CAREER GUIDANCE SYSTEM

A PROJECT REPORT

Submitted by

Joshi Milin C

Shah Vatsal N

In fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

Computer Engineering

C.U.Shah College of Enggineering &Technology

Wadhwan City- 363030

Gujarat Technological University, Ahmedabad

April, 2013

C. U. SHAH COLLEGE OF ENGINEERING & TECHNOLOGY,

WADHWAN CITY- 363 030.

COMPUTER ENGINEERING

2013

CERTIFICATE

DATE: 25/04/2013

This is to certify that the Industry Defined Problem(IDP) entitled **"Career guidance system"** has been carried out by **Shah Vatsal N**. under my guidance in fulfillment of the degree of Bachelor of Engineering in Computer Engineering (8^{th} Semester) of Gujarat Technological University, Ahmedabad during the academic year 2012-13.

Mr. Vijay.K.Jadeja **Dr. K.H.Wandra**

(Internal Guide) **(Principal, CCET)**

C. U. SHAH COLLEGE OF ENGINEERING & TECHNOLOGY,

WADHWAN CITY- 363 030.

COMPUTER ENGINEERING

2013

CERTIFICATE

DATE: 24/04/2013

This is to certify that the Industry Defined Problem(IDP) entitled **"Career guidance system"** has been carried out by **Joshi Milin C**. under my guidance in fulfillment of the degree of Bachelor of Engineering in Computer Engineering (8^{th} Semester) of Gujarat Technological University, Ahmedabad during the academic year 2012-13.

Mr. Vijay.K.Jadeja **Dr. K.H.Wandra**

(Internal Guide) **(Principal, CCET)**

Acknowledgement

We take upon this opportunity endowed upon us by the grace of the almighty, to thank all those who have been part of this endeavor.

Firstly, we would like to thank our Institute **C. U. Shah College of Engg. & Tech.** and the **Computer Engineering** Department for giving us the opportunity as well as lending our the resources to fulfill our work.

We sincerely thank to **Dr. K. H. Wandra (Principal, CCET)** for giving us the chance as well as the support for all the time being.

We would like to thank our Project guide **Mr. V.K.Jadeja (Assistant Professor CE/IT dept.)** for giving us the right direction to follow and proper guidance regarding the topic. Without his active involvement and the right guidance this would not have been possible.

Last but not the least, we heartily appreciate all those people who have helped us directly or indirectly in making this tasks a success.

Joshi Milin C. (8^{th} CE-2)

Shah Vatsal N. (8^{th} CE-2)

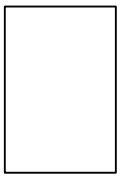
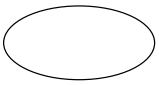
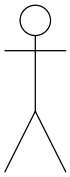
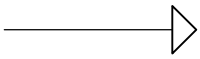
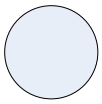
Abstract

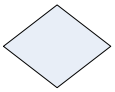
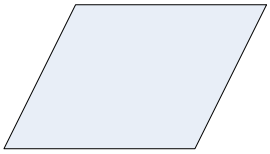
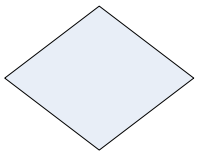
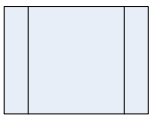
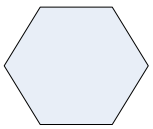
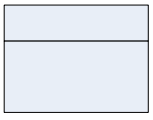
Career guidance system is a system designed for the better guidance of the users for selecting the appropriate option. This system solves the dilemmas faced by the user for the selection of the career option. There is a test module which is interacted by the user and based on the results of that test, the user is suggested to select the right option for their further studies. The system also provides various other additional facilities like providing updates through newsletter, answering FAQs and inquiries. This system is also helpful to various institutes for advertising themselves. They can offer details about various courses they offer. Admin , after logging approves institute and creates polls. Students can fill polls on various topics , give test, ask FAQs and inquiries and at the end provide feedback.

List of figures

List of tables

Notations

SYMBOL	DESCRIPTION
	System Boundary
	Use Case
	Actor
	Uses
Data Flow Diagram	
	Actor
	Data Process
	Data Flow
	Data Store
	Attribute

	Relationship
Flowchart	
	Process
	Input
	Decision
	Terminator
	Flow
	Procedure
	Decision
	Event
	Divide Process

	Start
	Stop

Naming and Coding Conventions

We used following coding standards while developing this product.

- Comments are available on top of each and every page.
- Variable are given understandable names.
- There are different event handling classes for handling events from different components.
- Function names have their first letter as capital and contain ’fuc’ as a prefix string.
- Variable are declared in a consistent manner.
- At the beginning of every routine, it is helpful to provide standard, boilerplate comments, indication the routines purpose, assumptions and limitations. A boilerplate comments should be a brief introduction to understand why the routine exists and what can it do.
- Use Hungarian naming convention, for naming the variables.
- Use Comments on functions explaining its task and working that what the function does.
- When naming tables, express the name in the singular form.

Primary key start with ‘pk’ and its value Is not null means null is not allowed for it.

Table of Contents

1.0 Introduction

1.1 Project Summary

Career guidance system will help the user to choose an appropriate career option for his future. There are number of educational courses available to select as a career, as education is occupying wide areas. Therefore students and their parents face a great dilemma to select the right option. Career guidance system suggests the user to take the respective course by offering test modules of intelligence, capabilities and areas of interest. The system also provides details about various institutes and the courses they offer in India as well as abroad. This system covers span of information from school streams, graduation degrees and post-graduation degrees.

1.2 Purpose

Goals

To develop an efficient system for guidance of institute and course selection . To provide suggestions for career selection based on one's interest and capabilities .To make the system user friendly.

Objectives

The main objective of this system is to solve the queries of the user to select an appropriate career option.

1.3 Scope

This system can be used for solving the queries of the user suggesting to select the respective field as a career option and providing various institute details.

1.4 Technology and Literature review

Literature Review:

As this is a web-based application, it requires lots of information to be stored. All the information stored should be accurate and in secured database, as this information can be used in the future for verifications.

Being a web application, there would be lots of confidential information stored in database like admin id, password etc. So security must be required.

When this information is to be maintained manually lots of difficulties will be faced related to accuracy and precision. Lots of files were prepared and paper work was very tedious. All the details should be included to obtain proper processing on the data.

Technology Review:

Front end: ASP .Net.

Back end: SQL Server

The Microsoft .NET strategy was presented by Microsoft officials to the rest of the world in June 2000:

- .NET is Microsoft's new Internet and Web strategy
- .NET is NOT a new operating system.
- .NET is a new Internet and Web based infrastructure
- .NET delivers software as Web Services
- .NET is a framework for universal services
- .NET is a server centric computing model
- .NET will run in any browser on any platform
- .NET is based on the newest Web standards

What is Microsoft .Net Framework?

The .NET Framework is Microsoft's comprehensive and consistent programming model for building applications that have visually stunning user experiences, seamless and secure communication, and the ability to model a range of business processes.

The following technologies are introduced with the .NET Framework 3.0: New compilers for C#, Visual Basic, and C++.

2.0 PROJECT MANAGEMENT

2.1 Project Planning and Scheduling

2.1.1 Project Development Approach

- For developing the project, the approach that is used is incremental model.
- Incremental model combines elements of the waterfall model in an iterative fashion.
- It applies linear sequences.
- The flow of incremental model is as follows:

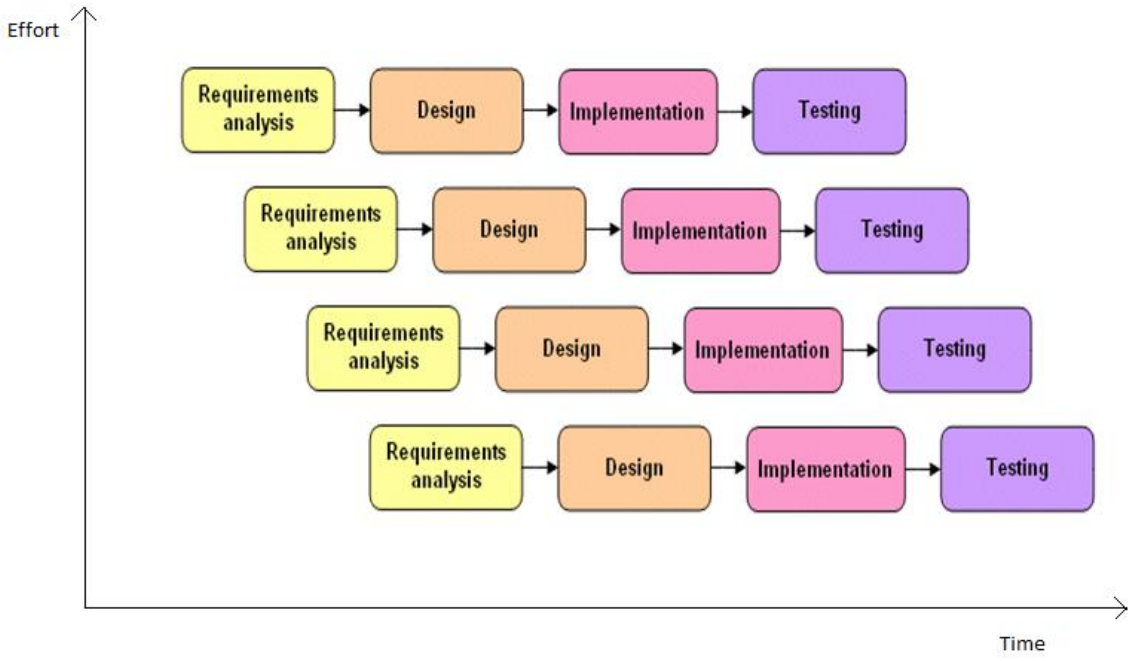


Fig 2.1: Flow of Incremental Model

- The first product is always a core product, that is basic requirement are addressed but many supplementary features remains undelivered.
- There might occur many cases where we add some features to system that adds to the development of the system.

There are basically four phases in incremental model:

- **Analysis:** During this stage, research must be done to identify the project requirements. The advantage of the current system will be that it will produce test and will reduce the manual working.
- **Design:** In detailed design, the design elements describe the desired system features in detail and generally include functional hierarchy diagrams, screen layout diagrams, business rules, business process diagrams and class diagram. These design elements are intended to describe the software in sufficient detail that programmers may develop the system with minimal additional input.
- **Coding:** After requirements and detailed design is specified, the coding for the system is developed. After every 100 lines of codes are coded, debugging process is carried out to detect error and fix it immediately. The purpose of debugging is to reduce error and to ensure the system will be executed successfully.
- **Testing:** Unit test stage will be done by programmers for checking each small module. This unit tests to make sure test template can function on the platform defined and with the expected output when specific action is taken by user.

- So, all such increments can help us to add such functionalities or new features to the system
- So the incremental model was selected for development purpose.

Advantages:

- Reusable
- Interoperable
- Up-Gradable
- Saving the programmers from complexity
- Time effective
- Cost effective
- Makes programmers Efficient
- Reliable
- Improved Quality

Disadvantages:

- Requires considerable expertise in risk evaluation and reduction
- Complex and relatively difficult to follow strictly
- Applicable only to large systems
- Risk assessment could cost more than development
- Need for further elaboration of spiral model steps (milestones, specifications, guidelines and checklists)

2.1.2 Project plan

The planning stage is first step must be done before start develop a project. The planning must refer to the suitable requirement and specification for that project that we want to develop.

Months	Plan
July	Problem Identification
Aug-Sept	Requirement Gathering And Analysis
Oct-Nov	System Design
Jan-March	Engineering Construction And Release
April	Testing and Customer Evaluation

Table 2.1: Project Plan

- Milestones:

  The milestones associated with the system is that it is the web based project and back side it calls web services so lots of validation are needed to be provided to the system. So the output produced will be the perfect match as any user wants.

- Deliverable:

  A deliverable of the project is the result that is delivered to the user usually at end of some major project phase such as specification, design, coding, testing, etc.

- Roles

The main role of the system is to acknowledge the students about various career options.

- Responsibilities:

  The main responsibility of the system is to provide remarks and suggestions based on the test results interacted by the student.

2.1.3 Schedule Representation

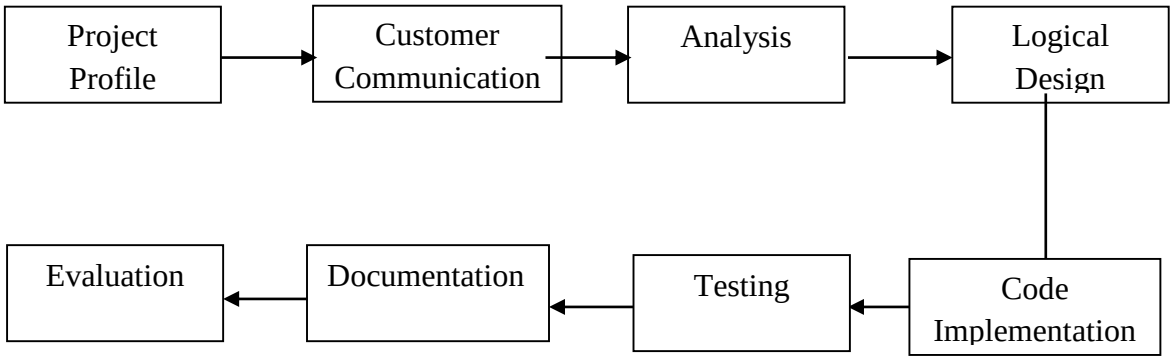


Fig 2.2: Schedule Representation

2.2 Risk Management

The art of managing the risks so that WIN-WIN situation and friendly relationship is developed between team manager and customer is called Risk management.

2.2.1 Risk Identification

Techniques that will be used to identify risk factors will be stated at the beginning of the project and on an on-going basis.

There are three main categories of risks which can affect a software project:

- Project Risks
- Technical Risks
- Business Risks

Project Risks

Project risks concern various forms of budgetary, schedule, personnel, resource and customer related problems.

1. **Miscommunication:** It leads to misunderstanding, delay, frustration, lack of coordination amongst the team members.
2. **Time shortage:** It leads to delay in the delivery of the product.
3. **Personal conflicts between team members:** It leads to unnecessary delay in each and every phase of software cycle and loss of direction.
4. **Illness or absence of team members:** Absence of a team member increases the load of the project on other team members.
5. **Lack of expertise to fulfill certain tasks:** Lack of knowledge in some areas leads to insignificant delay.
6. **Technical Advisor not available when needed:** Due to the absence of a technical advisor there was a delay in understanding the database**.**

Technical Risks:

Technical risks concern potential design, implementation, interfacing, and testing and maintenance problems.

1. **Too many planned features lead to infeasible design**
2. **Design errors:** Due to lack of experience design errors are bound to happen.
3. **The customer changes the requirements:** The scope of our project keeps changing as per user's requirements
4. **The customer disapproves of the prototype:** The customer may find the developed prototype unsuitable to his requirements

Business Risks:

Business risks threaten the viability of the software to be built. Business risks often jeopardize the project or the product.

1. **Market risk:** Building a excellent product or system that no one really wants
2. **Strategic risk:** Building a product that no longer fits into the overall business strategy for the company

3. **Management risk:** Losing the support of senior management due to a change in focus or a change in people
4. **Product is not put in service**

2.2.2 Risk Analysis

We need to identify and understand the nature of the risks. After understanding the nature of the risks, we need to prioritize the risk and on the basis of prioritization we need to solve the risks.

Personnel Shortfalls:

They may cause the developmental delays and would cause a change in the working strategy due to developmental dependencies of the functional modules.

Unrealistic Schedule:

It may cause the developers to give unrealistic commitments to the users and so lose their faith when the deliverables are not produced as per schedule.

Developing wrong Software functions:

It can be caused due to wrong requirement analysis or wrong programming method used to automate the requirements. It may cause the system to fail and not be implemented at all.

Developing wrong User Interface:

It may be caused due to lack of user acceptance and user involvement during development.
The probability of the risk might be assessed as very low (<10%), low (10-25%), moderate (20-25%), high (50-75%) or very high (>75%).

Sr No.	Risk	Probability	Effect
1	Organizational financial problem force reduction in the product budget	Low	Catastrophic
2	Required knowledge is not available	High	Catastrophic
3	Change to requirement which require major design of rework	Moderate	Serious
4	Organization is restructured so that different management are responsible for project	High	Serious
5	The time required to develop software is underestimated	High	Serious
6	User fail to understand the impact of requirement change	Moderate	Tolerate
7	The rate of defect repair is underestimated	Moderate	Tolerate
8	The size of software is underestimated	High	Tolerate
9	Misuse of the system	High	serious

Table 2.2 Risk Analysis

2.2.3 Risk Planning

The risks encountered in the project should be resolved in order to deliver the desired result to the end user. The project should be managed in such a way that risks don't affect the project in big way. Risk planning refers to the strategies to be applied and getting the desired output.

The three identified Risk types, Technical, Project, and Business all have different mitigation strategies that can be used to reduce or eliminate their impact or probability of occurrence. In the following sections, the general outline for each case is discussed.

1. **Project Risk:** In general, project risks will be minimized by realistic planning and close surveillance.
2. **Technical Risk:** Clear and concise specifications and implementation of QA provisions will minimize technical risk. Technical risks can be further minimized by exploiting previous experience to the greatest extent possible. Making deliberately conservative design choices, where possible, where new technologies are involved has minimized technical risk throughout the Project.

