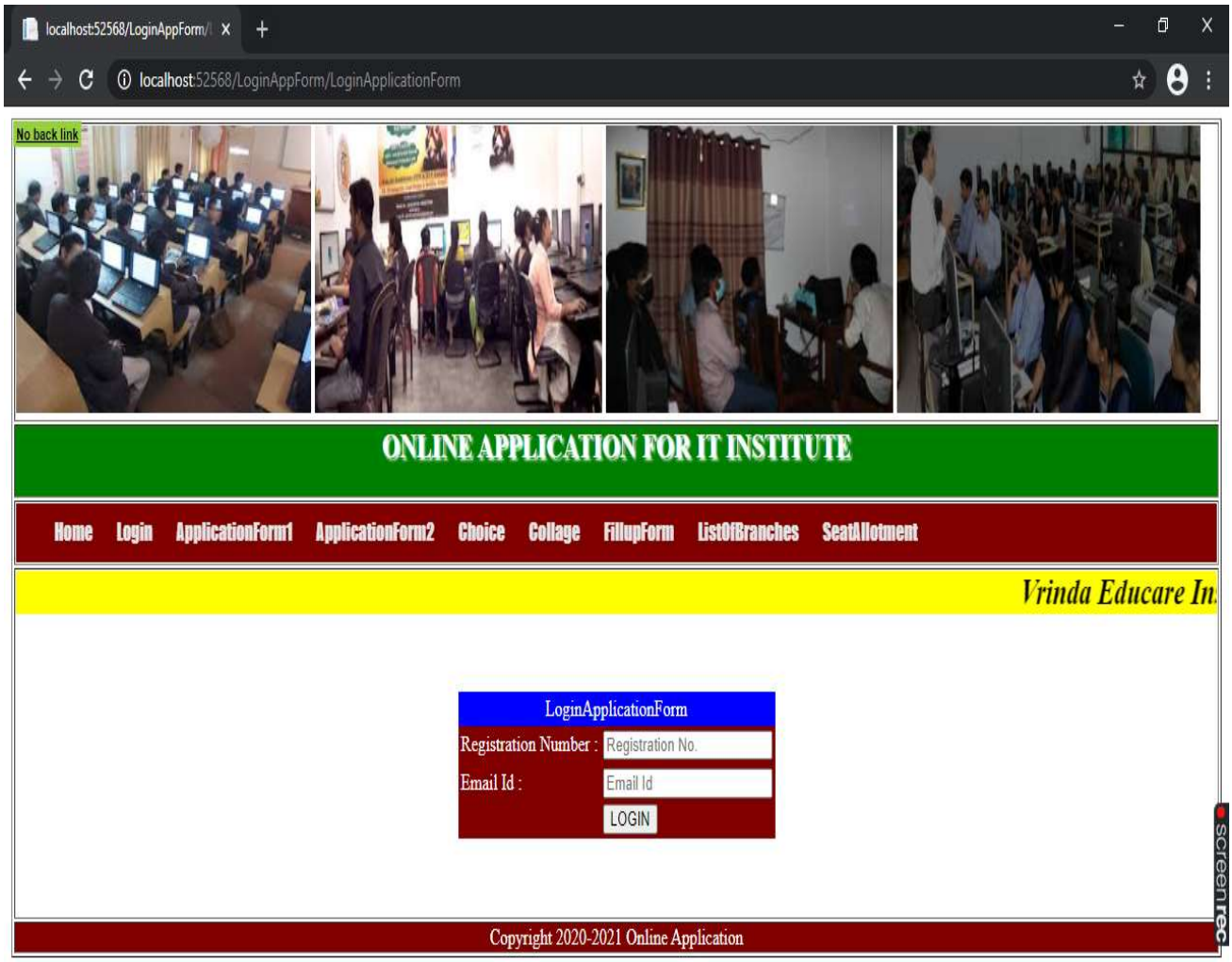
```
                10.
            </td>
            <td>
                West Bengal
            </td>
            <td>
                662
            </td>
        </tr>
        <tr>
            <td>
                11.
            </td>
            <td>
                Himachal Pradesh
            </td>
            <td>
                359
            </td>
        </tr>
        <tr>
            <td>
                12.
            </td>
            <td>
                Gujarat
            </td>
            <td>
                395003
            </td>
        </tr>
        <tr>
            <td>
                13.
            </td>
            <td>
                Andhra Pradesh
            </td>
            <td>
                867
            </td>
        </tr>
    </table>

    <br />
    <br />
    <br /
```

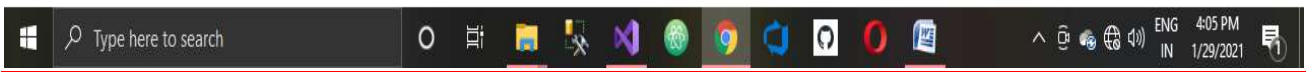
SYSTEM INTERFACE

:Home Page:


localhost:52568/Home/Home
No back link
ONLINE APPLICATION FOR IT INSTITUTE
Home Login ApplicationForm1 ApplicationForm2 Choice Collage FillupForm ListOfBranches SeatAllotment
Vrinda Educare Institute of Career Development
VELS UNIVERSITY
Copyright 2020-2021 Online Application
Type here to search
ENG IN
4:07 PM 1/29/2021

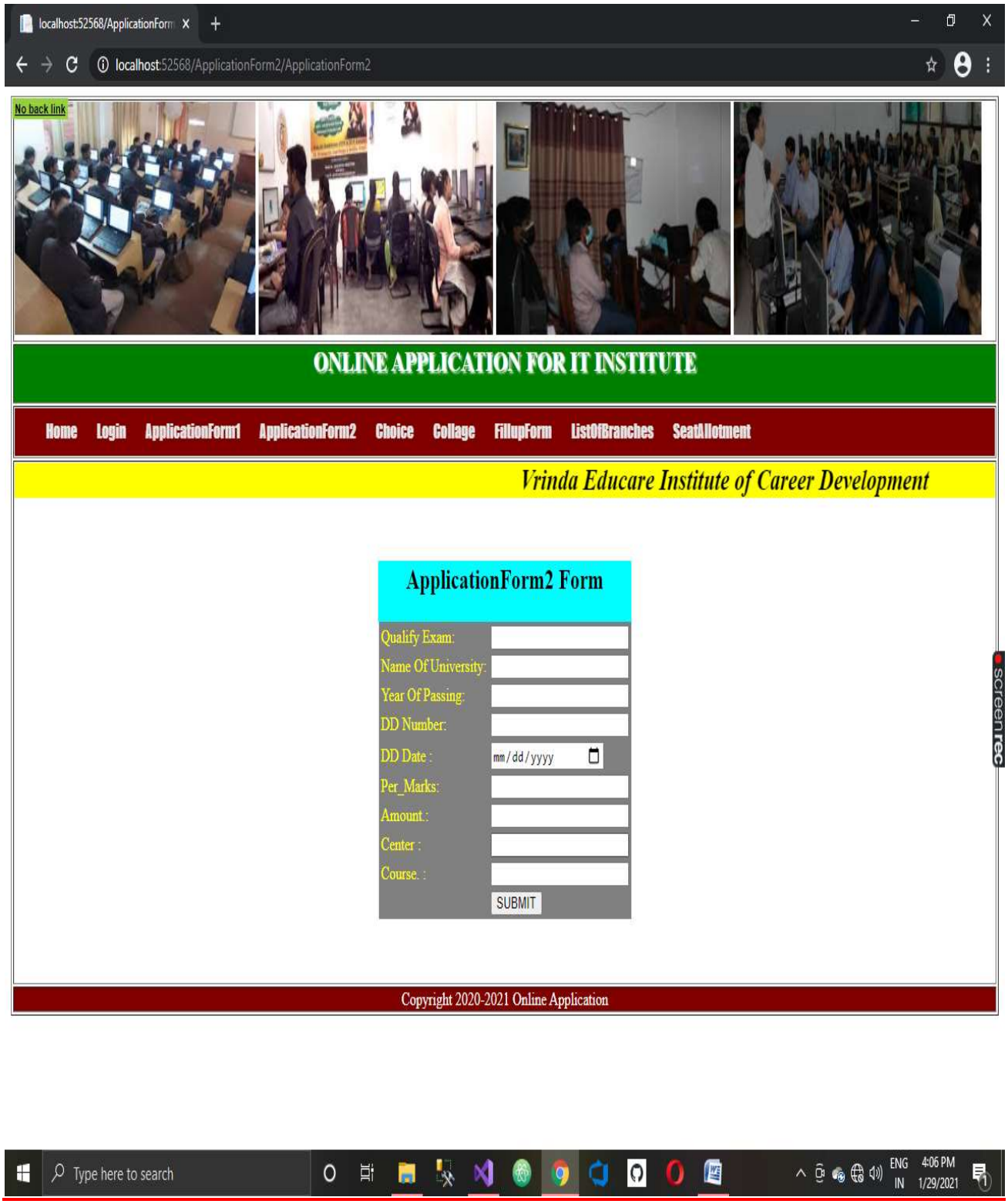
:Login Page:


localhost:52568/LoginAppForm/LoginApplicationForm
No back link
ONLINE APPLICATION FOR IT INSTITUTE
Home
Login
ApplicationForm1
ApplicationForm2
Choice
Collage
FillupForm
ListOfBranches
SeatAllotment
Vrinda Educare In
LoginApplicationForm
Registration Number :
Registration No.
Email Id :
Email Id
LOGIN
Copyright 2020-2021 Online Application

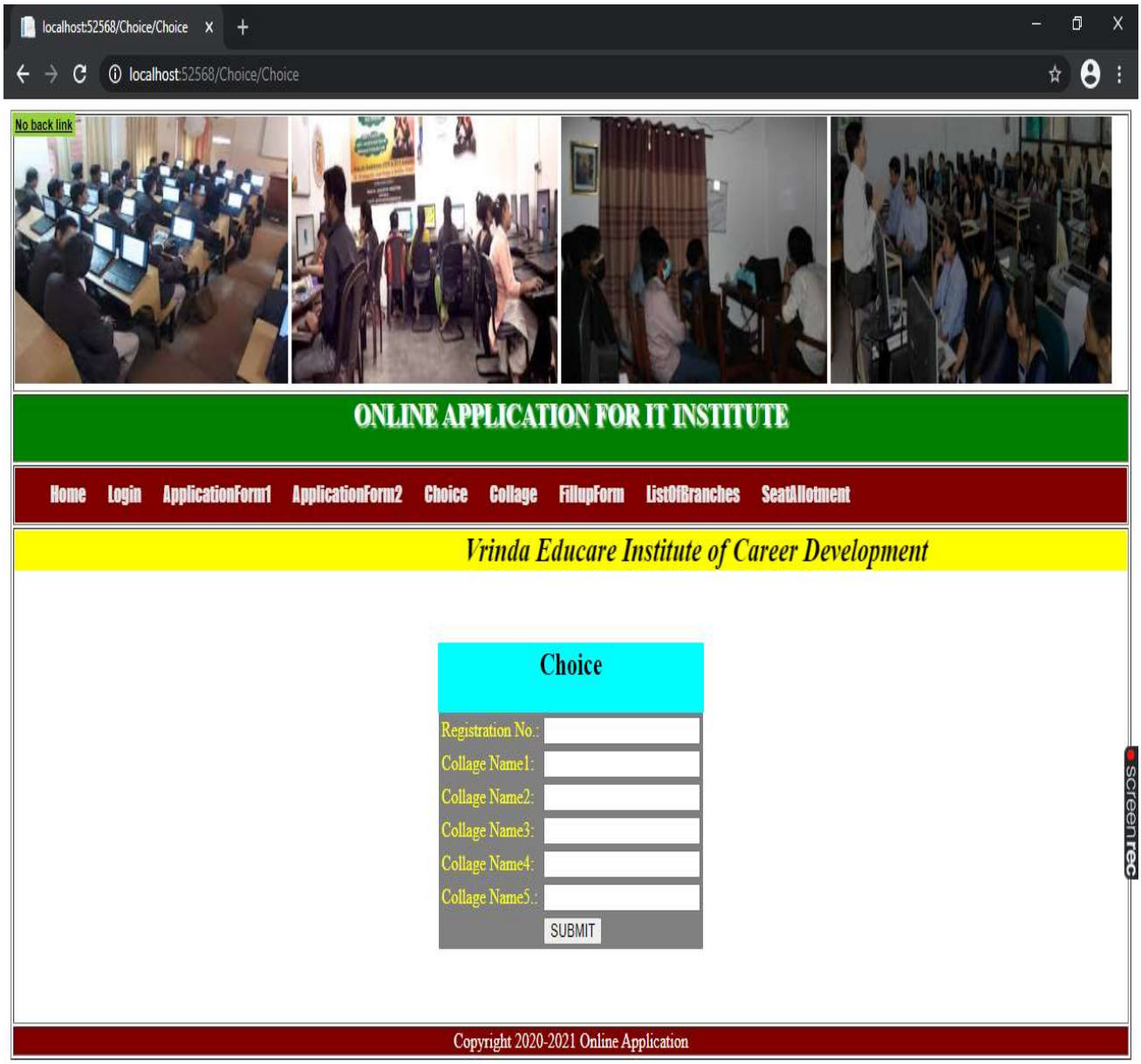

Type here to search
ENG
IN
4:05 PM
1/29/2021

:ApplicationForm1:

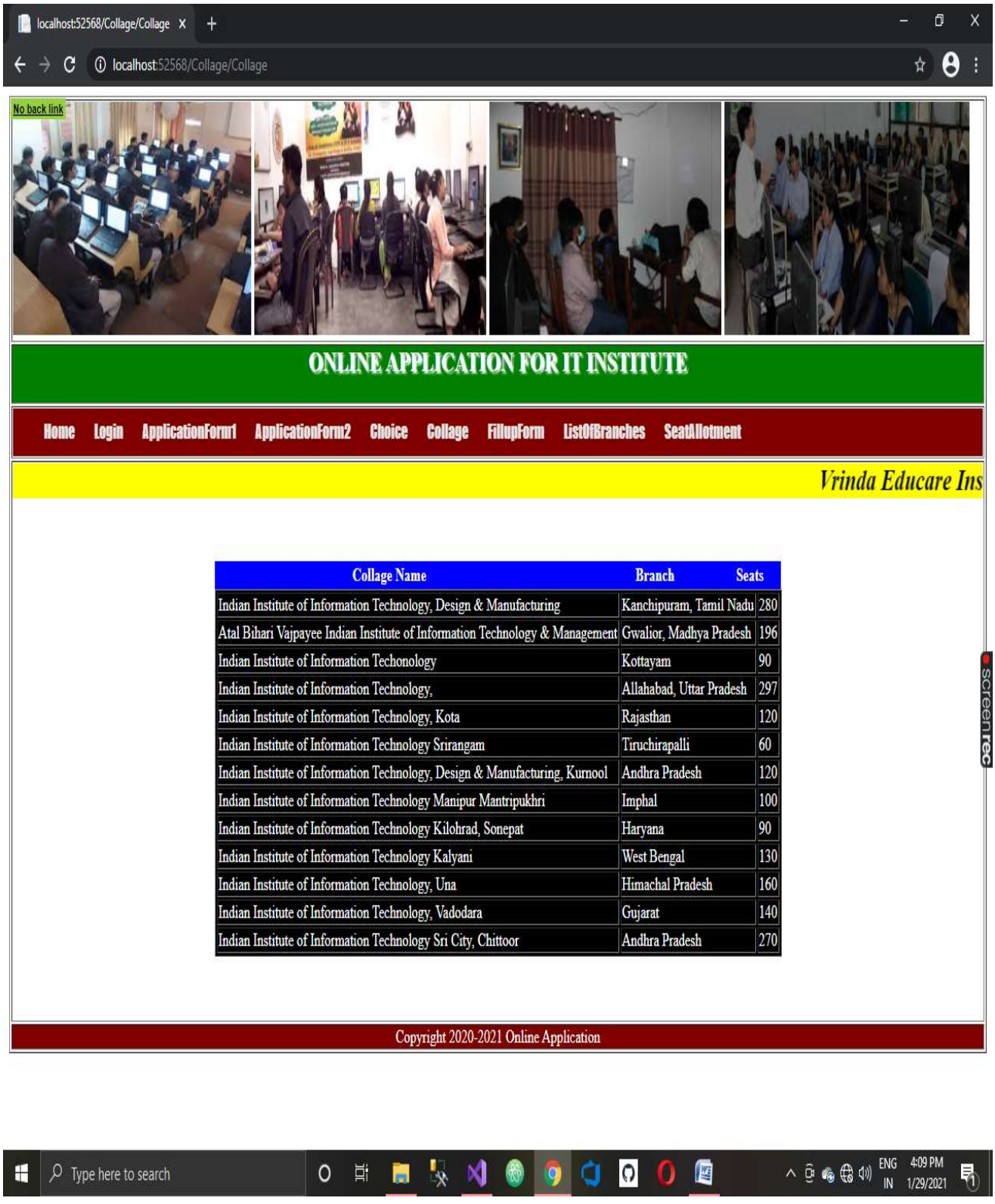
:ApplicationForm2:



:Choice:


localhost:52568/Choice/Choice
localhost:52568/Choice/Choice
No back link
ONLINE APPLICATION FOR IT INSTITUTE
Home
Login
ApplicationForm1
ApplicationForm2
Choice
Collage
FillupForm
ListOfBranches
SeatAllotment
Vrinda Educare Institute of Career Development
Choice
Registration No.:
Collage Name1:
Collage Name2:
Collage Name3:
Collage Name4:
Collage Name5.:
SUBMIT
Copyright 2020-2021 Online Application
Type here to search
ENG
IN
4:07 PM
1/29/2021

:Collage:

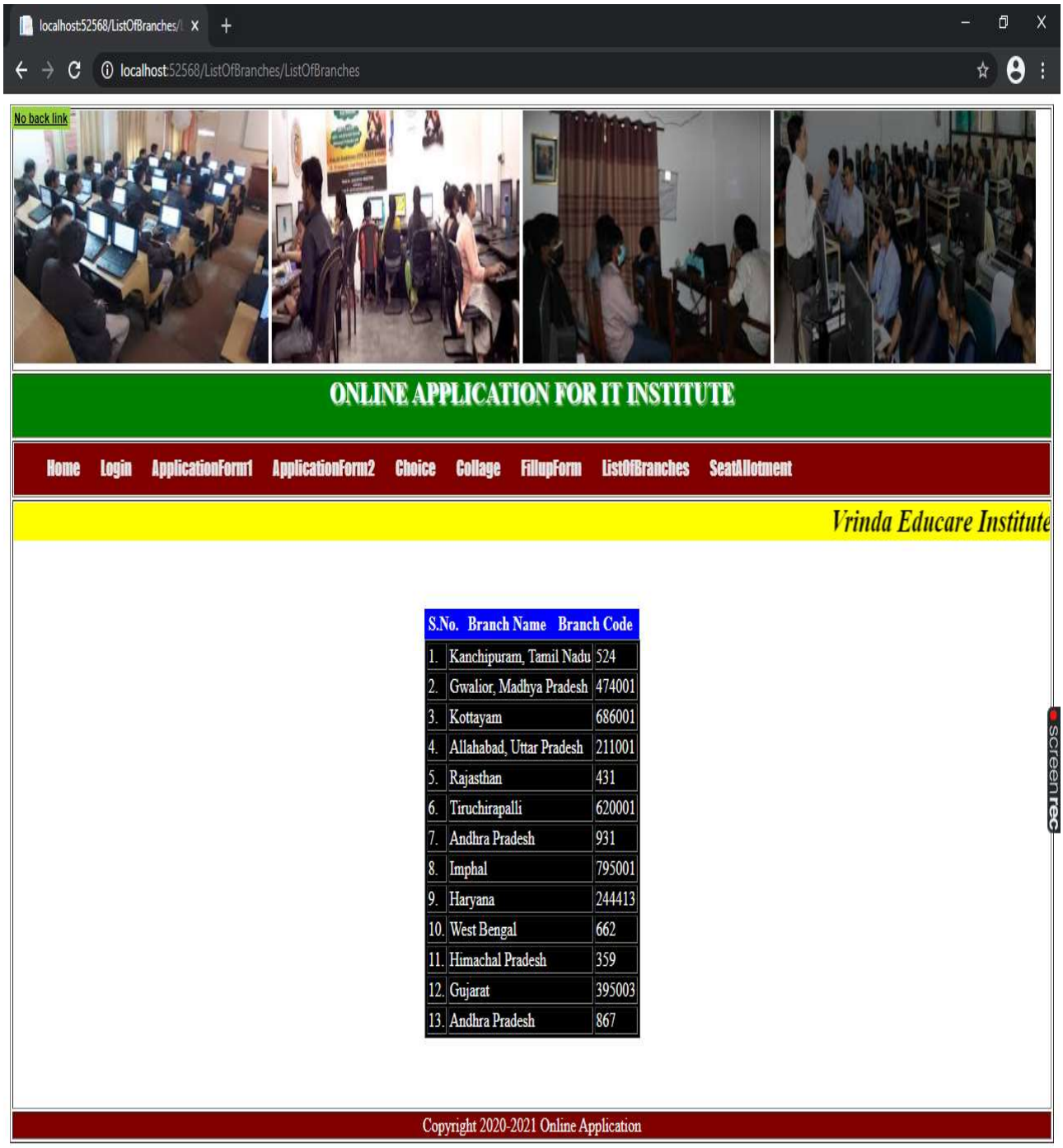


Collage Name	Branch	Seats
Indian Institute of Information Technology, Design & Manufacturing	Kanchipuram, Tamil Nadu	280
Atal Bihari Vajpayee Indian Institute of Information Technology & Management	Gwalior, Madhya Pradesh	196
Indian Institute of Information Techonology	Kottayam	90
Indian Institute of Information Technology,	Allahabad, Uttar Pradesh	297
Indian Institute of Information Technology, Kota	Rajasthan	120
Indian Institute of Information Technology Srirangam	Tiruchirapalli	60
Indian Institute of Information Technology, Design & Manufacturing, Kurnool	Andhra Pradesh	120
Indian Institute of Information Technology Manipur Mantripukhri	Imphal	100
Indian Institute of Information Technology Kilohrad, Sonepat	Haryana	90
Indian Institute of Information Technology Kalyani	West Bengal	130
Indian Institute of Information Technology, Una	Himachal Pradesh	160
Indian Institute of Information Technology, Vadodara	Gujarat	140
Indian Institute of Information Technology Sri City, Chittoor	Andhra Pradesh	270