3. **Business Risk:** Business risks can be minimized by studying the feasibility of the project and the requirement specification closely.

2.3 Estimation

2.3.1 Effort estimation

Project Estimation proper evaluation of the system and to get the estimation of the project, it was needed to do some metrics calculation for this project. Software project metrics are the way to do this task efficiently. Project metrics allow knowing the size and complexity of the project and helping us on the planning and cost estimation.

For this project, function-oriented metrics were used to get the size of the project at the abstract level (without taking the language in consideration). For that purpose, Function Points (FP) was used.

Function Points can be counted using the following information domains:

- **Number of external inputs (EIs) –** inputs given by a user or another system
- **Number of external outputs (EOs) –** outputs derived within the system or given to another system
- **Number of external inquiries (EQs) –** as an online input that results in generation of some immediate software response in the form of an on-line output
- **Number of internal logical files (ILFs) –** logical grouping of data that resides within the application's boundary and is maintained via external inputs
- **Number of external interfaces files (EIFs) -** logical grouping of data that resides external to the application but provides data that may be of use to the application

Measurement Parameters	Count		Simple	Average	Complex		Total
			○	◉	○		
Number of user inputs	10	X	3	4	6	=	40
Number of user outputs	4	X	4	5	7	=	20
Number of user inquiries	5	X	3	4	6	=	20
Number of files	50	X	7	10	15	=	500
Number of external interfaces	2	X	5	7	10	=	14
Count=Total [UFP] Unadjusted function point							594

Question	0	1	2	3	4	5
1. Does the system require reliable backup and recovery?	○	○	○	◉	○	○
2. Are data communications required?	○	○	○	○	◉	○
3. Are there distributed processing functions?	○	○	○	◉	○	○
4. Is performance critical?	○	○	○	○	◉	○
5. Will the system run in an existing, heavily utilized operational environment?	○	○	○	○	◉	○

6. Does the system require on-line data entry?	○	○	○	○	○	◉
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	○	◉	○	○	○	○
8. Are the master file updated on-line?	○	○	○	○	◉	○
9. Are the inputs, outputs, files, or inquiries complex?	○	○	○	◉	○	○
10. Is the internal processing complex?	○	○	○	○	◉	○
11. In the code designed to be reusable?	○	○	○	○	◉	○
12. Are conversion and installation included in the design?	○	◉	○	○	○	○
13. Is the system designed for multiple installations in different organizations?	○	○	○	○	◉	○
14. Is the application designed to facilitate change and ease of use by the user?	○	○	○	○	◉	○
Degree of influence **DI TOTAL** 48						

Table 2.3 Information Domain Values

Formula FP=UFP x [0.65+0.01 x DI]

The Function Point is: 671.22

2.3.2 Cost Estimation

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level, *Basic COCOMO* is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes (*Cost Drivers*). *Intermediate COCOMO* takes these Cost Drivers into account and *Detailed COCOMO* additionally accounts for the influence of individual project phases.

Basic COCOMO

Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC)

COCOMO applies to three classes of software projects:

- Organic projects - "small" teams with "good" experience working with "less than rigid" requirements
- Semi-detached projects - "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements
- Embedded projects - developed within a set of "tight" constraints. It is also combination of organic and semi-detached projects.(hardware, software, operational, ...)

The basic COCOMO equations take the form

Effort Applied (E) = $a_b(KLOC)^{b_b}$ **[man-months]**

Development Time (D) = $c_b(\text{Effort Applied})^{d_b}$ **[months]**

People required (P) = Effort Applied / Development Time **[count]**

where, **KLOC** is the estimated number of delivered lines (expressed in thousands) of code for project. The coefficients a_b, b_b, c_b and d_b are given in the following table:

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38

Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Basic COCOMO is good for quick estimate of software costs. However it does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and so on.

Intermediate COCOMOs

Intermediate COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes. This extension considers a set of four "cost drivers",each with a number of subsidiary attributes:-

- Product reliability
- Required software reliability
- Size of application database
- Complexity of the product
- Hardware attributes
- Run-time performance constraints
- Memory constraints
- Volatility of the virtual machine environment
- Required turnabout time
- Personnel attributes
- Analyst capability
- Software engineering capability
- Applications experience
- Virtual machine experience
- Programming language experience
- Project attributes
- Use of software tools
- Application of software engineering methods
- Required development schedule

The Intermediate Cocomo formula now takes the form:

$$\mathbf{E=a_i(KLoC)^{(b_i)}.EAF}$$

where E is the effort applied in person-months, **KLoC** is the estimated number of thousands of delivered lines of code for the project, and**EAF** is the factor calculated above. The coefficient $\mathbf{a_i}$ and the exponent $\mathbf{b_i}$ are given in the next table.

Software project	$\mathbf{a_i}$	$\mathbf{b_i}$
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

The Development time **D** calculation uses **E** in the same way as in the Basic COCOMO.

Detailed COCOMO

Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

The detailed model uses different effort multipliers for each cost driver attribute.
These **Phase Sensitive** effort multipliers are each to determine the amount of effort required to complete each phase.

In detailed COCOMO, the effort is calculated as function of program size and a set of cost drivers given according to each phase of software life cycle.

A Detailed project schedule is never static.

The five phases of detailed COCOMO are:-

- Plan and requirement.
- System design.
- Detailed design.
- Module code and test.
- Integration and test.

3.0 System Requirement Study

3.1 User characteristics

It is mandatory that the user of any application should be intricately aware of how to use the application. The user interface of the application should be such that the user can easily get acquainted to it. The user must have the knowledge about using the system. The user must have a clear idea about what he is supposed to do with the system.

Mainly there are two types of users in this system:

Administrator:

He is the key person for controlling the whole system. He will give appropriate permissions to the users for performing specific tasks. He will cover areas such as database, security and integration.

End User (Student/Institute):

Basically, they are the registered users of the site. This user must be familiar with the internet and basic functionalities of the application.

3.2 Hardware and Software requirements

Hardware requirements:

- 512MB Ram
- 1 GB Free Hard disk
- Processor Pentium IV 2GHZ(or later)

Software requirements:

- Language: ASP.NET,C#,
- Database: SQL Server 2005 Express.
- Platform: .net
- OS: windows 7

3.3 Constraints

A constraint is anything that prevents the system from achieving more of its goal.

3.3.1 Hardware Limitation

It requires minimum 256MB RAM to be loaded or run. It requires IIS to browse the application.

3.3.2 Interfaces to Other Application.

Since it is a standalone application it cannot be interfaced with other applications as such.

3.3.3 Reliability Requirements.

Since the application is almost error- free, the reliability of the system is pretty high. Even in case of natural calamities or power failure or connection failure, the application is not affected as data remains secured in the corresponding database.

3.3.4 Criticality of Application.

Criticality of the application lies in the fact that when any field in any one of the six modules is empty then that particular record is not added or updated in the database. Even without proper name & password the administrator cannot access the application.

4.0 System Analysis

4.1 Study of Existing System

Existing guidance systems does more work manually, they lack good environment, and they also fail to generate reports.

4.2 Limitations of existing system

- **Manual control:** For any modification, existing e-catalog requires more manual work
- **Need of IT person:** Existing e-catalog has to deal with lot of coding work, which requires the need of IT person.
- **Time consuming:** It is more time consuming because the existing system are not automated.
- **Less Control over system**: The admin does not have total control over the existing system.

4.3 Requirement of new system

As mentioned above, due to limitations of the existing system, there was a need to develop a new system which will overcome the drawbacks of the existing system. The proposed system has following advantages:

Advantages of Proposed System:-

- The working of proposed system is dynamic.
- In the proposed system any modification made in the admin panel is directly reflected in the website.
- It is less complicated and it is less time consuming.
- There is no need of IT person for any modification.
- It is comparatively cheaper.

4.4 Feasibility Study

The purpose of feasibility study is not to solve the problem, but to determine whether the problem is worth solving. The feasibility study concentrates on the following area:

- Operational Feasibility
- Technical Feasibility
- Schedule Feasibility
- Economic Feasibility

4.4.1 Operational Feasibility

Performance

Our system provides adequate throughout and its response time is very quick. Because when any visitors see the application for searching the data, it will search form the database and display the output.

Information

The system provides end users with timely, accurate and usefully formatted information. When any manufacturer or administrator wants the information about system, he or she will just log into system. And get their desire information.

Efficiency

Does the system make maximum use of available resources including Manual Work of Administrator, time, and flows of forums, minimum processing delays & the like?

In the matter of efficiency, our system is totally computerized, so no need for any person to explain anything regarding our system, and it's very easy to understand as well as operate.

Each and every information is given very briefly so user is able to view the catalogue without any Administrators help. Here also only few person are required for moderating and controlling the system, so with the help of very less human resource and manual work system will work.

All the forms are well designed as well as developed so user can easily deal with system.

4.4.2 Technical Feasibility

It is a measure of practically of a specific technical solution and the availability of technical resource and expertise. The analyst must find out whether current technical resources, which are available in the system is capable of handling the visitor's requirements or not. If not, then the analyst with the help of developers should confirm whether the technology is available and capable or not.

Factor considered

- Here we have to consider those tools, which will be requiring for developing the project.
- The tools which are available and the tools, which will be required, have to take in account.
- As far as basic knowledge is concerned we have studied we have basic knowledge of C# and SQL server. Various technical books, e-books etc are available.
- Dealing with database is the main issues in our system. Using SQL server as backend provided this functionality.

4.4.3 Schedule Feasibility

Schedule feasibility corresponds to whether sufficient time is available to complete the project.

Factor considered

- Schedule of the project.
- Time by which the project has to be completed.
- Reporting period.

Considering all above factors it was decided that we have sufficient time and we decide to start the project. By marinating the schedule we were able to complete the project on time.

4.4.4 Economic Feasibility

For declaring that the system is economically feasible, the benefits obtained from the system we have to be rated against the cost incurred to actually develop the system. The benefits must equal or exceed the cost for development.

The basic resources to consider are:

- Management time.
- Time spent by the system analysis team.
- Cost of doing full system study.
- Estimated cost of hardware.
- Estimated cost of software and /or software development.

The following are benefits that would be derived from the proposed system:-

- The application is developed using C# and SQL server technology. Application does not increase hardware cost as minimal configuration required for developing code. In this way developing application does not generate any overhead costs.
- As this application has no overhead in development and installation but economically this application will be beneficial to user by providing them free open source user support system with common user interface on all windows platforms. In that way the application is economic.

4.5 Requirement validation

As our project is to build a dynamic web-site Carrier Guidance System, there no criteria such as windows authentication but still some security must be provided in essence of making the rights of certain entity to be limited to them. Certain validation criteria that are needed listed below.

- There are three types of users who can use the web-site in different manner with different rights, so the user cannot do the operations that are not permitted to him/her.

- Admin – It is the main user of the system. It creates test, manages the whole website and approves institute.
- Institute – It registers on the website and provides courses and advertisements.
- Student- It is the main purpose of creating the system. It applies for the test and gets guidance.

4.6 Function of system

Career guidance system is a system designed for the better guidance of the users for selecting the appropriate option. This system solves the dilemmas faced by the user for the selection of the career option. There is a test module which is interacted by the user and based on the results of that test, the user is suggested to select the right option for their further studies. The system also provides various other additional facilities like providing updates through newsletter, answering FAQs and inquiries. This system is also helpful to various institutes for advertising themselves. They can offer details about various courses they offer. Admin , after logging approves institute and creates polls. Students can fill polls on various topics , give test, ask FAQs and inquiries and at the end provide feedback.