:FillupForm:


localhost:52568/FillupForm/Fillu
localhost:52568/FillupForm/FillupForm
No back link
ONLINE APPLICATION FOR IT INSTITUTE
Home
Login
ApplicationForm1
ApplicationForm2
Choice
Collage
FillupForm
ListOfBranches
SeatAllotment
Vrinda Educare
Fillup Form
Registration No.:
Name:
Father Name:
Stream:
Total Marks:
Rank:
SUBMIT
Copyright 2020-2021 Online Application
Type here to search
ENG
IN
4:10 PM
1/29/2021

:ListOfBranches:



Code Efficiency

For code efficiency, the following conditions must be fulfilled:

1. **Explicit Strict:-**All variables must declare with proper data type. Option Explicit keyword must be used, it forces each variable or expression in an application **to be defined with a specific data type through the use of the Private, Public, static keywords.** If **Explicit Strict** is not used, undefined variables are automatically defined as **Variant.** Variant data type must be avoided because it takes 16 bytes of memory**.**

2. **Early Binding Vs Late binding:-Early binding is faster than late binding** :-because the application doesn't have to interrogate the object at runtime to determine the objects properties and methods. In Late Binding the objects are defined as objects. Late biding is slower than Early Binding because the application must interrogate the object to determine its properties and methods.

3. **Reclaiming Space:-**Reclaim space form strings by setting a **string variable to the to the empty string ('')** space can also be reclaim from object by setting an **object variable to Nothing**. It frees the space associated with the object if it is the last reference to the object. It also has the advantage of reducing the amount of cleanup work needed when the program terminates.

4. **Control Array: -** Complex forms need more memory. The more controls put on a form, the more memory it will require when it becomes loaded in to memory. Therefore, Reducing the number of controls reduces the memory requirements. For this a control array must be used. A control array counts as only one name. Control arrays consume fewer resources than the equivalent number of independent controls.

5. **Reducing Memory Requirement:-**By using **Dynamic Arrays** we can reduce memory requirements by trimming the array to just the elements we need. Use the **ReDim** statement to change the dimensions of the array. **Erase** statement can be used to free the memory associated with the dynamic array. Use fewer graphics. Graphic images consume memory. The more graphics we have, the more memory we are going to use. Destroy forms when we finished with them. If we are finished with a form we can use the unload statement to remove the form from memory. This will free some memory resources. This will free some memory resources. To free all of the form resources we must set forms object name to nothing.

6. Optimization of Code: - The code of any application must be optimized. An optimized code effect the efficiency of the code and then the application .For application of code, we have used generalized function and that function are used in the entire application .An optimized code is also useful in terms of storage space .It require less hard disk space then the normalize code. It increase the speed of the compiler to compile the program .It needs less memory to run the application .An optimized program is also useful for the compiler when they are writing to give more assistance to their customer.


TESTING

Software Testing Strategy:-

A strategy for software testing may be viewed in the context of the spiral. Unit testing begins at the vortex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progresses by moving outward along the spiral to integration testing, where the focus is on design and the construction of the software architecture. Taking another turns outward on the spiral testing, we encounter validation testing, where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.

Considering the process from a procedural point of view, testing within the context of software engineering is actually a series of four steps that are implemented sequentially. Initially, tests focus on each component individually, ensuring that it functions properly as a unit. That is why it is called unit testing. Unit testing makes heavy use of white-box testing techniques, exercising specific paths in a module's control structure to ensure complete coverage and maximum error detection. Integration testing addresses the issues associated with the dual problems of verification and program construction. Black-box test case design techniques are the most prevalent during integration, although a limited amount of white-box testing may be used to ensure coverage of major control paths. Validation criteria must be tested. Validation testing provides final assurance that software meets all functional, behavioral, and performance requirements. Black-box testing techniques are used exclusively during validation.

Following are the systematic strategy for software testing:-

1. Specify product requirements in quantifiable manner long before testing commences.
2. State testing objectives explicitly.
3. Understand the users of the software and develop a profile for each user category.
4. Develop a testing plan that emphasizes, “Rapid cycle testing”.
5. Build “robust” software that is designed to test itself.
6. Use effective formal technical reviews as a filter prior to test itself.
7. Conduct formal technical reviews to assess the test strategy and test cases themselves.
8. Develop a continuous improvement approach for the testing process.

Software Testing Techniques:-

This is an important stage regarding to product and can overlap with programming stage. This phase consist of testing this project to make sure that the expected output is generated by a given output. It is important to test all kinds of input, not just valid input. Suppose that WE have a form that instruct user to enter the marks, student secure in examination after course completion, is greater than total marks of exam. Although WE obviously want to test values such as 50, 77 or 99, WE should also try inputs like 101 and hello. The importance of software testing and its implementing with respect to software quality cannot be overemphasized. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

Testing Objectives:-

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet undiscovered error.
- A successful test is one that uncovers an as-yet-undiscovered error.

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

Testing Consist Of the Following Steps:-

1) Program Testing
2) String Testing
3) System Testing
4) System Documentation
5) User Acceptance Testing

Various Levels of Testing:--Before implementation the system is tested at two levels, Level I and Level II.

A) LEVEL I TESTING (VRINDA EDUCARETESTING):-

At this level a test data is prepared for testing. Project leaders test the system on this test data keeping the following points into consideration

- ✓ Proper Error handling
- ✓ Exit Points in code

- ✓ Exception handling
- ✓ Input / Output format
- ✓ Glass Box testing
- ✓ Black Box testing

If the system is through with testing phase at LEVEL I then it is passed on to LEVEL II.

(B) LEVEL II TESTING (BETA TESTING):-

Here the testing is done on the live database. If errors are detected then it is sent back to LEVEL I for modification otherwise it is passed on to LEVEL III

This is the level at which the system actually becomes live and implemented for the use of END USERS.

(C) LEVEL III:-

The philosophy behind testing is to find errors. Test cases are devised with the sole purpose in mind. A "Test case" is a set of data that a system will process as normal input. However, the data was created with excess intent of determining whether the system will process them correctly. Thus testing is a process of executing programs with the explicit intention of finding errors, and careful testing makes the system error free. Following were the steps that were/are implemented for testing.