4.6.1 Use-Case Diagram

Fig 4.1: Use Case

4.7 Data Modeling

4.7.1 E-R Diagram

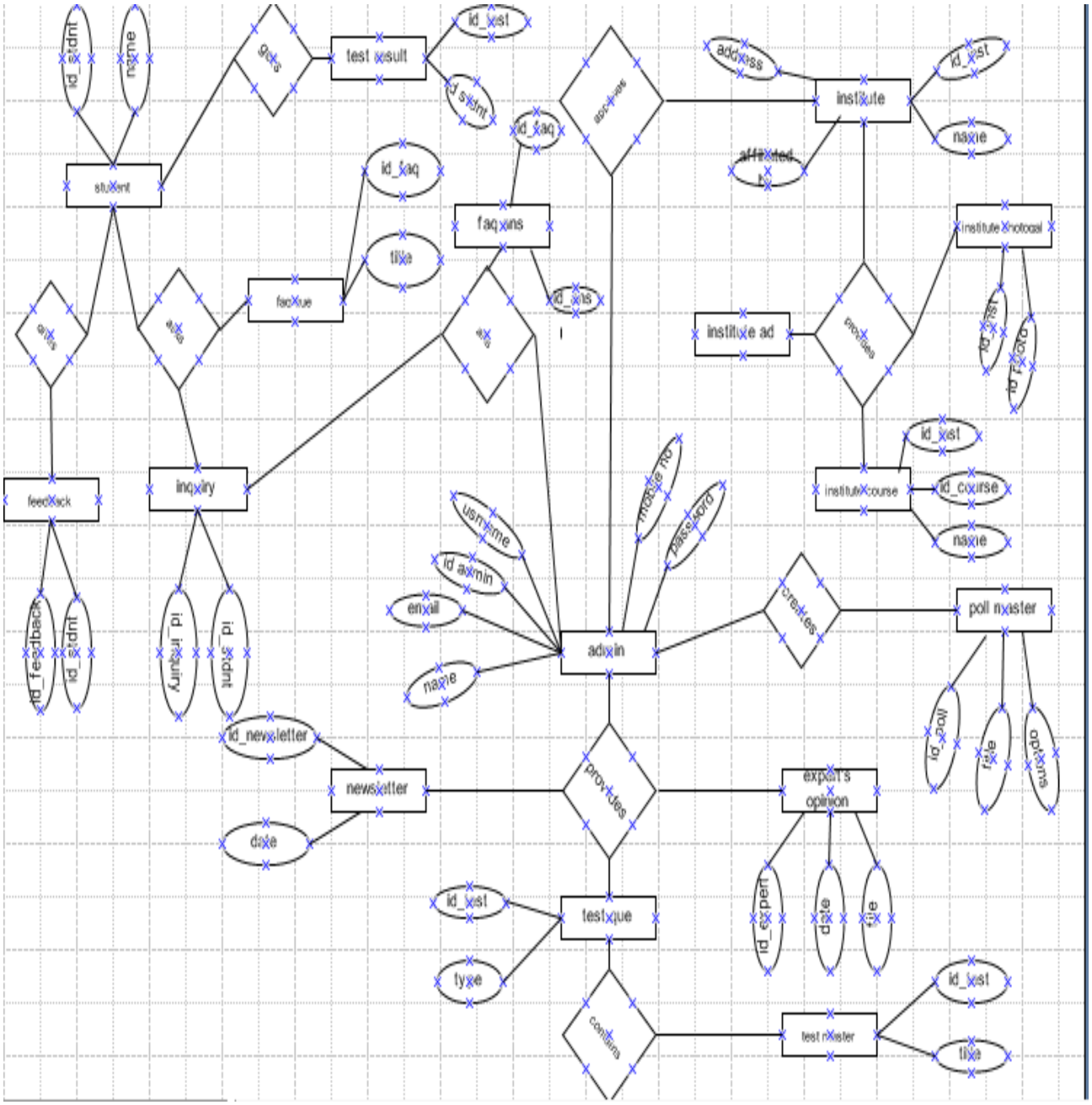


Fig 4.2: E-R Diagram

4.7.2 Activity Diagram

Admin activity diagram

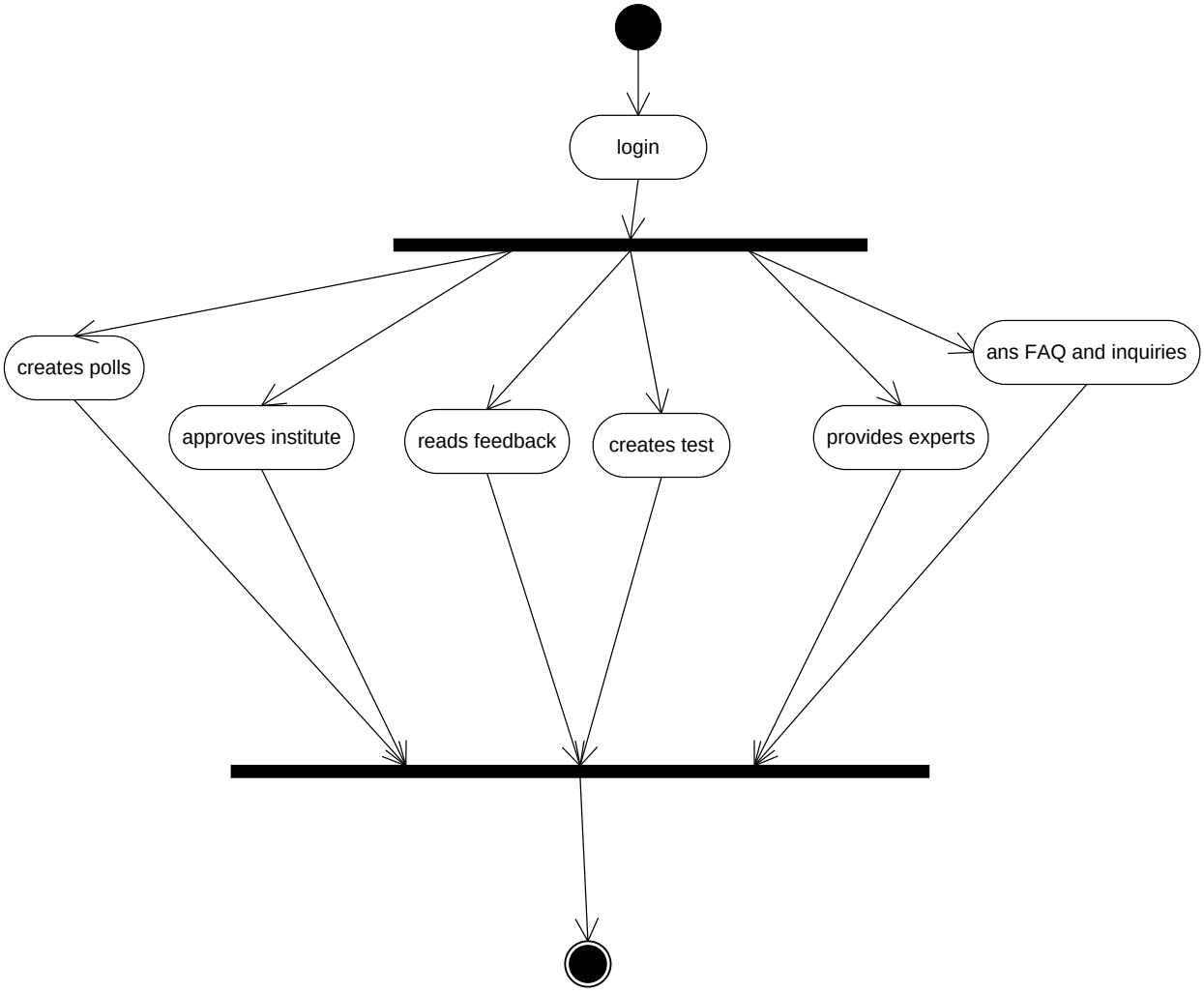


Fig 4.3 : Admin Activity Diagram

Institute Activity Diagram

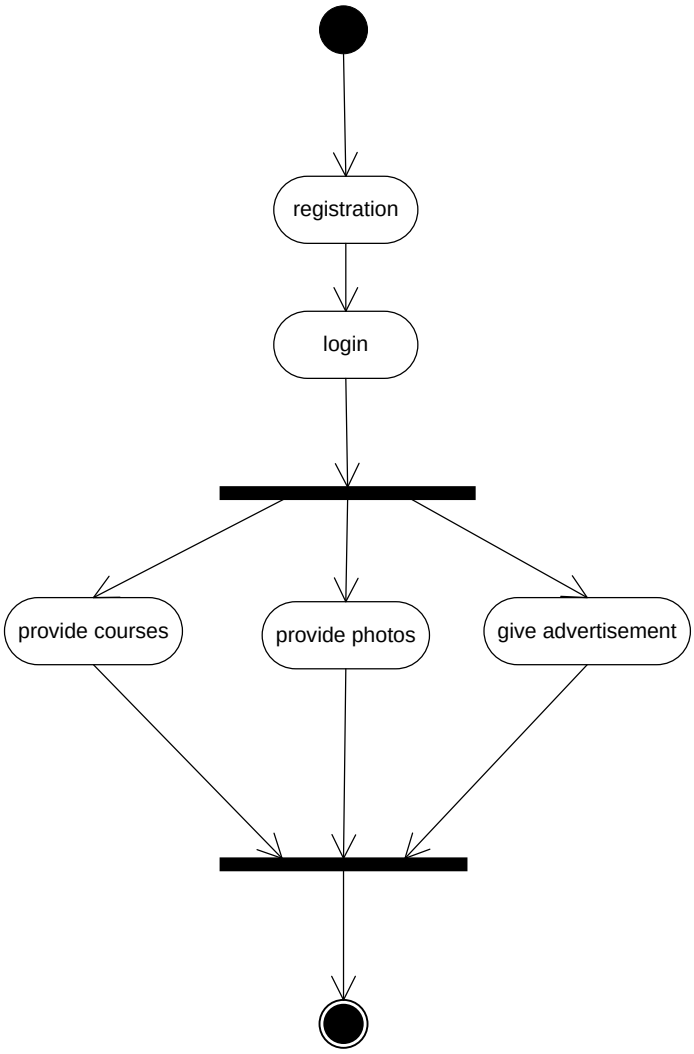


Fig 4.4 : Institute Activity Diagram

Student activity diagram

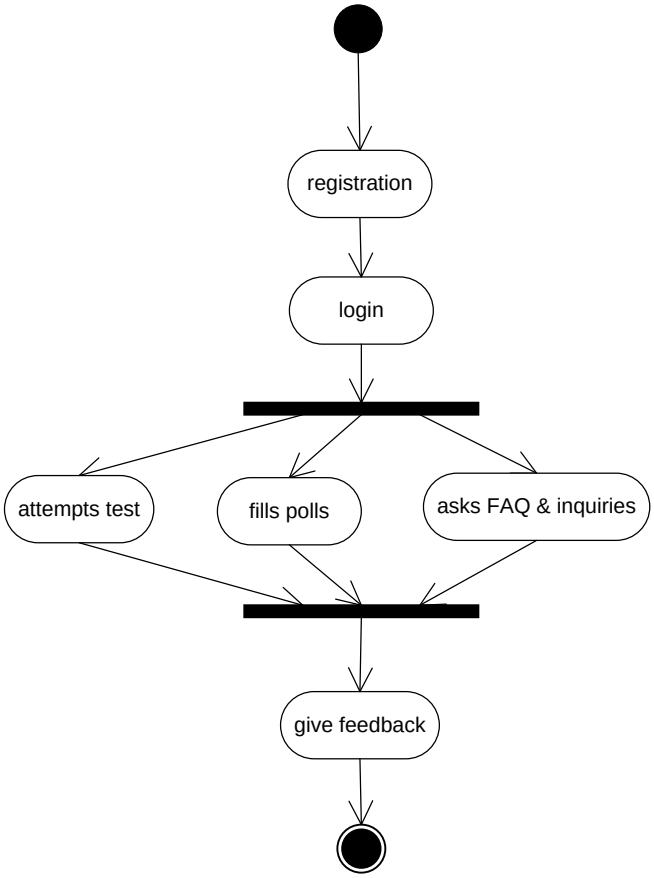


Fig 4.5: Student Activity Diagram

4.7.3 Data Dictionary

Student_master			
Fieldname	**Datatype**	**size**	**Constraints**
Id_student	Number	10	Primary key
Name	Varchar2	20	Not null
Date of birth	Number	10	Not null
Age	Number	2	Not null
Image	Varchar2	100	
Address	Varchar2	100	Not null
Recent qualifications	Varchar2	20	Not null
e-mail	Varchar2	20	Not null
City	Varchar2	30	Not null
State	Varchar2	30	Not null

Table No: 4.1 Student_master

Adminmaster			
Field name	**Datatype**	**Size**	**constraints**
Id_admin	Number	10	Primary key
Name	Varchar2	20	Not null
E-mail	Varchar2	30	Not null
Username	Varchar2	30	Not null
Password	Varchar2	30	Not null
Contactno	Number	10	Not null

Table No. 4.2 : Adminmaster

Coursemaster			
Fieldname	**Datatype**	**Size**	**Constraints**
Id_course	Number	10	Primary key
Coursename	Varchar2	30	Not null
Detail	Varchar2	100	Not null
Fees	Varchar2	10	

Table No. 4.3 Cousremaster

InstitutePhotogallery			
Fieldname	**datatype**	**Size**	**Constraints**
Id_institute	Number	10	Reference key
Id_photo	Varchar2	10	Primary key
Photofile	Varchar2	50	Not null

Table No. 4.4: InstituePhotogallery

Newsletter			
Fieldname	**datatype**	**Size**	**constraints**
Id_newsletter	Number	10	Primary key
Title	Varchar2	30	Not null
Detail	Varchar2	50	Not null
Date	Number	10	Not null
Time	Number	10	Not null

Table No. 4.5 Newsletter

FAQque			
Fieldname	**datatype**	**Size**	**Constraints**
Id_faq	Number	10	Primary key
Title	Varchar2	20	Not null
Detail	Varchar2	50	Not null
Date	Number	10	Not null
Time	Number	10	Not null

Table No. 4.6 FAQque

FAQans			
Fieldname	**Datatype**	**Size**	**Constraints**
Id_faq	Number	10	Reference key
Title	Varchar2	20	
Ans	Varchar2	50	Not null
Date	Number	10	Not null
Time	Number	10	Not null

Table No. 4.7: FAQans

Poll			
Fieldname	**Datatype**	**Size**	**constraints**
Id_poll	Number	10	Primary key
Title	Varchar2	30	Not null
poll_option1	Varchar2	50	Not null
poll_option2	Varchar2	50	Not null
poll_option3	Varchar2	50	Not null
poll_option4	Varchar2	50	Not null

Table No. 4.8: Poll

Expert			
Fieldname	**Datatype**	**Size**	**constraints**
Id_expert	Number	10	Primary key
File	Varchar2	100	Not null
Date	Number	10	Not null
time	Number	10	Not null

Table No. 4.9: Expert

Feedback			
fieldname	**Datatype**	**Size**	**constraints**
Id_feedback	Number	10	Primary key
Id_student	Number	10	Reference key
Title	Varchar2	30	Not null
Detail	Varchar2	100	Not null
Date	Number	10	Not null
Time	Number	10	Not null

Table No. 4.10 Feedback

Testmaster			
Fieldname	**Datatype**	**Size**	**constraints**
Id_testno	Number	10	Primary key
Type	Varchar2	30	Not null
Category	Varchar2	10	Not null
Noofque	Number	5	Not null
Timelimit	Number	5	Not null

Table No. 4.11 Testmaster

Testque			
Fieldname	**Datatype**	**Size**	**Constraints**
id_test number	Number	10	Primary key
Qno	Number	3	Not null
Que	Varchar2	200	Not null
Option1	Varchar2	200	Not null
Option2	Varchar2	200	Not null
Option3	Varchar2	200	Not null
Option4	Varchar2	200	Not null

Table No. 4.12 Testque

Testans			
Fieldname	**Datatype**	**Size**	**Constraints**
id_test	Number	10	Primary key
id_student	Number	10	Reference key
Score	Number	5	Not null
Result	Number	5	Not null
Remarks	Varchar2	200	Not null

Table No. 4.13 Testans

Inquiry			
Fieldname	**Datatype**	**Size**	**Constraints**
id_inquiry	Number	10	Primary key
id_student	Number	10	Reference key
about	Varchar2	50	
Date	Number	5	
Time	Number	5	

Table No. 4.14 Inquiry

Institute master			
Field name	**Datatype**	**Size**	**Constraints**
Id_institute	Number	10	Primary key
Name	Varchar2	20	Not null
Address	Varchar2	100	Not null
Helpline no	Number	20	Not null
Category	Varchar2	30	Not null
Sub-category	Varchar2	30	Not null
City	Varchar2	30	Not null
State	Varchar2	30	Not null
Country	Varchar2	30	Not null
About	Varchar2	100	Not null
Affiliatedby	Varchar2	30	Not null
website	Varchar2	30	Not null
e-mail	Varchar2	30	Not null
Username	Varchar2	30	Not null
password	Varchar2	30	Not null
Approved	Boolean		
photogal	Varchar2	100	

Country	Varchar2	30	Not null
Username	Varchar2	30	Not null
Password	Varchar2	30	Not null
Signupfornewsletter	Boolean		

Table No. 4.15 Institute master

4.8 Functional and behavioural modeling:

4.8.1. Context Diagram (DFD level 0):

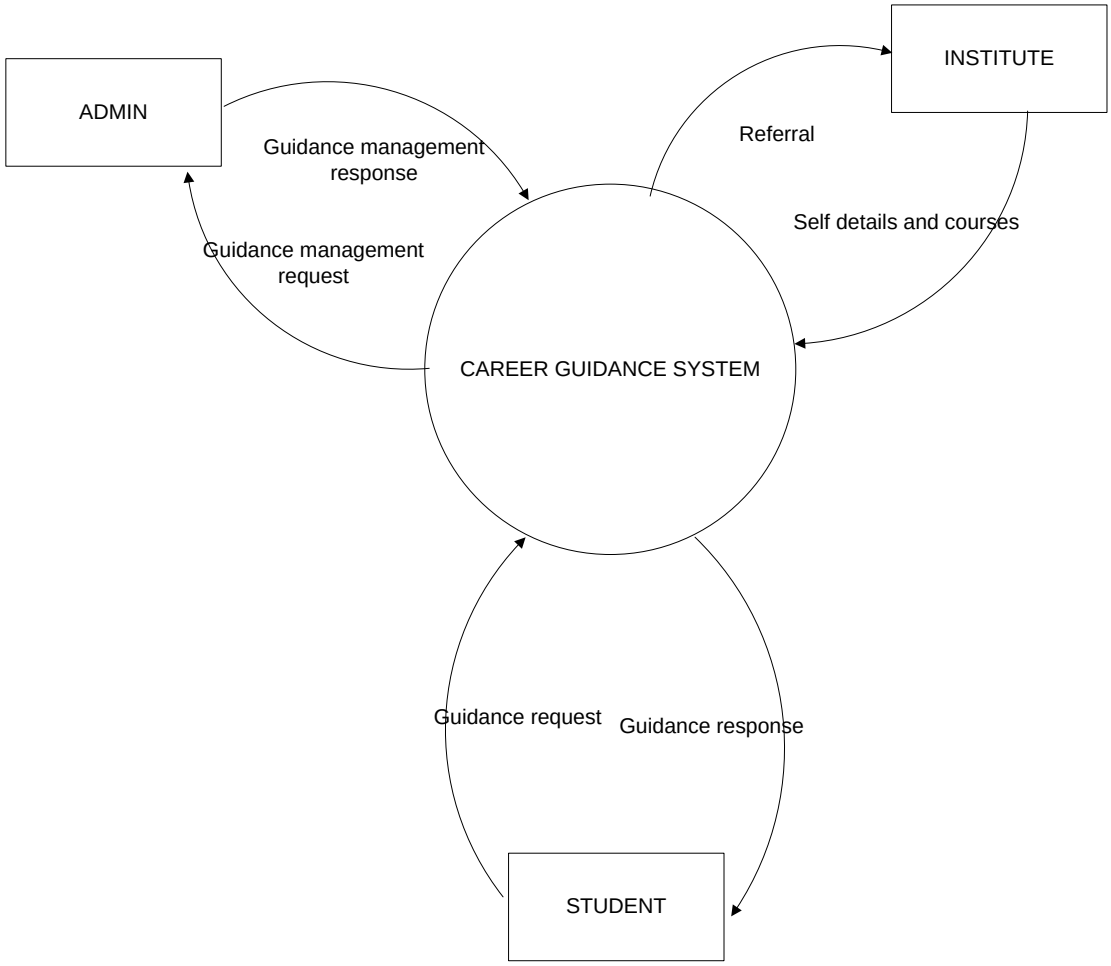


Fig 4.6: Context Diagram

DFD (Level 1)

Fig 4.7: DFD Level 1 Diagram

4.8.2Control flow diagram

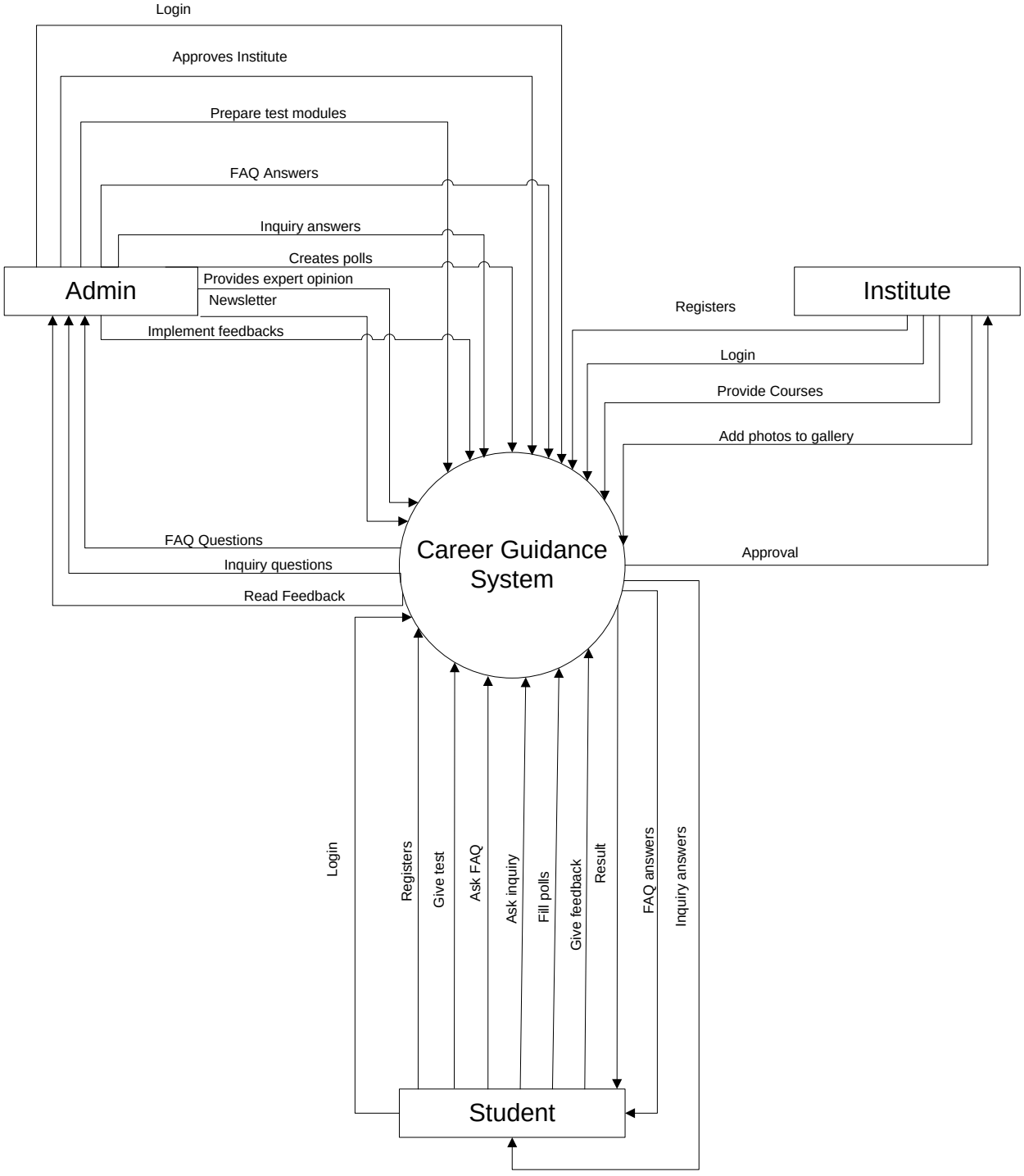


Fig 4.8: Control Flow Diagram

4.9 Main Modules

- Admin – It is the main user of the system. It creates test, manages the whole website and approves institute.
- Institute – It registers on the website and provides courses and advertisements.
- Student- It is the main purpose of creating the system. It applies for the test and gets guidance.
- Test- The main module of the system. It includes various categories of test. The actual guidance is provided on the basis of test results.
- Courses- It includes various courses provided by the various institutes.
- Newsletter- It is a facility provided by the system to update user about various courses and institutes.
- FAQ- They include various general questions asked of website by the student and answered by the admin.
- Inquiry- It is asked by the students about various institutes and courses about their specific dilemma and answered by the admin.
- Experts opinion- It is an additional facility provided by admin for better guidance.
- Polls- It displays ratings about various topics provided by the students.
- Feedback- It is the suggestion asked by the admin from the students.