A) Unit testing:-

In unit testing the program are tested which make up the system. The software units in a system are modules and routines that are assembled and integrated to perform a specific

function. Unit testing keep focus on modules and routines, independent of one another to locate errors.

b) System testing:-

System testing does not test the software per routine or module but rather the integration of each module, in the system. It also tests the differences between the system & its original objective, current specifications and system documentation. The primary job thus is to check compatibility of individual modules.

c) Storage Testing:-

It determines the capacity of the system to store transaction data on a disk or in other files. For example, verify documentation statements that the system will store 10,000 records of 400 bytes length on a single flexible disk.

d) Recovery Testing:-

This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.

e) Procedure Testing:-

It determines the clarity of documentation on operation and use of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.

f) White box testing:-

It is called glass-box testing, is a test case design method that uses the control structure of the procedural design to derive test cases. Using white-box testing methods, the software engineer can derive test cases that (1) guarantee that all independent paths within a module have been exercised at least once,

(2) exercise all logical decisions on their true and false sides,

(3) execute all loops at their boundaries and within their operational bounds, and

(4) exercise internal data structures to ensure their validity.

White-box testing of software is predicated on close examination of procedural detail. Logical paths through the software are tested by providing test cases that exercise specific sets of conditions and/or loops. The "status of the program" may be examined at various points to determine if the expected or asserted status corresponds to the actual status. Basis path testing is a white-box testing technique first proposed by Tom McCabe. The basis path method enables the test case designer to derive a logical complexity measure of a procedural design and use this measure as a guide for defining a basis set of execution paths. Test cases derived to exercise the basis set are guaranteed to execute every statement in the program at least one time during testing.

g) Black-box testing:-

Also called behavioral testing, focuses on the functional requirements of the software. That is, black-box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black-box testing is not an alternative to white-box techniques. Rather, it is a complementary approach that is likely to uncover a different class of error than white-box methods. When computer software is considered, black-box testing alludes to tests that are conducted at the software interface. Although they are designed to uncover errors, black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced and that the integrity of external information is maintained. A black-box test examines some fundamental aspect of a system with a little regard for the internal logical structure of the software. Black-box testing attempts to find errors in the following categories: (1) incorrect or missing functions, (2) interface errors, (3) errors in data structures or external database access, (4) behavior or performance errors, and (5) initialization and termination errors. By applying black-box techniques, we derive a set of test cases that satisfy the following criteria: (1) test cases that reduce, by a count that is greater than one, the

number of additional test cases that must be designed to achieve reasonable testing and (2) test cases that tell us something about the presence or absence of classes of errors, rather than an error associated only with the specific test at hand. White-box testing should not, however, be dismissed as impractical. A limited number of important logical paths can be selected and exercised. Important data structures can be probed for validity. The attributes of both black and white box testing can be combined to provide an approach that validates the software interface and selectively ensures that the internal workings of the software are correct.

TEST CASE:-

Test Case1:-

Login Form

Valid User name and Password

Input=Login Id, Password

Output: Corresponding Form Show

Implementation of Test Case:

User Enters valid User name and Password (administrator, User).Login Values are

Mapped in database table to check login id and password is ok. If entry exit then Login

Process will be successful.

Test Case 2:-

Wrong user Id and Password:

Input: Wrong Login Id and Password:

Output: Error Message generated

Implementation of Test Case:

User enters wrong user name and Password. Error message is shown only for 3 times after that it will exit from the current form by showing an aborting message.

Test Case 3:-

From

Output: Details of the Student

Implementation of Test case:

Viewing the details of the Student as well as the message of the Student

Test case 4:-

Reply

Input: Enters the Message to be sent

Output: Conformation of admission

Implementation of Test Case:

User enters the Message to the student.

Implementation of Test Case:

Test Case 5:-

Express Interest

Input: Selecting or Typing the Message

Output: The message is send to the Candidate

Implementation of the Test Case:

Sending a selected Message.

Test case 7:-

Ignore

Input: Entering Comments

Output: Conformation

Implementation of Test case:

Comment to be send to the Candidate

Test case 8:-

View profile

Output: Viewing the Candidate Profile

Implementation of the Test Case:

Viewing the Profiles of the Candidate.

Implementation of Test Case:

User Enters the details depends upon that member Id will generated automatically

Test case 9:-

Free Scanning

Input: Enter the details

Output: Free Scanning

]

IMPLEMENTATION

The implementation view of software requirements presents the real world manifestation of processing functions and information structure. It presents the structural and behavioral aspects of the system.

A design description of an object can take one of two forms:

1) **A protocol description:-** that establishes the interface of an object by defining each message that the object can receive and the related operation that the object performs when it receives the information or

2) **An implementation description: -** that shows implementation details for each operation implied by a message that is passed to an object. Implementation details include information about the object's private part that is internal details about the data structures that describe the object, attributes and the procedural details that describe operation. An implementation description of an object is composed of following information:

1. **A specification of the object's name and reference to class**
2. **A specification of private data structure with indication of data item and types.**
3. **A procedural description of each operation**

The following points are necessary for the implementation

Procurement of necessary hardware and software for installation.

- ✓ Preparation of the user manual for the fulfillment of the user's requirements.
- ✓ To train the end user to how to operate the system.
- ✓ To run the system parallel with the existing system on trial basis for evaluation

Maintenance

An application has served the business needs of an organization for 10 to 15 years. During that time it has been corrected, adapted, and enhanced many times. The application becomes unstable. It still works, but every time a change is attempted. Much of the software we depend on today is on average 10 to 15 years old. Even when these programs were created using the best design in coding techniques known at the time and (most were not), they were created when program size and storage space were principle concern. They were then migrated to new platform adjusted for changes in machine and operating system technologies and enhance to meet new user needs all without enough regard to overall architecture. The maintenance of existing software can account for over 60% of all effort expended by a development organization and the percentage continues to rise as more software is produced. There are four different maintenance activities: -

(A) Corrective Maintenance: - Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective maintenance changes the software correct defects.

(B) Adaptive Maintenance: - Over time, the original environment (e.g. CPU, operating system, business rules, external product characteristics) for which the software was developed is likely to change. Adaptive maintenance results in modification to the software to accommodate the changes to its external environments.

(C) Perfective Maintenance: - As software is used, the customer/user will recognize additional functions that will provide benefit. Perfective maintenance extends the software beyond its original functional requirements.

(D) Preventive Maintenance or Reengineering:-Computer software deteriorates due to change and because of this preventive maintenance often called software reengineering must be conducted to enable the software to serve the needs of its end users. In sense preventive maintenance make changes to computer program easily corrected, adapted and enhance For the project, we have used the C# as front end, and we know that the Executable file created by C# .NET is a true EXE i.e. one can run it in an environment where the C# does not exist. Extra features that is provided by C# is the facility to make packages and deployment of software very easily and we make the software in a particular version. While doing the maintenance work we can make another version with the more features including the existing features. After completion of maintains work, one can run a batch program that is available in package. It will make a new exe file which replace the older one. This makes us easy to make changes to the software while having the database consistent.

Security Mechanisms

Only the registered organization will be able to use the application. The system provides the facility to take backup and restoring of the data. There is no redundancy in the system at any stage. The structure of each table has designed in such a way that to remove any kind of encryption or decryption firewalls in the processing of the data. If a transaction completes, then the record is saved in the concerned tables otherwise not.

There is a lot error handling technique will be used in the system. There is no deadlock situation comes at any stage in the system. Because each and every user will have its own permissible rights to add/update/delete operation in the tables In the **System we** have taken mainly the three security preservers through software coding, to protect the data and system by unauthorized users:

1) User of the system will have to go through the login screen and must enter the User Id and Password to enter to the Main Menu. If anyone who doesn't know the password and gives wrong user id and password then system will inform about wrong information entered. If he/she cancels this login prompt then application will unload itself.

The User Name and Password provided to the Administrator in this project is:

User Id	:	as
Password	:	as

2) Most important is the use of SQL Server system. It has its own password. Only valid user can enter in SQL Server database that knows user id and password. We also create trigger in SQL Server for security. It get fired when any updating, modification, deletion will be done on any table. It keeps record of date, time, user id, name of table, old value, and altered value on which above operation is done, and thus we can keep record of any discrepancy done.

3) The use of Windows-XP will not provide access the system to unauthorized users.

The above security will stop the unauthorized user to use the system. We make two provisions in case of hardware failure. These are:

(a) For hardware failure especially in the case of hard disk failure, SQL Server has the Option, of MIRROR hard disk, which back up the record in another hard disk.

(b) In the case of server failure Windows XP has the facility to run another CPU, which are, connected parallel to server and back up all the data of server and invoked in the case of server Security is the protection of the database against unauthorized access or change. Because you separate the data from the application, and because you allow multiple near accesses to the data, you must also take security need into account. To maintain & implement the security, risk analysis and management is done. Every user has a user ID & its own password .When user input the user ID & password then match if it is a valid ID & password then open the application but if the ID and password not valid then application is not open. –

VALIDATION CHECKS:-System security refers to various validations on data in form of checks. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employs two types of checks.

Cost Estimation of the Project

Cost in a project is due to the requirements for software, hardware, and human resources. Hardware resources are computer time, terminal time and memory required for the project. Software resources include the tools and compilers needed during development. The bulk of cost of software development is due to human resources needed. Cost estimates are determined in terms of person-months (**PM).**

Software costs constituted a small percentage of the overall computer-based system cost. An order of magnitude error in estimates of software cost had relatively little impact. Today, software is the most expensive elements of virtually all computer-based systems. For complex, custom systems, a large cost estimation error can make the difference between profit and loss.

Software cost and effort estimation will never be an exact science. Too many variables-human, technical, environmental, political-can effect the ultimate cost of software and effort applied to develop it. However, software project estimation can be transformed from a black art to a series of systematic steps that provide estimates with acceptable risk.eve reliable cost and effort estimates, a number of options are:

1. Delay estimation until in the project (obviously, we can achieve 100% accurate after the project is complete.)
2. Base estimates on similar projects that have already been completed.
3. Use relatively simple decomposition techniques to generate project cost and effort estimates.
4. Use one or more empirical models for software cost and effort estimation. Unfortunately, the first option, however attractive, is not practical. Cost estimates must be provided “up front”. However, we should recognize that the longer wait, the more we know, and the more we know, the less likely we are to make serious errors in our estimates.

The second option can work reasonably well; if the current project is quite similar to past efforts and other project influences (e.g. the customer, business conditions, the SEE, deadlines) are equivalent. Unfortunately, past experience has not always been a good indicator of future results.

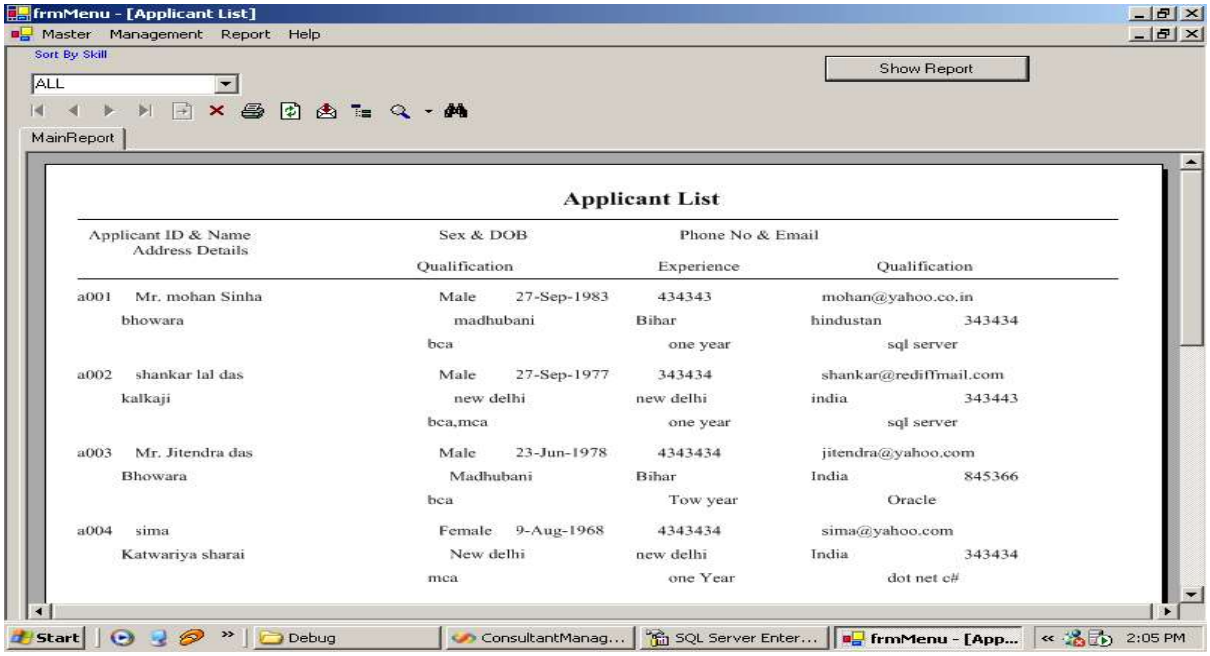
The remaining options are viable approaches to software project estimation. Ideally, the techniques noted for each option should be applied in tandem, each used as a crosscheck for the other. Decomposition techniques take a divide and conquer approach to software project estimation. By decomposing a project into major functions and related software engineering activities, cost and effort estimation can be performed in a stepwise fashion. Empirical estimation models can be used to complement decomposition techniques and offer a potentially valuable estimation approach in their own right. Am model is based on experience and takes the form