4.10 Selection of hardware and software justification

4.10.1 Software

- The software selected for the modules are .NET(Asp), SQLServer Express.
- This software has been selected for this module for the following reasons:
- .NET is selected for this module because it supports multiple languages.
- .NET provides many built in functionality so we choose to develop our project in .NET
- SQL Server Express is used as database as it easier to built database in it and connect.

- SQL Server Express is used as database, because as stated earlier MYSQL is Open source there is no need to purchase license for that.

4.10.2 Hardware

- PROCESSOR : Pentium IV.
- RAM : 1GB.
- MEMORY : 80 GB HDD SPACE.
- I have used the above mentioned hardware specification in order to obtain optimized performance.

5.0 SYSTEM DESIGN

5.1 Database Design/Data Structure Design

5.1.1 Mapping Objects/Classes to tables (if non OO language)

5.1.2 Tables and Relationship

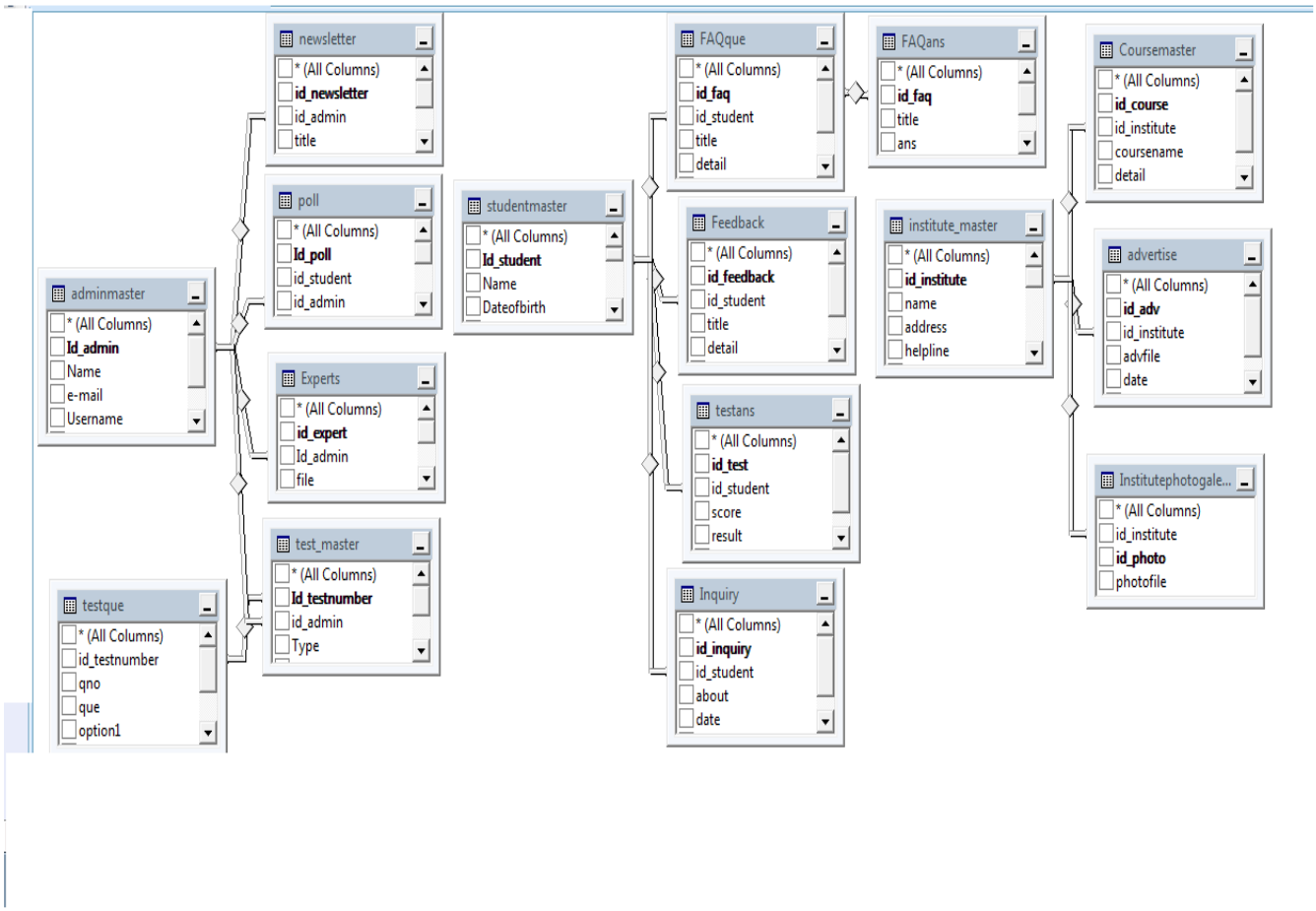


Fig 5.1 : Tables And Relationship

5.1.3 Logical Description of Data

- **Admin**

  Description

  Admin is a person who manages the system.

  Relation

  The key factor for creating tests, polls, answer in FAQs, inquiries

- **Institute**

  Description

  Institute (here user) is the user who registers to the system and handles the institute interactions behalf of the INSTITUTE that has registered for the system.

  Relation

  Institute provides self-details and courses offered by it.

- **Student**

  Description

  A student is a user who registers to the system for the guidance.

  Relation:

  A student can give tests, answer polls,ask FAQs,give feedback.

- **Login Details**

  Description

  Admin create Login Username and Password. After first time login user can change the Username and Password. Information is stored in AdminMaster

Project Diary

Description

Project Diary Describe the basic information of student project training and about his/her project. Information stored is project basic details, company name, address, project title, platform, database, internal guide, external guide, daily progress.

Relation

Project title, company name, daily progress, external guide, internal guide details.

5.2SYSTEM PROCEDURAL DESIGN:

5.2.1 Designing Pseudo code or algorithm for Method or operations

Algorithms:

Insert: Here saving of the data or record is done. The record is saved in the database.

1) Start Application Form

Fill Application Form

2) Define, create and open connection.

3) Raise the query.

ExecuteNonQuery(call the Stored Procedure)

Saved Record

Modify: Here modification of the data or record is done. The record is modified in the database and in the grid.

1) Start Application Form

Modify Application Form

2) Define, create and open connection.

3) Raise the update query

ExecuteNonQuery(call the Stored Procedure)

Update Record

Delete: Here delete of the data or record is done.

The record has been deleted from database and grid.

1) Start Grid view Form

Select Particular record

2) Define, create and open connection.

3) Raise the delete query

ExecuteNonQuery(call the Stored Procedure).

Delete Record

5.2.2 Flowcharts

Module 1: Admin

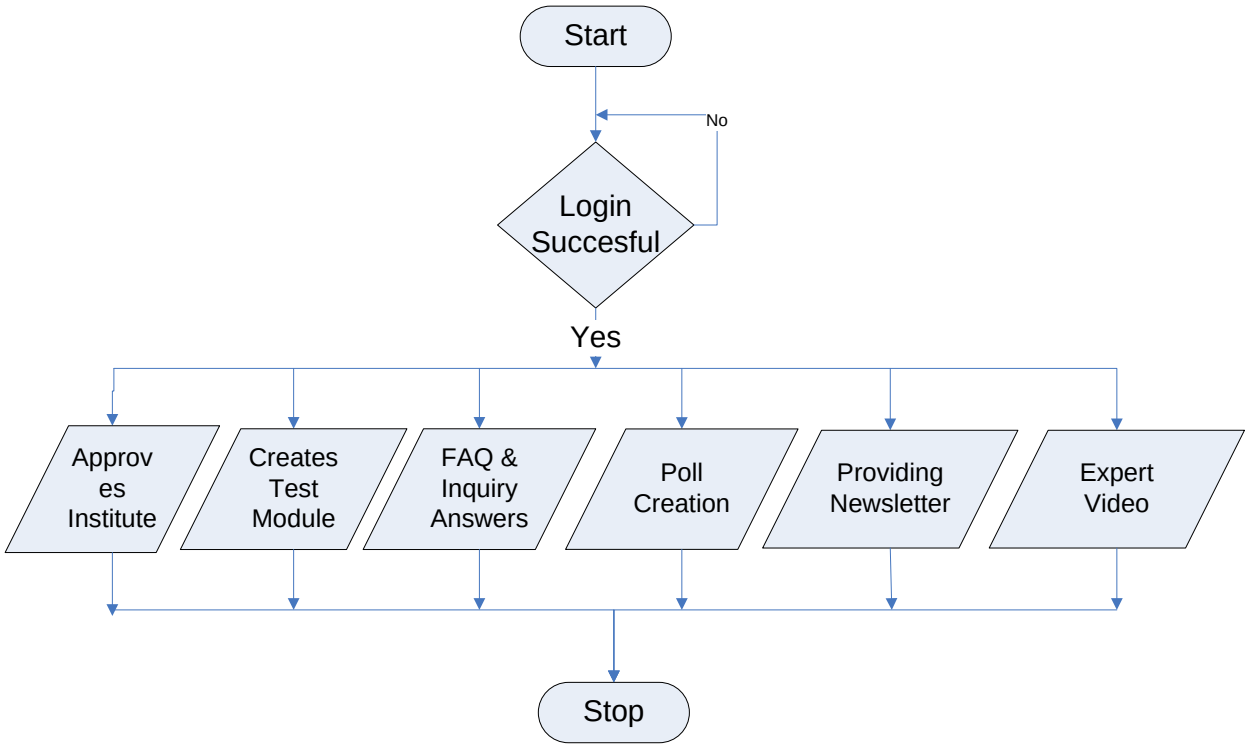


Fig 5.2 : Flowchart for admin

Module 2: Institute

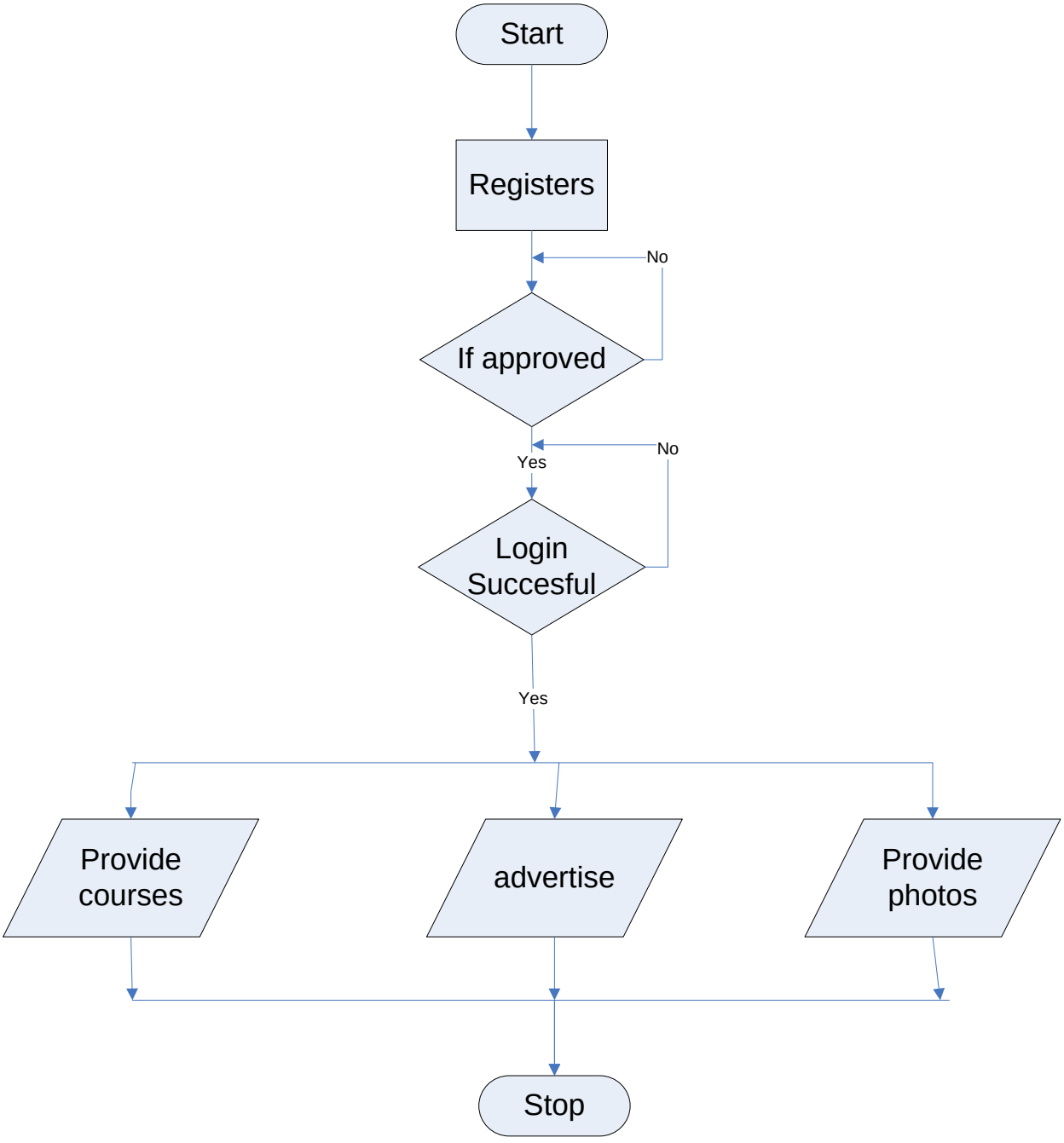


Fig 5.3 Flowchart for institute

Module 3: Student

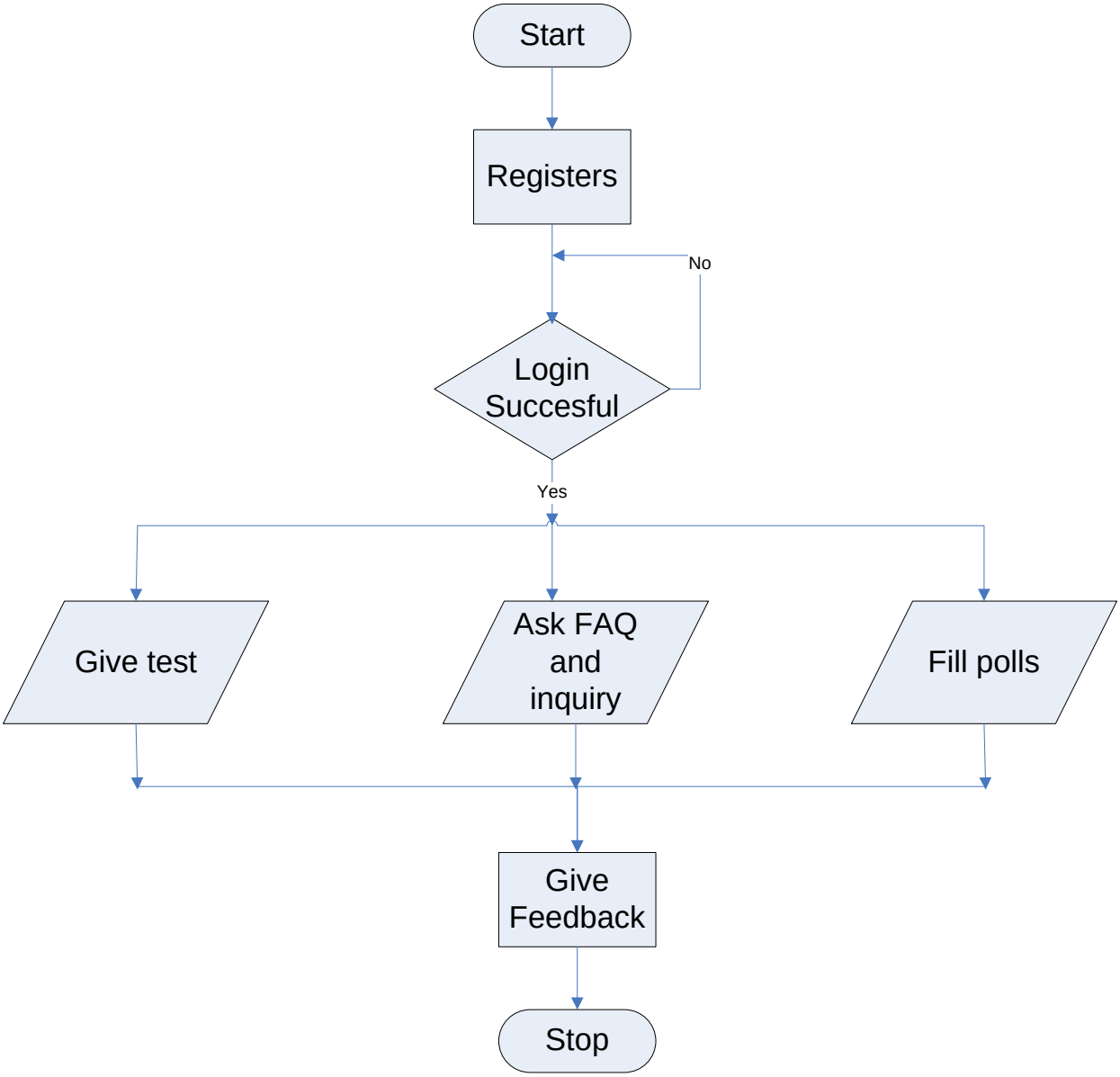


Fig 5.4: Flowchart for student

5.3 Input/output and Interface Design

5.3.1 Samples of Forms, Reports and Interface

5.3.2 Access Control & Security

Perhaps the biggest advantage of using the .NET Framework is its simplicity and compatibility with different programming languages. For example, Visual Basic and C++ are complex languages, and it's a challenge to go from one to another. The .NET Framework is very similar to both languages and easy to work with while also making it easy to transfer projects to another programming platform.

Another advantage of the .NET Framework is that it's easily integrated into multiple formats. The data can be integrated with a PC, laptop, smartphone, tablets, and more. This makes it easy to access and use the program, regardless of what hardware the user is currently on. Installation is also very easy and less problematic than other formats.

A third advantage is the .NET Framework's extensive security. Developers and the system administrators can set the security level, allowing organizations to free up the need for security since they can just set it directly. There are many options available, as everything from HTTP, XML, TCP/IP and SOAP can be used to manage the software in question.

The .NET Framework doesn't use the Windows registry when running and installing. Rather than using the registry, the data of the software is stored in the code and accessed when it's running. This results in fewer conflicts between different applications and also ensures that there are fewer problems with the registry. Now you can run multiple apps without any issues.

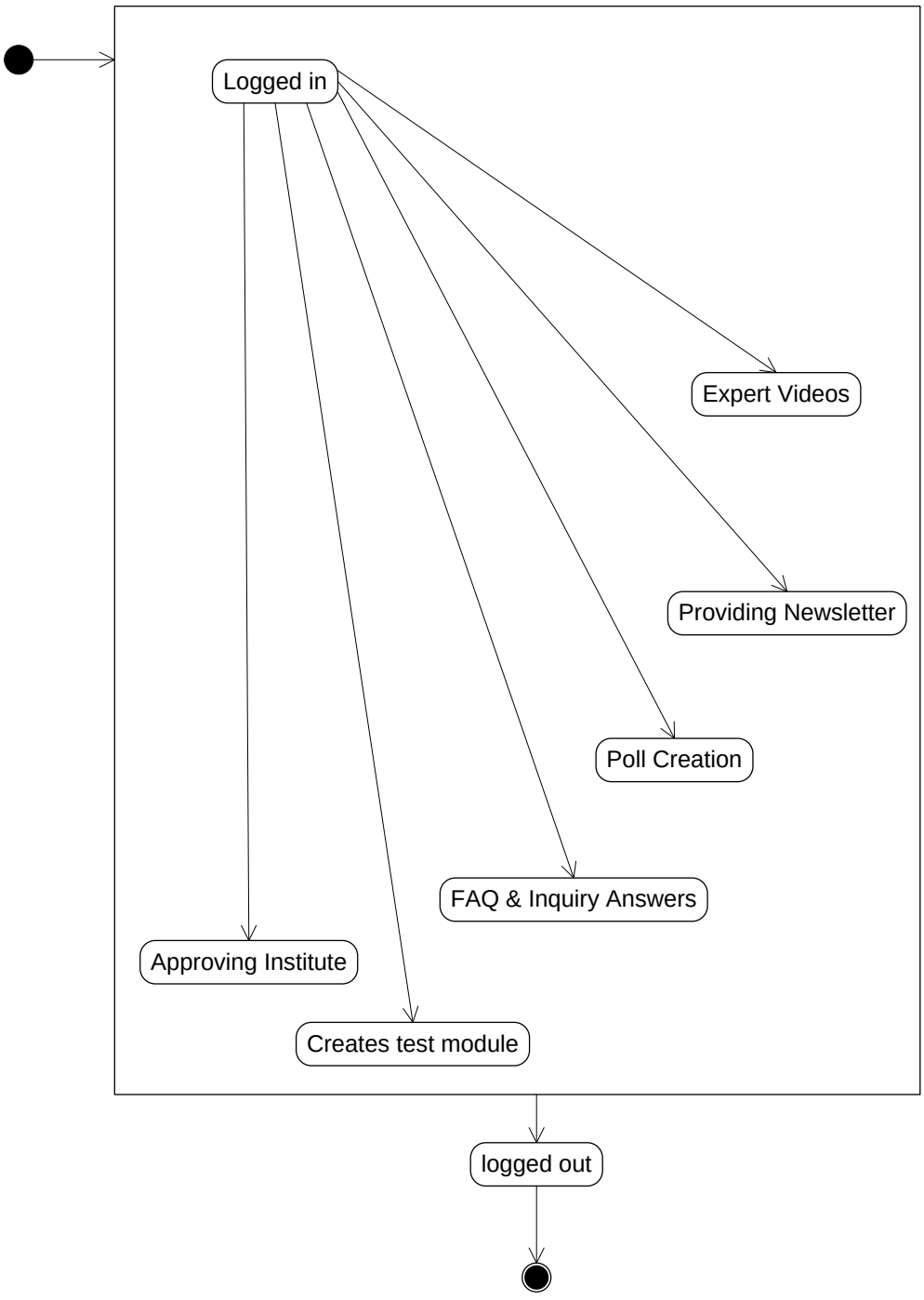
There's a huge market for online applications and plenty of tools for building them quickly while on a budget. Yet another advantage of the .NET Framework is that it allows you to quickly scale your projects and use shortcuts to create intuitive, compelling applications. There's no need to spend a small fortune on development and expect an endless development process that seems to drag on forever.

The .NET Framework offers extensive security, enhanced usability, flexibility, and fewer problems for the network. Business owners who invest in training their staff in .NET

development or hire external .NET application development partners can reap the benefits of this flexible Microsoft framework via the enterprise applications it can produce when properly leveraged.

5.3.3 State transition diagrams

Admin state transition diagram


Logged in
Expert Videos
Providing Newsletter
Poll Creation
FAQ & Inquiry Answers
Approving Institute
Creates test module
logged out

Institute state transition diagram

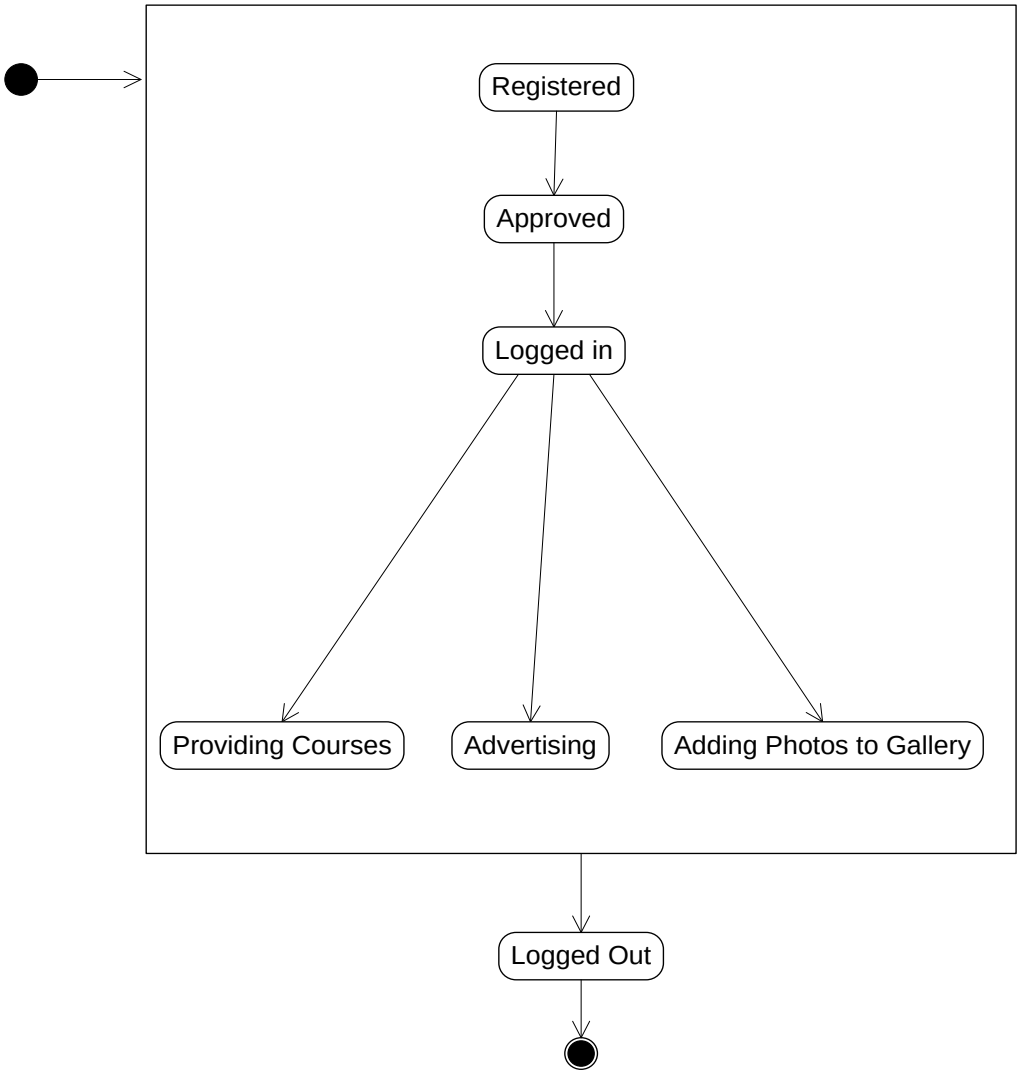


Fig 5.5 Institute State Transition Diagram

Student state transition diagram

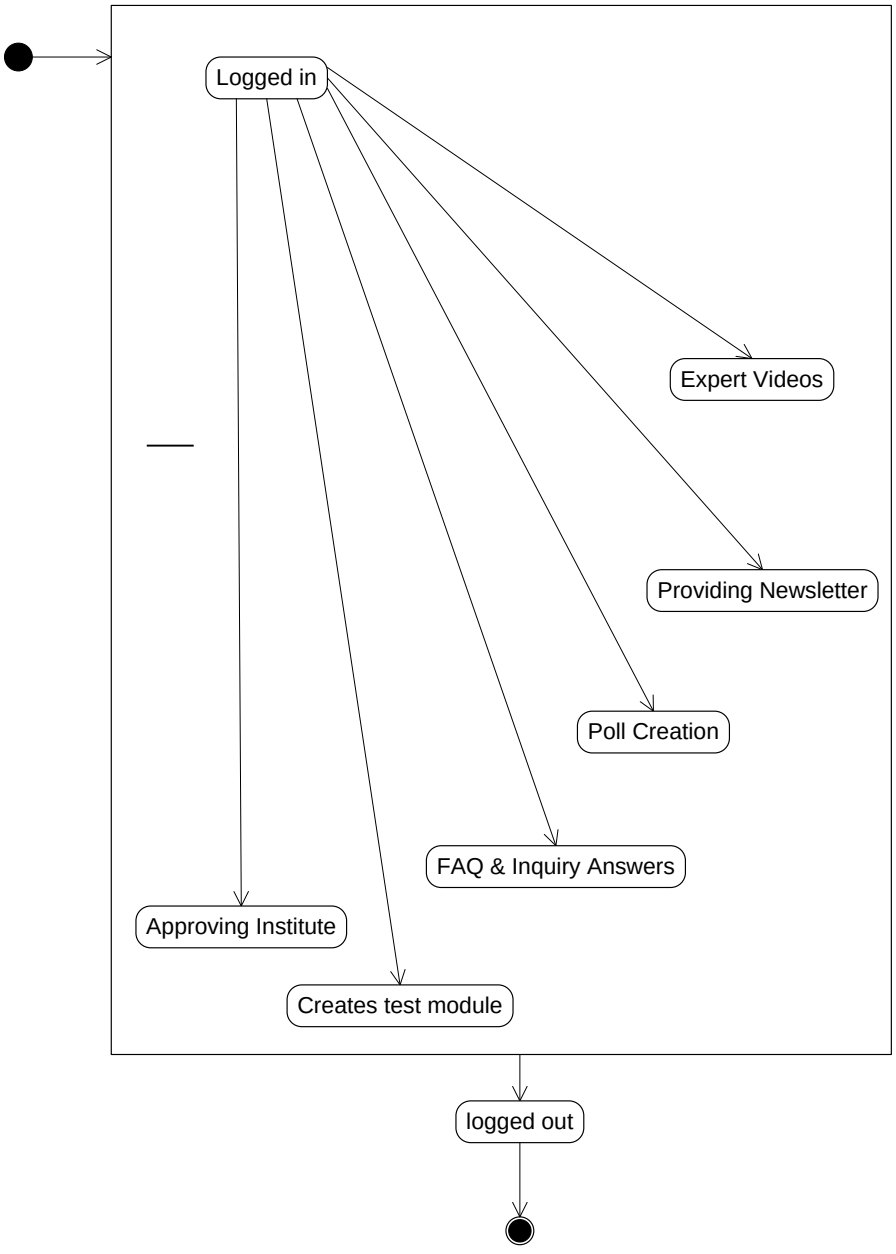


Fig 5.6: Student State Transition Diagram

5.3.4 System Design Architecture

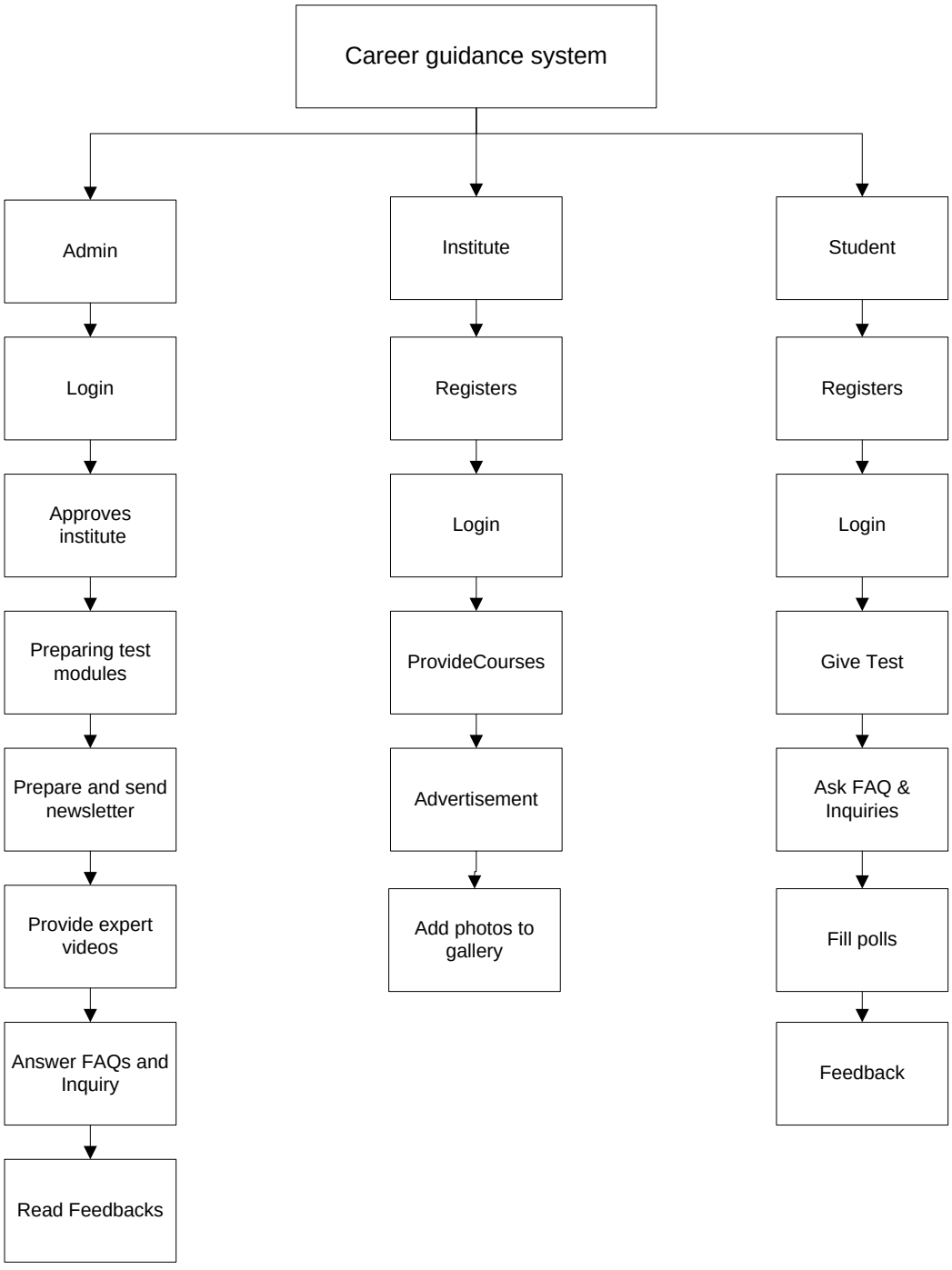


Fig 5.7: System Design Architecture

6.0 Implementation Planning and Details

6.1 Implementation Environment (Single vs Multiuser, GUI vs Non GUI)

6.1.1 Single vs. Multi-user

Career Guidance system is web application that displays various career options online. It is going to be used in distributed environment where any number of users like Students, Institutes and admin can access it from anywhere at the same time and get the information related to various courses and institutes where that courses are done. As various users can access our application one time it is Multi user application.