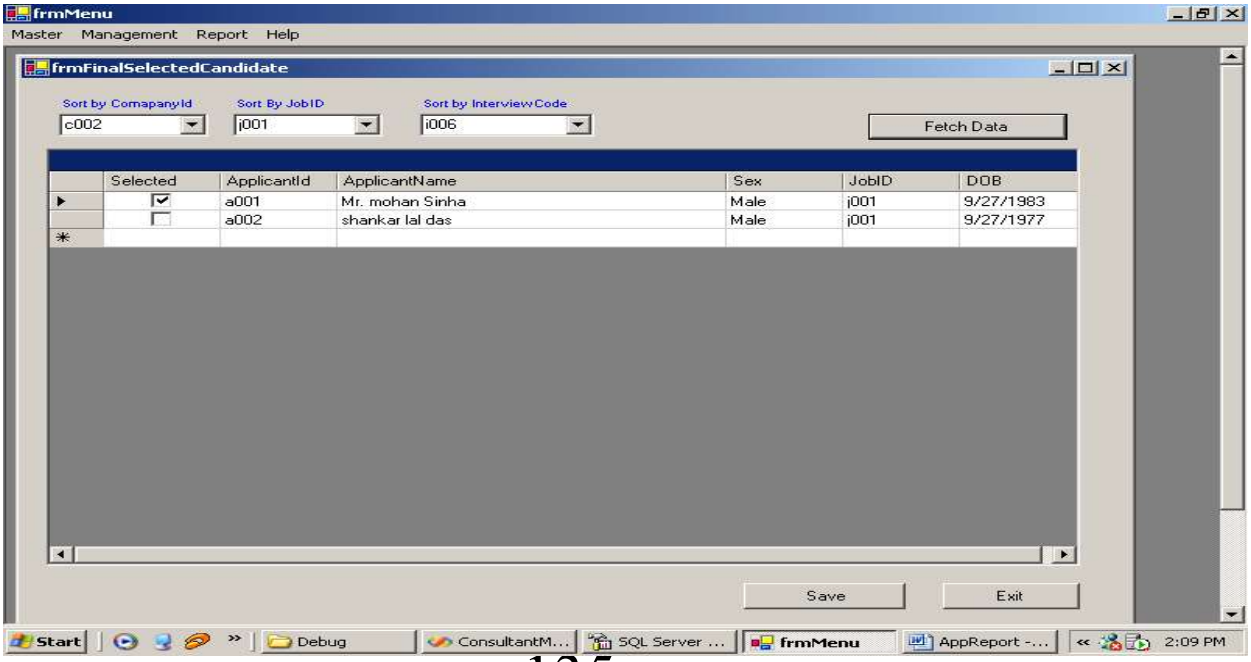
$$D = f(Vi)$$

Where d is one of a number of estimated values (e.g. effort, cost, project duration and Vi are selected independent parameters (e.g. estimated LOC or FP).

Cost Estimate: (Cost of Reference Book + Coaching fee for Software learn + Traveling Cost + Computer resources Used + Man Power + Software cost + Hardware cost)

Report


frmMenu - [Applicant List]
Master Management Report Help
Sort By Skill
ALL
Show Report
MainReport
Applicant List
Applicant ID & Name Address Details
Sex & DOB Qualification
Phone No & Email Experience Qualification
a001 Mr. mohan Sinha bhowara Male 27-Sep-1983 madhubani bca 434343 Bihar one year mohan@yahoo.co.in hindustan 343434 sql server
a002 shankar lal das kalkaji Male 27-Sep-1977 new delhi bca,mca 343434 new delhi one year shankar@rediffmail.com india 343443 sql server
a003 Mr. Jitendra das Bhowara Male 23-Jun-1978 Madhubani bca 4343434 Bihar Tow year jitendra@yahoo.com India 845366 Oracle
a004 sima Katwariya sharai Female 9-Aug-1968 New delhi mca 4343434 new delhi one Year sima@yahoo.com India 343434 dot net c#
Start Debug ConsultantManag... SQL Server Enter... frmMenu - [App... 2:05 PM


frmMenu
Master Management Report Help
frmFinalSelectedCandidate
Sort by ComapanyId
c002
Sort By JobID
j001
Sort by InterviewCode
i006
Fetch Data
Selected ApplicantId ApplicantName Sex JobID DOB
a001 Mr. mohan Sinha Male j001 9/27/1983
a002 shankar lal das Male j001 9/27/1977
Save
Exit
Start Debug ConsultantM... SQL Server ... frmMenu AppReport -... 2:09 PM

Planning and scheduling

Planning of this project will include following things:

Topic understanding:-

- Modular break-up of the system
- Processor logic for each module
- Database requirement

It is vital that the field of application as introduced in the project may be totally a new field.

So as soon as I took this project, I carefully went through the project to identify the

requirements.

Modular break-up of the system:

It consists of following phases:

- Identify the various modules in the system
- List them in the right hierarchy
- Identify their priority of development

PERT Chart, GANTT Chart:-

PERT stands for program evaluation review technique. Unlike bar charts, PERT can be both a cost and a time management system. PERT is organized by events and activities or tasks. PERT has several advantages over bar charts and is likely to be used with more complex projects. One advantage of pert is that it is scheduling device that also shows graphically which task must be completed before others are begun. Also by displaying the various task paths, PERT enables the calculation of critical path. Each path consists of combinations of task, which must be completed. The time and cost associated with each

task along a path are calculated and the path, which requires greatest amount of time, is critical path. Calculation of the critical path enables project managers to monitor this series of tasks more likely than others and to shift resources to it begins to fall behind schedule. Pert controls time and costs during the project and also facilitates finding the right balance between completing it within the budget. The appropriate way to manage a project is to define and control each task. Because project often falls behind schedule, pert is designed to facilitate getting a project back on schedule. Pert is based in part on the premise that subjective estimates of total completion time for a project usually greatly inferior to the sum of subjective estimates for each task. As with Gantt charts, to build a pert chart for a project, one must determine the dependence of the activities required for completion of the project and estimate how long each will take. Then one must determine the dependence of the activities on each other. In facts, the pert chart gives a graphical representation of this information. Clearly this technique does not help in deciding which activities are necessary or how long each will take, but it does force the manager to take the necessary planning steps to answer these questions. Pert chart are available in next page:

Some of the advantages of pert chart

It forces the manager to plan.

- ✓ It shows the interrelationships among the task in the project.
- ✓ It clearly identifies critical path of the project.
- ✓ It exposes all possible parallelism in the activities and thus helps in allocating .
- ✓ It allows scheduling and simulation of alternative schedules.
- ✓ It enables the manager to monitor and control the project.

Gantt Chart

Gantt chart is also known as Bar chart Hennery ***L. Gantt*** has developed Gantt chart. Gantt charts are a project control technique that can be used for several purposes, including scheduling, budgeting and resources planning. A Gantt chart is a bar chart, with each bar representing an activity. The bars are drawn against a time line. The length of each bar is proportional to the length of time planned for the activity. A Gantt charts help in scheduling the activities of a project, but it does not help in identifying them. One can begin with activities identified in work breakdown structure as we did below for quality management project during the scheduling activity, and also during implementation of the project quality management system, new activities may be identified that were not envisioned during the initial planning. The project manager must then go back and revise the breakdown structure and the schedules to deal with news activities. The Gantt chart in the figure is actually an enhanced version of standard Gantt charts. In below figure of Gantt chart, horizontal bars have been shown the duration of actions or tasks. The white bar shows the Slack time that is, latest time by which a task must be finished. Gantt chart, in which, each bar would be represented on of the engineers. Gantt charts are useful for resource, planning and scheduling. When a bar chart is used as a project control method, milestones or checkpoints usually are placed at the completion of each task. They indicate the completion of a particular task and are the basis for determining whether the task and the project are on schedule. Reviewer can ask whether resources allocated have been properly utilized and whether the task has been satisfactorily completed. However, because the bar chart incorporates only the scheduling dimension of a project, it gives little indication of which tasks must be completed before others are begun, and projects cost must be accumulated and evaluated using other method

Phase	Work Task	Apr.	May.	June.	July.	Aug.	Sept.	Oct.	Nov.	Dec.	Jan.
1	Feasibility study Problem Definition System specification S/W Specification										
2	System Design Data Design Input Output Design Functional Design Procedural Design										
3	Database Creation Main Interface & Menu Forms Report										
3	Unit Testing Integration testing Validation Testing System testing										
5	Documentation										

FUTURE SCOPE OF SYSTEM

After sufficient volume of data is stored in the system, the data mining and data warehousing techniques should be employed for information analysis and decision making

- Demand forecasting based on the available data.
- Decision support oriented reports
- Any type of report that not been generated earlier for generating new hypothesis using the available data.
- Loan provisions and recovery

Manpower analysis and deployment. The further scope of the project is very good. With a few of more changes this project can be deployed in the real life environment. This Project will reduce the scope of error, the workload and time; Increase efficiency taken to generate reports. The automated system will also help management to keep its data up to date.