6.1.1 GUI vs. Non-GUI

Career Guidance System is GUI. This is because it serves large number of images to advertise the institutes. As the project is web based, it can be used by any end-user. It is mainly built in a way that any end user can feel convenient while using this web application. Thus system is user-friendly.

6.2 Program/Modules Specification

Career Guidance System has 11 different modules which are useful when any user interacts with the system. Those important modules are as follows:

MODULE DESCRIPTION

- **ADMIN**: It is the main user of the system. It creates test, manages the whole website and approves institute.

- **INSTITUTE**: It registers on the website and provides courses and advertisements.

- **STUDENT**: It is the main purpose of creating the system. It applies for the test and gets guidance.
- **TEST**: The main module of the system. It includes various categories of test. The actual guidance is provided on the basis of test results.

- **COURSES**: It includes various courses provided by the various institutes.

- **NEWSLETTER**: It is a facility provided by the system to update user about various courses and institutes.

- **FAQ:** They include various general questions asked of website by the student and answered by the admin.

- **INQUIRY**: It is asked by the students about various institutes and courses about their specific dilemma and answered by the admin.

- **EXPERT OPINION**: It is an additional facility provided by admin for better guidance.

- **POLLS**: It displays ratings about various topics provided by the students.

- **FEEDBACK**: It is the suggestion asked by the admin from the students.

6.3 Security Features

Following security features are provided in our system:

- First thing because it is a three tier or say a Model View Controller type of model it provides a secure database connectivity.
- Secondly the Admin has the only authority as he logs in to the system. And he is the only one who can change or modify the system. So no changes can be made to the Career Guidance System.

- Also the Institute is provided with login. So no random institutes can enter the system and misguide the students.

6.4 Coding Standards

The Coding Standard is the well-defined and standard style of coding. Coding standards define how a variable is to be declared, how code file is to be placed, how to define functions, how to place comments. It provides a standard way to code any document. It is good to implement coding standards and following coding standards are easy to understand.

We have used 3-tier architecture for developing this system, so all the layers become loosely coupled and we can make changes and test them independently. Basically we have divided Coding part in 3 different layers. Those are as follows:

- **Client tier :**
  Client tier or presentation layer contains pages like .aspx where data is presented to the user or input is taken from the user.
- **Business logic tier:**
  Business logic tier contains business logic, validations or calculations related with the data if needed.
- **Database tier :**
  Database layer contains methods that helps business layer to connect the data and perform required action, might be returning data or manipulating data (insert, update, delete etc).

Suggested Prefixes for Variables:

- Declaration of all public variables for the module form together at the beginning of the code. Global variables are as little as possible.
- Use Meaningful, descriptive words to name variables. Do not use abbreviations.

Good:

string address

int salary

Not Good:

stringnam

intsal

- Do not use single character variable names like i, n, s etc. Use names like index, temp
- Do not use underscores (_) for local variable names.
- All member variables must be prefixed with underscore (_) so that they can be identified from other local variables.
- Do not use variable names that resemble keywords.
- Prefix boolean variables, properties and methods with "is" or similar prefixes
  Ex: private bool _isFinished

Function / Methods:

- The names of the method or function are specified in such a way that characterizes its functionality or purpose.
- The name of the method or function have Pascal Casing format i.e. Camel Case.
- A method should do only 'one job'. Do not combine more than one job in a single method, even if those jobs are very small.

For example:

```
public void bindcategory()
    {
      // this method binds category with the fields. The only task is binding the main
            categories with the dropdown list.
    }

public void bindfield(string Prod_code)
```

```
    {
     // This method binds the passed product code in the argument with the fields of
                that page.

}
```

Following description describes how the above code cannot be appropriate

If we perform the task of binding the category and also the other fields in the same method, then the code is inappropriate.

Scope of the Variable:

- Declaration of all public variables for the module or form together at the beginning of the code.
- If required initialization also should be done along with declaration.

Error Handling:

- Use try-catch-finally in all codes for handling errors.
- Wherever required the emails should be send to the technical team and/or system admin regarding the exceptions/errors.
- Always catch only the specific exception, not generic exception.

Good:

```
voidReadFromFile ( string fileName )
{
        try
        {
                // read from file.
        }
        catch (FileIOException ex)
        {
                // log error.
```

```
                    //  re-throw exception depending on your case.
                    throw;
                }
            }
```

Not Good:

```
        voidReadFromFile ( string fileName )
         {
                try
                  {
                    // read from file.
                  }
                catch (Exception ex)
                  {
                    // Catching general exception is bad... we will never know whether
                   // it was a file error or some other error.
                  // Here you are hiding an exception.
                 // In this case no one will ever know that an exception happened
                return "";
          }
        }
```

- When you re throw an exception, use the throw statement without specifying the original exception. This way, the original call stack is preserved.

  Good:

```
        catch
        {
                // do whatever you want to handle the exception

                throw;
        }
```

Not Good:

```
catch (Exception ex)
{
        // do whatever you want to handle the exception

        throw ex;
}
```

- Write your own custom exception classes if required in your application. Do not derive your custom exceptions from the base class System Exception. Instead, inherit from Application Exception

Indentation and Spacing:

- Use TAB for indentation. Do not use SPACES.
- Comments should be in the same level as the code (use the same level of indentation)

Good:

```
// Format a message and display

stringfullMessage = "Hello " + name;
DateTimecurrentTime = DateTime.Now;
string message = fullMessage + ", the time is : " +
currentTime.ToShortTimeString();
MessageBox.Show( message );
```

Not Good:

```
// Format a message and display
        stringfullMessage = "Hello " + name;
        DateTimecurrentTime = DateTime.Now;
```

```
string message = fullMessage + ", the time is : " +
currentTime.ToShortTimeString();
MessageBox.Show( message );
```

- Curly braces ({}) should be in the same level as the code outside the braces.

```
if ( ... )
{
        // Do something
        // ...

        return false;
}
```

- Use one blank line to separate logical groups of code.

Good:

```
boolSayHello ( string name )
{
        stringfullMessage = "Hello " + name;
        DateTimecurrentTime = DateTime.Now;

        string message = fullMessage + ", the time is : " +
        currentTime.ToShortTimeString();

        MessageBox.Show( message );
        if ( ... )
```

```
        {
            // Do something
            // ...

            return false;
        }

        return true;
    }
```

Not Good:

```
    boolSayHello (string name)
    {
        stringfullMessage = "Hello " + name;
        DateTimecurrentTime = DateTime.Now;
        string message = fullMessage + ", the time is : " +
        currentTime.ToShortTimeString();
        MessageBox.Show( message );
        if ( ... )
        {
            // Do something
            // ...
            return false;
        }
        return true;
    }
```

- The curly braces should be on a separate line and not in the same line as if, for etc.

Good:

```
if ( ... )
{
        // Do something
}
```

Not Good:

```
if ( ... )      {
        // Do something
}
```

- Use a single space before and after each operator and brackets.

Good:

```
if ( showResult == true )
{
        for ( int index = 0; index < 10; index++ )
        {
                //
        }
}
```

Not Good:

```
if(showResult==true)
```

```
{
        for(int         i= 0;i<10;i++)
        {
                //
        }
}
```

Type Casting:

- Type casting should be done appropriately whenever it is required.

Comments, Change history and assumptions:

Good and meaningful comments make code more maintainable. However,

- Code should be properly commented
- The follow of the code should be made clear by the comments on the page
- The commenting should be indented as the code and should not hang out, as this beats the purpose of indenting the code.
- The effect of this is more visible when the level of nesting of the blocks is deep, say more then 2-3 levels.
- Do not write comments for every line of code and every variable declared.
- Use // or /// for comments. Avoid using /* … */
- Write comments wherever required. But good readable code will require very less comments. If all variables and method names are meaningful, that would make the code very readable and will not need many comments.
- Do not write comments if the code is easily understandable without comment. The drawback of having lot of comments is, if you change the code and forget to change the comment, it will lead to more confusion.
- Fewer lines of comments will make the code more elegant. But if the code is not clean/readable and there are less comments, that is worse.
- If you have to use some complex for any reason, document it very well with sufficient comments.
- If you initialize a numeric variable to a special number other than 0, -1 etc, document the reason for choosing that value.

- The bottom line is, write clean, readable code such a way that it doesn't need any comments to understand.
- Perform spelling check on comments and also make sure proper grammar and punctuation is used.
- Change history needs to be maintained after the first release of the system/application (and not while initial development). This makes it easy to analyze the changes that may have been made to the code and also specifies the sequence in which they have been made

Code reuse, optimization and modularization :

- If some part of code (a set of statements/ functionality) is required throughout the page, then a function / method should be written that would perform the functionality instead of writing the same group of statements again and again.

Cross-Browser Compatible:

- All the client side JavaScripts are cross-browser compatible.
- All are required to use cross-browser compatible methods instead of direct JavaScript methods wherever possible as provide by the team lead/project lead.

6.5 Sample Coding

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

public partial class Student_c : System.Web.UI.Page
{
    Class1 x = new Class1();
    protected void Page_Load(object sender, EventArgs e)
    {

        x.connect();

        if (Session["score"]==null)
            Session["score"] = 0;
```

```
        if (!IsPostBack)
        {

            if (Request.QueryString["Id_testno"] != null)
            {
                testname.Text = Request.QueryString["Id_testno"].ToString();
                DataTable dt = new DataTable();


                string q = "select * from testque where Id_testno=" + testname.Text +
" order by qno";
                dt = x.display(q);
                qno.Text = dt.Rows[0]["Qno"].ToString();
                Quesname.Text = dt.Rows[0]["Que"].ToString();
                opt1.Text = dt.Rows[0]["Option1"].ToString();
                opt2.Text = dt.Rows[0]["Option2"].ToString();
                opt3.Text = dt.Rows[0]["Option3"].ToString();
                opt4.Text = dt.Rows[0]["Option4"].ToString();
                rightans.Text = dt.Rows[0]["Rightans"].ToString();

            }

        }

    }
    protected void next_Click(object sender, EventArgs e)
    {
        string givenans = "";
        int s;
        if (opt1.Checked)
        {
            givenans = opt1.Text;
        }
        else if (opt2.Checked)
        {
            givenans = opt2.Text;
        }
        else if (opt3.Checked)
        {
            givenans = opt3.Text;
        }
        else if (opt4.Checked)
        {
            givenans = opt4.Text;
        }

        if (givenans == rightans.Text)
        {
            s = int.Parse(score.Text) + 1;
            score.Text = s.ToString();
        }

        qno.Text = (int.Parse(qno.Text) + 1).ToString();
        string q = "select * from testque where qno=" + qno.Text + " and id_testno=" +
Request.QueryString["Id_testno"].ToString();

        DataTable dt = new DataTable();
        dt = x.display(q);

        if (dt.Rows.Count == 0)
        {
```

```
            Session["score"] = score.Text;
            Response.Redirect("result.aspx");

        }
        else
        {
            opt1.Checked = false;
            opt2.Checked = false;
            opt3.Checked = false;
            opt4.Checked = false;

            qno.Text = dt.Rows[0]["Qno"].ToString();
            Quesname.Text = dt.Rows[0]["Que"].ToString();
            opt1.Text = dt.Rows[0]["Option1"].ToString();
            opt2.Text = dt.Rows[0]["Option2"].ToString();
            opt3.Text = dt.Rows[0]["Option3"].ToString();
            opt4.Text = dt.Rows[0]["Option4"].ToString();
            rightans.Text = dt.Rows[0]["Rightans"].ToString();

        }

    }
}
```

7.0 Testing

Testing is the process of executing a program with the explicit intention of finding errors, which makes the program fail. The tester is actually trying to make the program fail. A successful test is the one that finds errors.

Software engineers are by their nature constructive people. Testing requires that the developer discards preconceived notions of the " corrected " of software just developed and overcome a conflict of interest that occurs when errors are uncovered.

Testing plays an important role in the System Development Life Cycle. Testing ensures the correctness of the system performance, before being deployed to the user's machine.

Many times this aspect of System Development Life Cycle is not concentrated upon and this ends developer with the laborious maintenance task. Moreover, the small errors when encountered by user make him unsatisfied and henceforth the quality of the system is affected.

So, the testing activity should be well planned. Test cases to test each and every aspect of software must be well formed.

7.1 Test Plan

A test plan document the strategies that will be use to verify and ensure that a product or system meets its design specification. A test plan is prepared by a test engineer.

Test planning can begin early in the software process. Test plan document format can changed as the product or organization changes but there are three major elements as follows.

1. Test Coverage :

   Test coverage in test plan states what requirements will be verify during what stages of the product lifecycle.

2. Test Method :

   Test method in test plan states how test coverage will be implemented.

3. Test responsibilities :

   Test responsibilities includes what data to be collected and how that data will be stored and reported.

7.1.1 Testing Objectives

There are number of rules that can serve well as testing objectives. Testing is the 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 undiscovered error. A successful test is one that uncovers undiscovered error. For testing of our software we have adapted two main levels of testing. The levels of testing are:

1. Unit Testing
2. System Testing

1. Unit Testing

This is use to test particular functions or code modules. Typically done by the programmer and not by testers so it is called the program testing. It requires detailed knowledge of the internal program design and code. Not always easily done unless the application has a well – designed architecture with tight code; may require developing test driver modules or test harness. This type of testing is done after development each and every unit of system. There are two types of the unit testing as follows.

- Bottom-up testing:

  This type of testing start from the smallest and lowest level modules.

- Top-down testing:

  This type of testing being with the upper level modules.

2. System Testing

The system testing is carried out to see that entire software works correctly as per the functions, which are specified during the requirement analysis.

System testing is testing conducted on a complete, integrated system. The system testing should require no knowledge of the inner design of the code or logic.

System test are designed to validate a fully developed system to assure that it meets it requirements. There are three mankind of system testing.

- Alpha testing :

  Alpha testing refers to the system testing carried out by the test team within the developing organization.

- Beta testing :

  Beta testing is the system testing performed by a select group of friendly customers.

- Acceptance testing :

  This testing is the system testing performed by the customer to determine whether to accept or reject the delivery of the system.

7.2 Testing Strategy

The purpose of test management is to ensure that a testing strategy is both devised and applied that is efficient, effective and economic. The testing strategy should define the objectives of all test stages and the techniques that apply. The testing strategy also forms the basis for the creation of a standardized documentation set, and facilitates communication of the test process and its implications outside of the test discipline. Any test support tools introduced should be aligned with, and in support of the test strategy.