Further Enhancement:-

This project will have bright future in which payment procedure will be performed through credit card this leads to better money transaction.
There may be online career counseling; it will help the applicant to select the better admissions. There will be online interview facility it will save time and money of Student and College.

Advantages of the proposed system:-

The user can access the system from anywhere .If the user wants any query about the clients, applicants or requirement then a few keystrokes on a computer keyboard by a operator can avail all the details helping him/her not to wander department to department to access the

details. The paper work is greatly minimized. No training is given to the users (operators). With the interface a layman can have pleasant experience working with the application.
A detailed analysis of software requirements would provide necessary information for estimates, but analysis often takes weeks or months to complete. Therefore, We must examine the product and the problem it is intended to solve at very beginning of the project, At a minimum, the scope of the product must be established and bounded. Scope is defined by answering the following questions-

Context- How does the software to be built into a larger system, product, or business context and what constraints are imposed as a result of the context?

Information Objectives– What customer–visible data objects are produced as output from the software? What data objects are required for input?

Function and Performance– What function does the software perform to transform input data into output? Is any special performance? In recent times in India and also other parts of the world record handling and maintaining the accuracy have become cumbersome processes. This software will assist the concerned client in maintaining their records and searching proper matching as per requirement only at mouse click. Manually find out the certain records in a very tedious. Cumbersome and risky job. This software can well assist in proper Real Estate Records Management.

Bibliography

system analysis and Design-Elias M. Awad

ASP.Net Bible- Mridula Parihar

Wrox-Professional SQL Server-Louis Davidson

HTML 4.0-E.Stephen Mack and Janan Platt

By Kevin Loney and George Koch

Roger S. Pressman, "Software Engineering (A Practitioner's Approach)",

McGraw-Hill 2000

MCS -032 Object Oriented Analysis and Design

MCS-034 Software Engineering

MCS-043 Advance Database Management System

APPENDIX

VISUAL C# CONTROLS USED IN THIS PROJECT

Text Box:-

A text Box control, sometimes called an edit field or edit control, displays information entered at design time, entered by the user, or assigned to the control in code at run time.

Command Button:-

Use a Command Button control to begin, interrupt, or end a process. When chosen, a Command Button appears pushed in and so is sometimes called push button.

List Box:-

A List Box control displays a list of items from which the user can select one or more. If the number of items exceeds the number that can be displayed, a scroll bar is automatically added to the List Box control.

Label:-

A Label control is a graphical control you can use to display text that a user can't change directly.

Frame:-

A Frame control provides an identifiable grouping for controls. You can also use a Frame to subdivide a form functionally – for example, to separate groups of Option Button controls.

Drop Down Box:-

A Dropdown Box control combines the features of a text box and a list box. This control allows the user to select an item either by typing text into the combo box, or by selecting it form the list.

Timer:-

A Timer control can execute code at regular intervals by causing a Timer event to occur. The Timer control, invisible to the user, is useful for background processing.

Picture Box:-

The primary use for the Picture Box control is a display a picture to the user. The actual picture that is displayed is determined by the picture property. The picture property contains the file name (and optional path) for the picture file that you wish to display.

Data Grid Control:-

The Data Grid control displays and operates on tabular data. It allows complete flexibility to sort, merge, and format table containing strings and pictures. When bound to a Data control, Data Grid displays read-only data.

Line Control:-

A Line control is a graphical control displayed as horizontal, vertical, or diagonal line. You can use Line control at design time to draw lines on forms.

Shape Control:-

The Shape control is a graphical control displayed as a rectangle, square, oval, circle,

Date and Time Picker Control:

A Data and Time Picker (DTP) Control provides a simple and intuitive interface through which to exchange data and time information with a user. For example, with a DTP control you can ask the user to enter a data and then retrieve his or her selection with ease.

Option Button:-

An Option Button control displays an option that can be turned on or off.

Image Control:-

Use the Image control to display a graphic. An Image control can display a graphic form an icon, bitmap or metafile, as well as enhanced metafile, JPEG, or GIF files.

Check Box Control:-

A Check Box indicates whether a particular condition is on or off. We use check boxes in an application to give users true/false or yes/no options. Because check boxes work independently of each other, a user can select any number of check boxes at the same time.

GLOSSARY

VISUAL C#.NET STATEMENTS & FUNCTIONS AND SQL SERVER KEYWORDS USED IN THIS PROJECT

Call Statement:-

Transfers control to a Sub procedure, Function Procedure, or dynamic link library (DLL) Procedure.

InStr Function:-

Returns a Variant (Long) specifying the position of the first occurrence of one string within another.

Lcase Function:-

Returns a string that has been converted to lowercase.

Ucase Function:-

Returns a string that has been converted to uppercase.

Left Function:-

Returns a Variant (String) containing a specified number of characters from the left side of a string.

Len Function:-

Returns a Long containing the number of characters in a string or a number of bytes required to store a variable.

Mid Function:-

Returns a Variant (String) containing a specified number of characters form a string.

OBColor Function:-

Returns a Long representing the RGB color code corresponding to the specified color number.

Rnd Function:-

Returns a Single containing a random number.

StrComp Function:-

Returns a Single containing a random number.

StrComp Function:-

Returns a Variant (Integer) indicating the result of a string comparison.

Time Function:-

Returns a Variant (Data) indicating the current system time.

Now Function:-

Returns a variant (Date) specifying the current date and time according to hour computer's system date and time.

Val function:-

Returns the numbers contained in a string as a numeric value of appropriate type.

Ltrim,Rtrim, and Trim Function:-

Returns a Variant(String) containing a copy of a specified string without leading spaces(Ltrim), trailing spaces(Rtrim), or both leading and trailing spaces(Trim).

With Statement:-

Executes a series of statement on a single object or a user defined type.

MesssageBox Function:-

Displays a message in a dialog box, waits for the user to click button, and returns an integer indicating which button the user clicked.

Int, Fix Functions:-

Returns the integer portion of a number.

Round Function:-

Returns a number rounded to a specified number of decimal places.

Exit Statement:-

Exists a block of DO…Loop, For…Next, Function, Sub, or Property code.

Option Explicit Statement:-

Used at module level to force explicit declaration of all variables in that module.

While….Wend Statement:-

Executes a series of statements as long as a given condition is True.

Do….Loop Statement:-

Repeats a block of statement while a condition is True of until a condition becomes True.

For…Next Statement:-

Repeats a group of statements a specified number of times.

Dim Statement:-

Declares variables and allocates storage space.

Public Statement:-

Used at module level to declare public variables and allocates storage space.

SQL SERVER:-

A database is stored in a very structured manner. Each database requires some way for a user to interact with the information within. Such interaction is performed by a database management system (DBMS). SQL Server is a member of a large category of products known as database management systems (DBMS). The general purpose of a DBMS is to provide for the definition, storage, and management of data in a centralized area that can be shared by many users. SQL Server's database management system is patterned on the relational model.

Relational databases allow us to store vast amounts of data with far sampler maintenance and smaller storage requirements than the equivalent flat database. Relations among tables in a relational database are established using keys. A primary key is a field that uniquely identifies a record so it can be referenced from a related table . A foreign key is a field that holds identification values to relate records stored on other tables.

Querying the database: -

With each query of the database, we form a virtual table that contains the results of our query. Database queries are made with a specific language named SQL (structured query language)