7.3 Testing Methods

Out of the above discussed testing methods, we adopted the Integration Testing and Unit Testing strategies. We divided all the main module of our project in the manageable units, prepared them individually and then tested them as per their working strategy. After the individual testing of each such module, we integrated them module wise and tested the whole modules individually. When we were satisfied with the working of each module with the other one. Hence, we travelled from the smallest manageable units to the whole project testing.

7.4 Test Case

A test case is a set of condition or variables under which a tester will determine whether an application or software system is working correctly or not. Test cases are after referred as test script particularly when they are written. There are three types of test cases.

a) Formal test cases
b) Informal test cases
c) Written test case formats

Information that may be included in a test case are as under.

- Test case ID.
- Test case description.
- Test step or order of execution.
- Related requirement.
- Test category.
- Author.
- Pass or fail (test oracle)
- Remarks.

Generally test cases are not written but the activities & results are reported after test have been run. Generally for each use case a test case is generated.

8.0 SCREENSHOTS AND USER MANUAL

8.1 ADMIN SNAPSHOTS

- **ADMIN LOGIN**

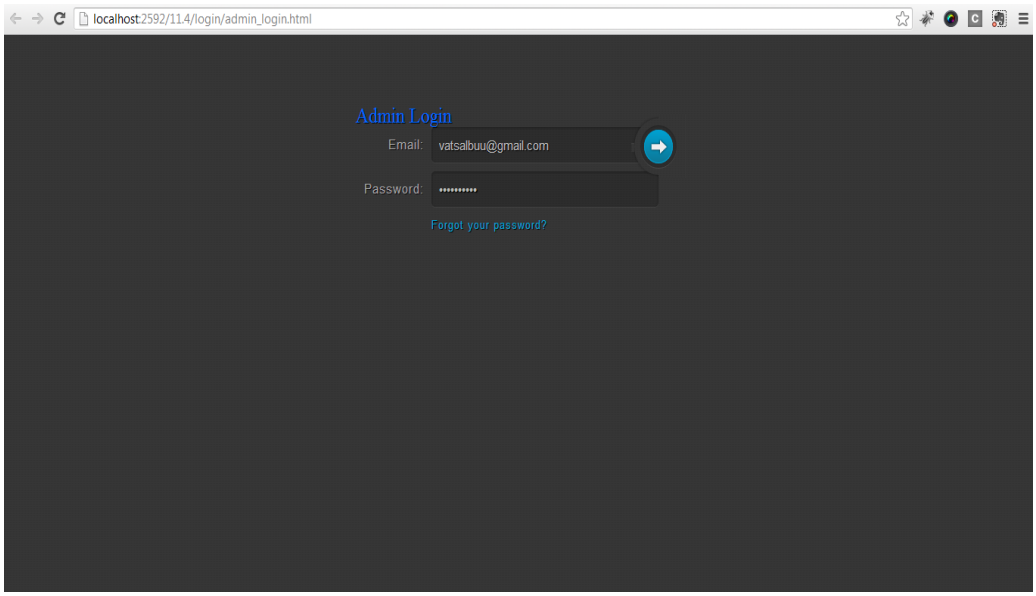


Fig 8.1: Admin Login

From this page the admin can login into our system with username and password.

- **ADMIN HOME**

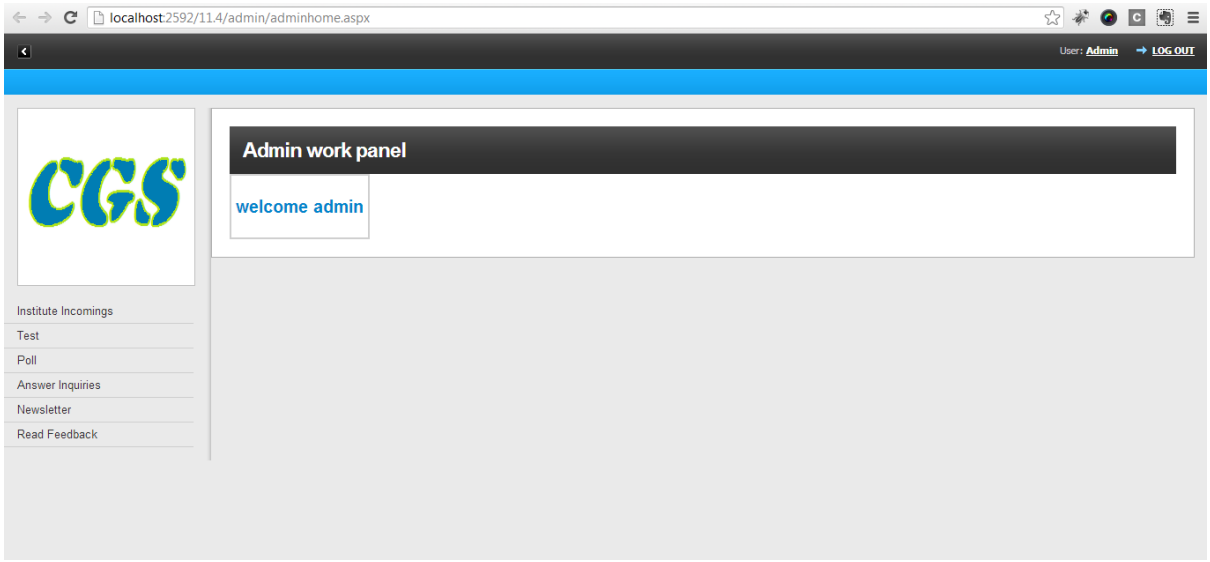


Fig 8.2: Admin Home

When the admin logs in to our system, this page is displayed. It provides the tasks that can be performed by the admin.

- **ADMIN INSTITUTE INCOMINGS**

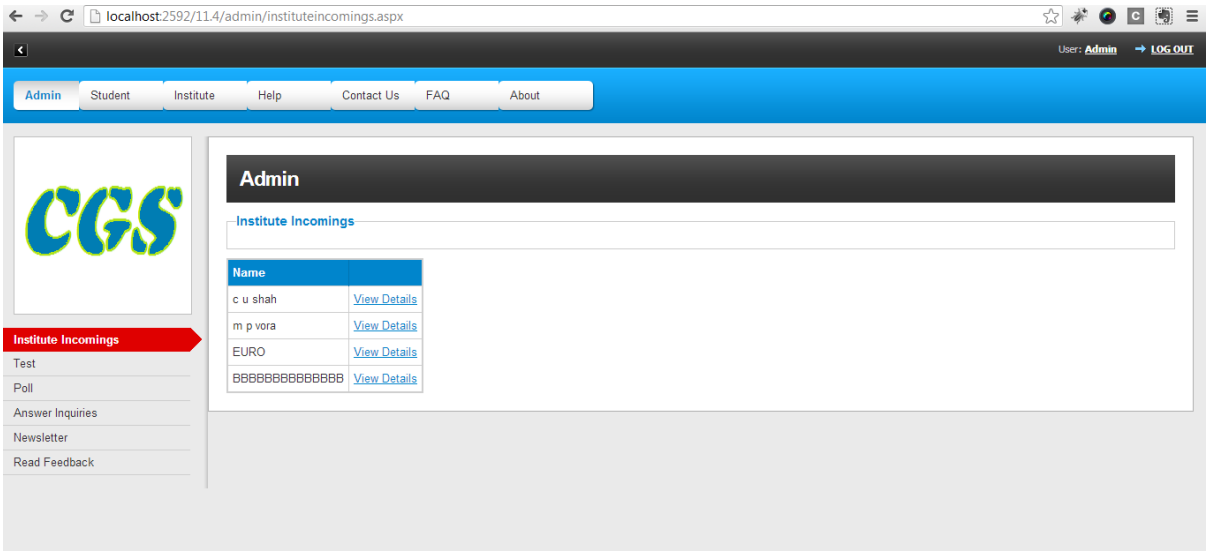


Fig 8.3: Admin Institute Incomings

This page gives a list of all the institutes that has requested to register in to our system. After the approval of the admin the institute will be registered in our system.

- **ADMIN INSTITUTE DETAILS**

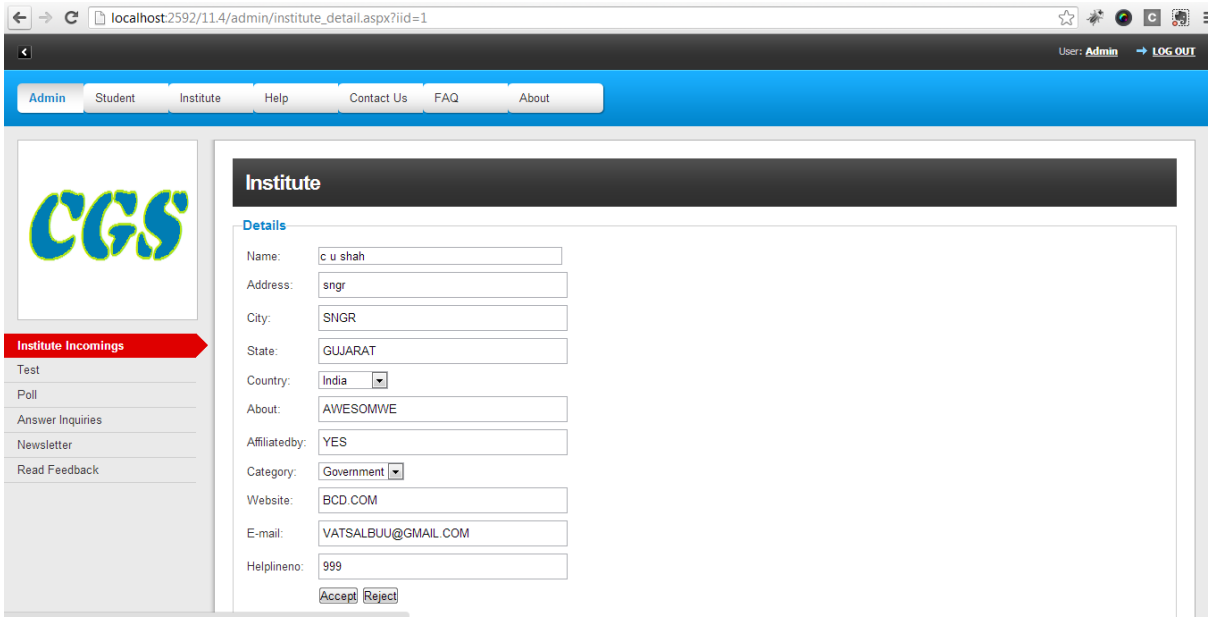


Fig 8.4: Admin Institute Details

For the approval of institute this page displays the details of the institute to the admin.

- **ADMIN TEST**

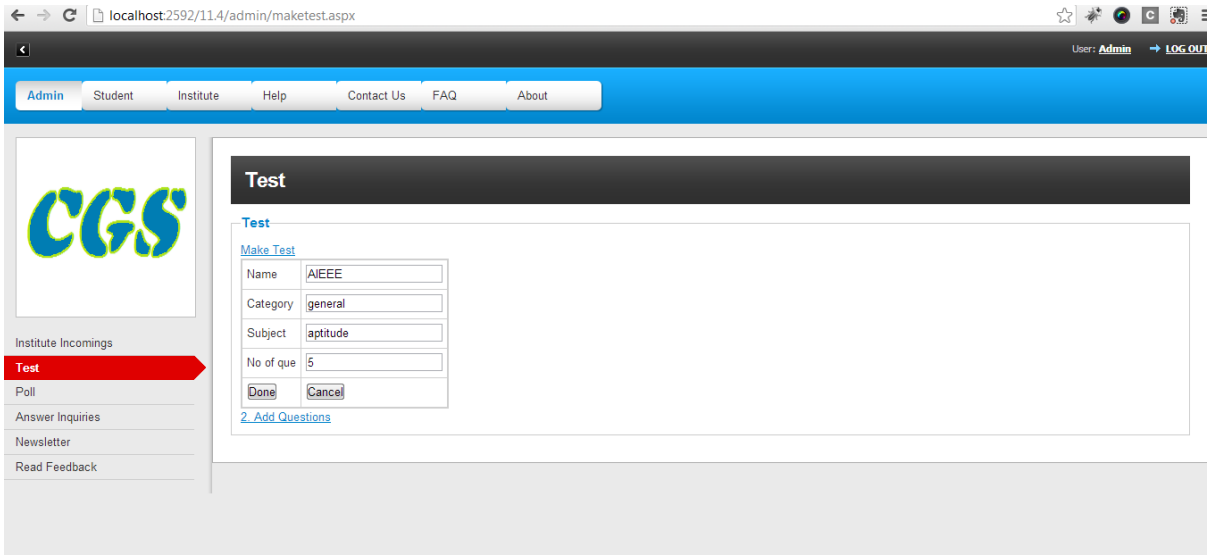


Fig 8.5: Admin Test

The page provides admin the facility of adding a test in the system that the student can appear.

- **ADMIN ADD POLL**

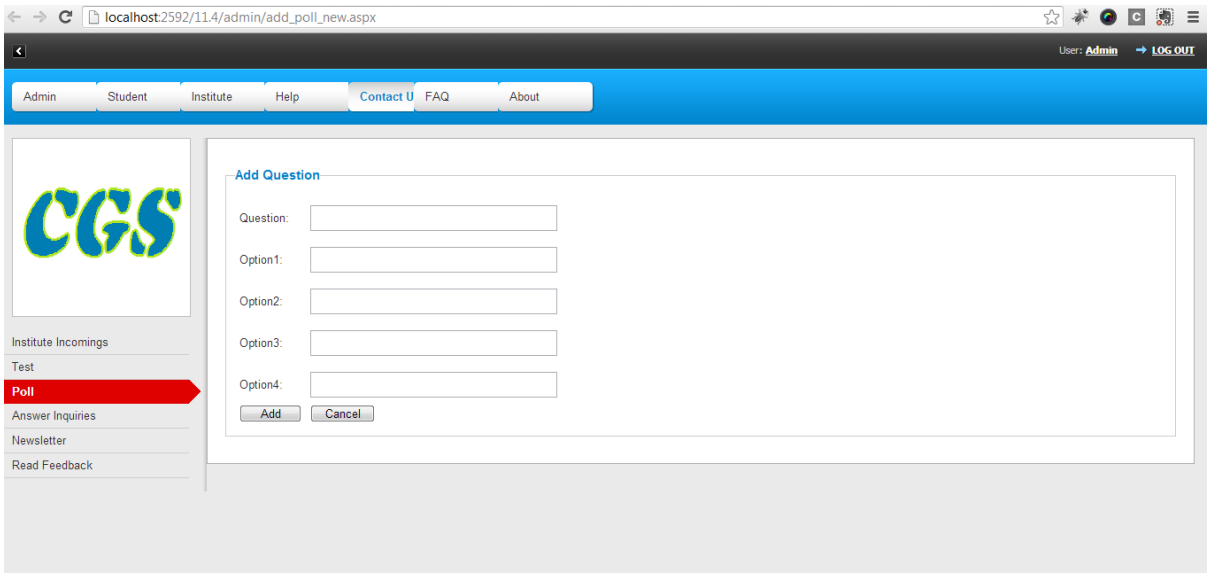


Fig 8.6: Admin Add Poll

This page lets the admin add a poll module that is to be answered by the student.

- **ADMIN DELETE POLL**

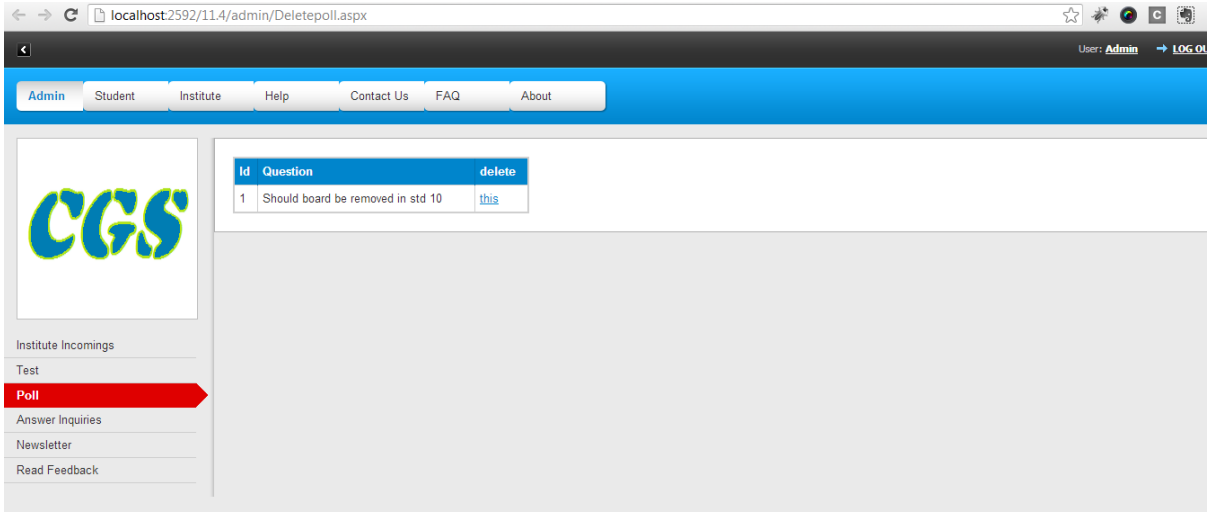


Fig 8.7: Admin Delete Poll

This page lets the admin delete a poll that is currently displayed for the student.

- **ADMIN NEWSLETTER**

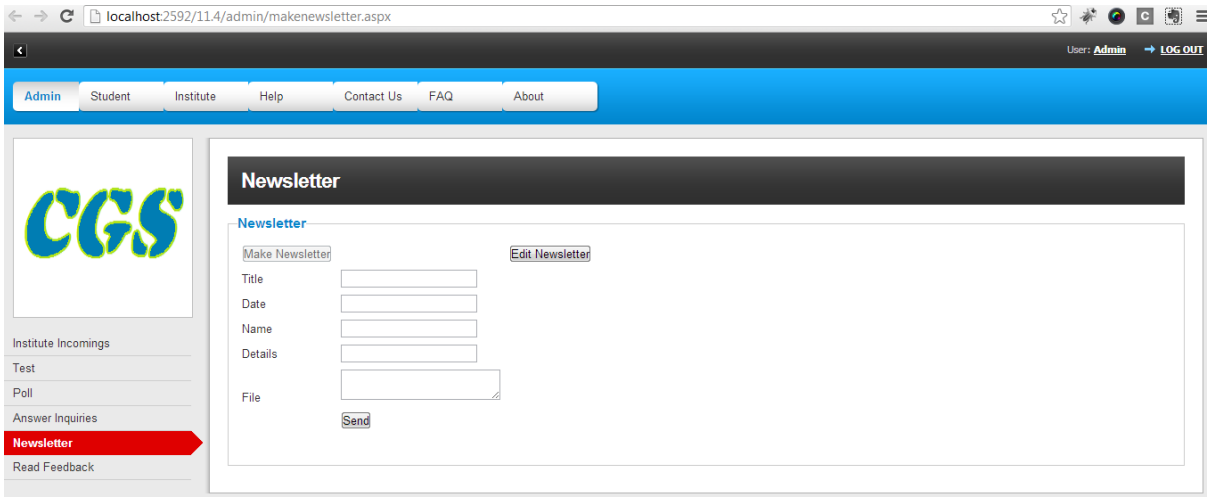


Fig 8.8: Admin Newsletter

Description: This page makes a newsletter that is to be sent to the specific student that have requested for it by E-mail.

- **ADMIN FEEDBACK**

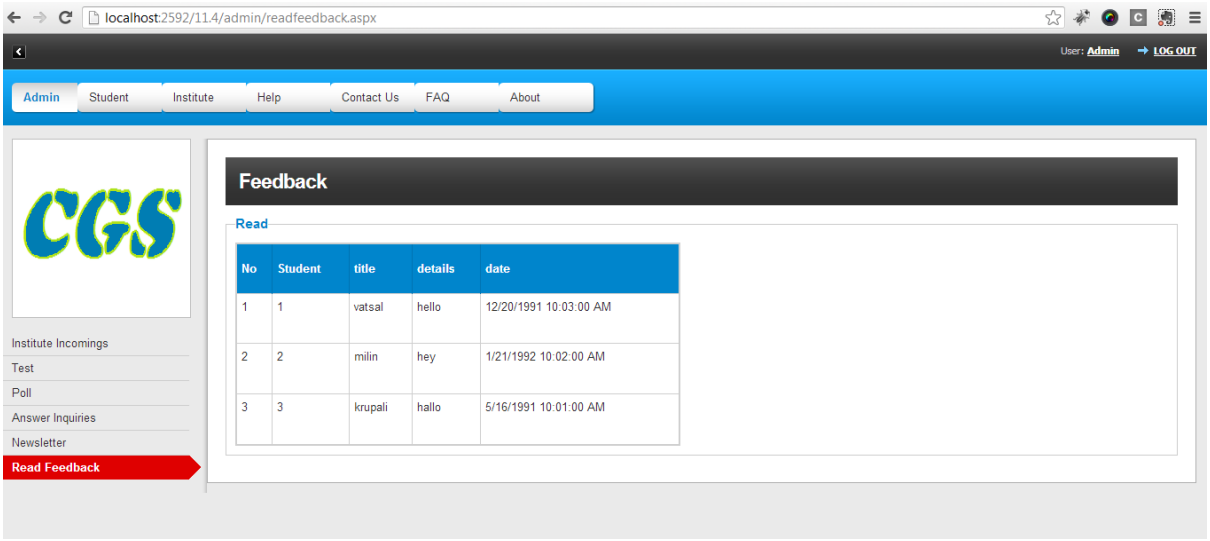


Fig 8.9: Admin Feedback

This page lets the admin read all the feedback provided by the students.

- **ADMIN FAQ**

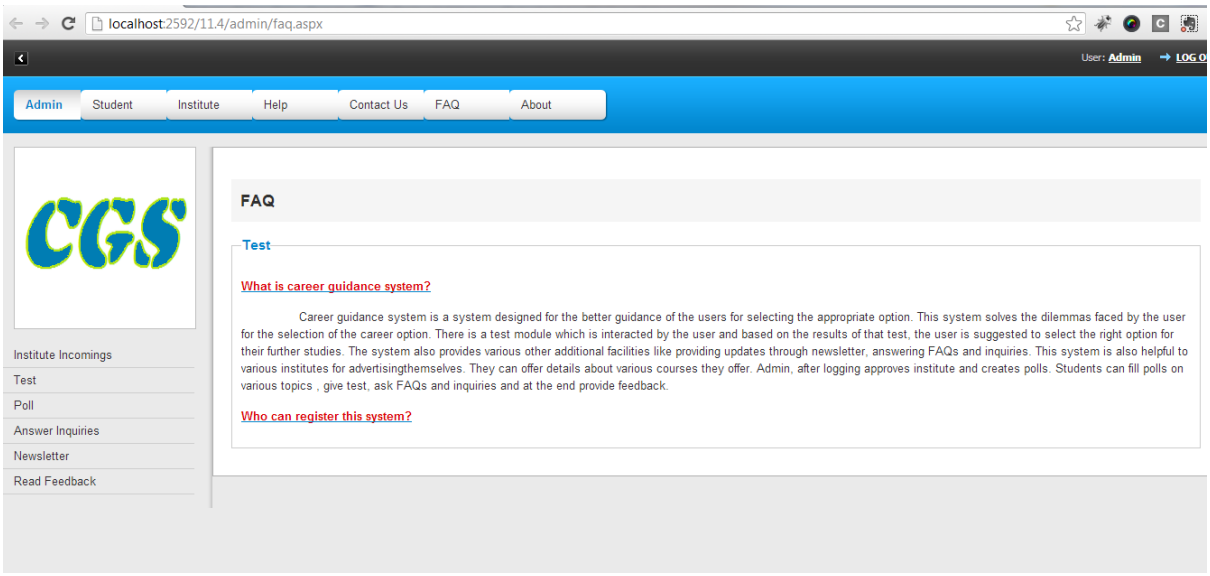


Fig 8.10: Admin FAQ

This page covers all the Frequently Asked Questions by the student and institute.

- **ADMIN PROVIDE EXPERTS**

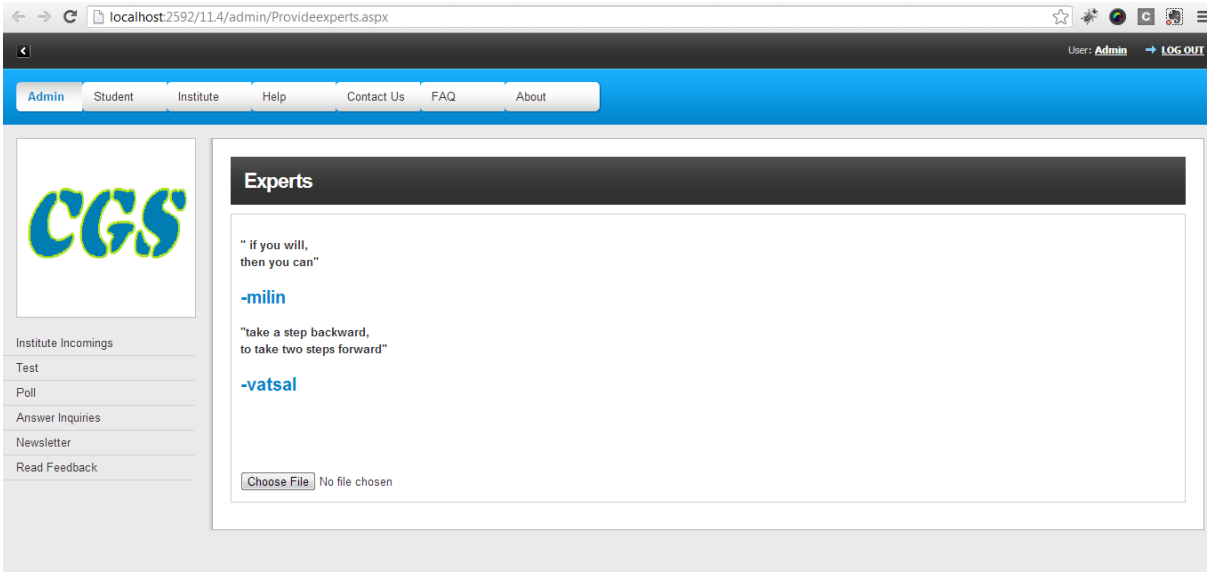


Fig 8.11: Admin Provide Experts

This page provides the expert's opinion that is provided by the admin.

8.2 INSTITUTE

- **INSTITUTE LOGIN**

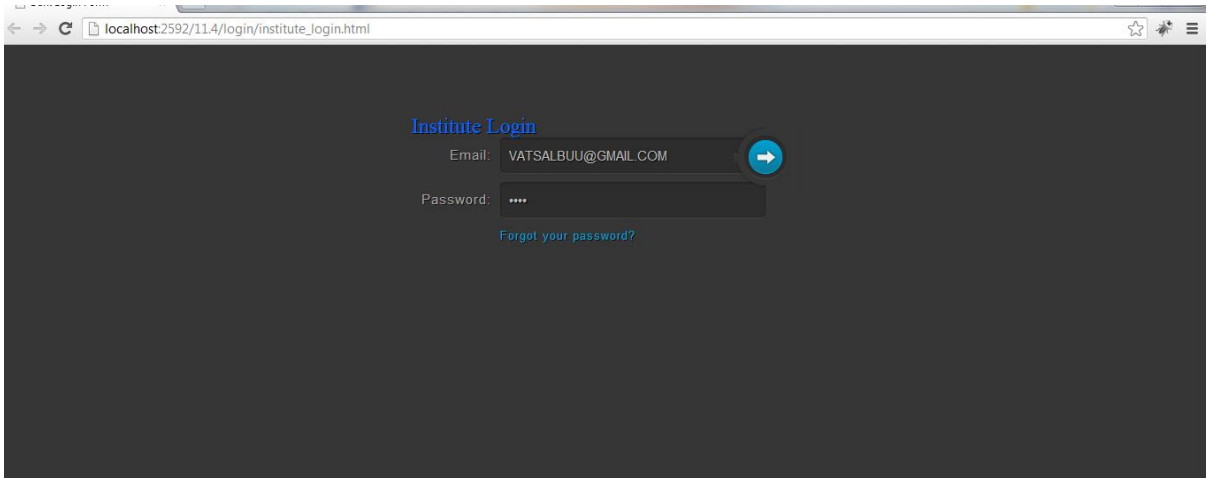


Fig 8.12: Institute Login

This page lets the institute login into our system.

- **INSTITUTE REGISTER**

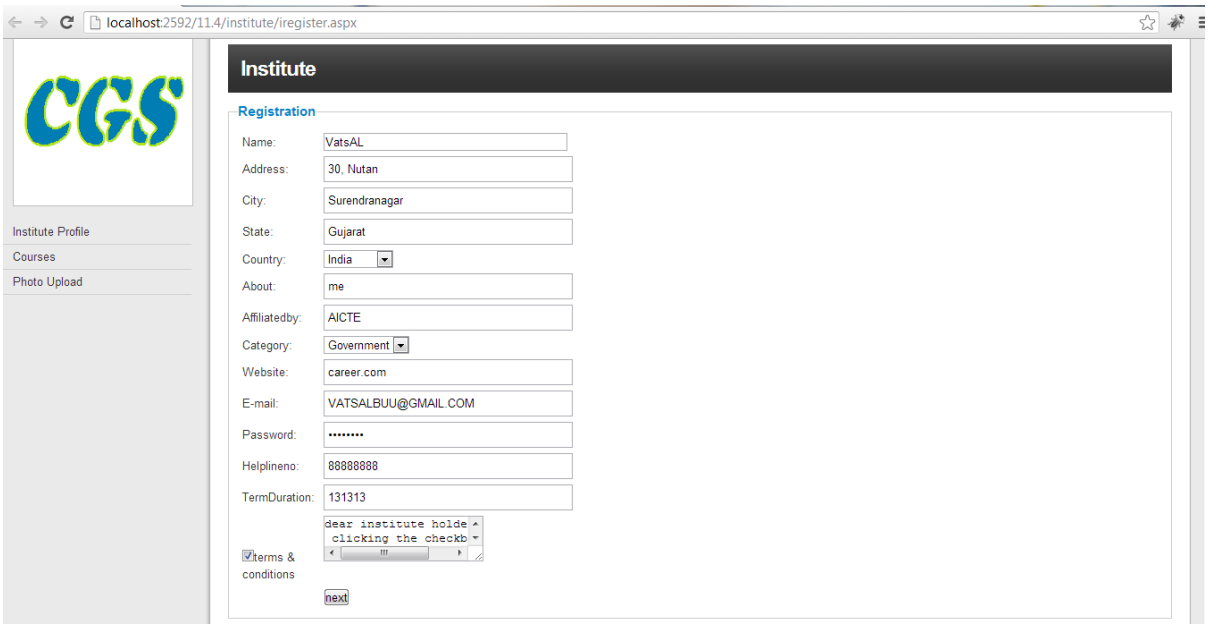


Fig 8.13: Institute Register

From this page the institute can register into our website. Only after the approval of the admin, the institute can login into our system.

- **INSTITUTE HOME**

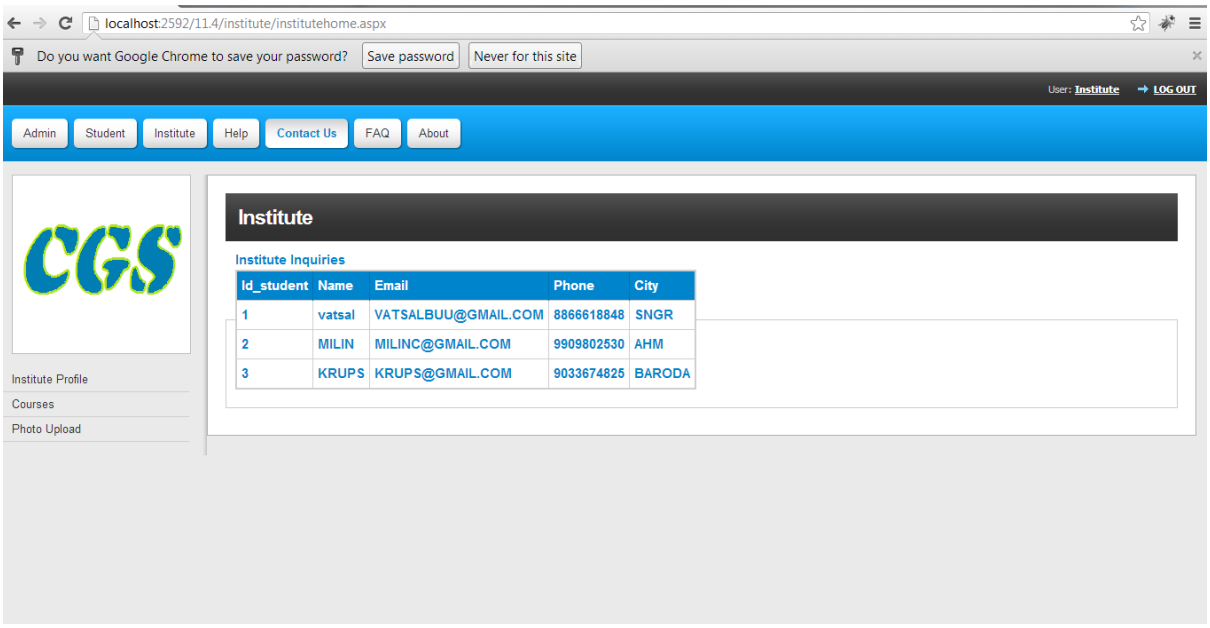


Fig 8.14: Institute Home

This is the home page for the institute. It displays the list of students that have inquired for them.

- **INSTITUTE PROFILE**

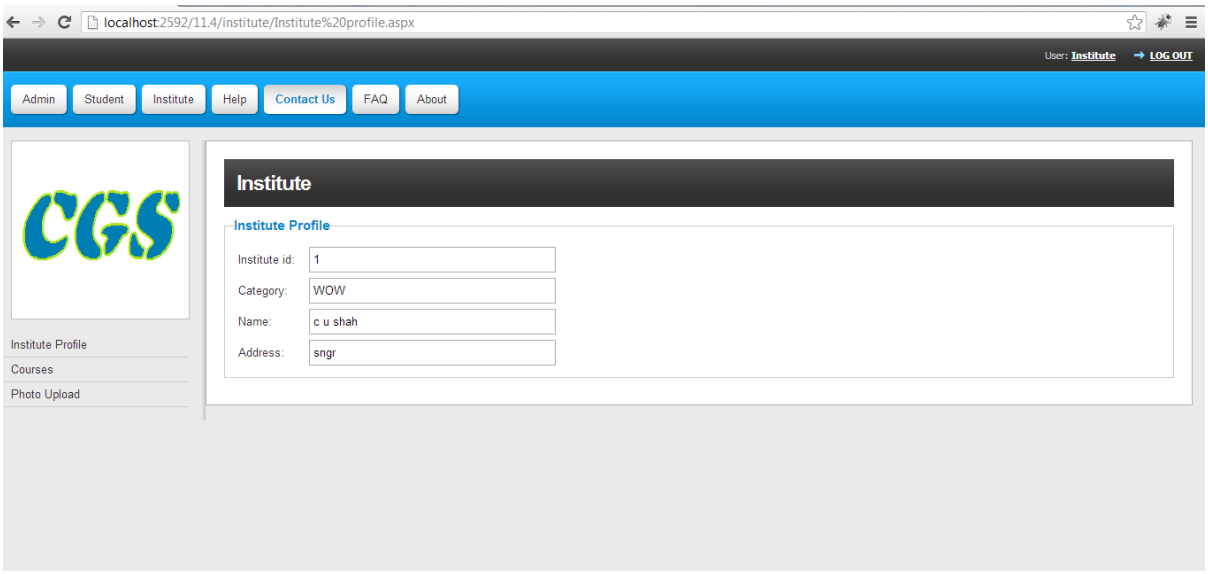


Fig 8.15: Institute Profile

This page displays the institute profile.

- **INSTITUTE COURSES**

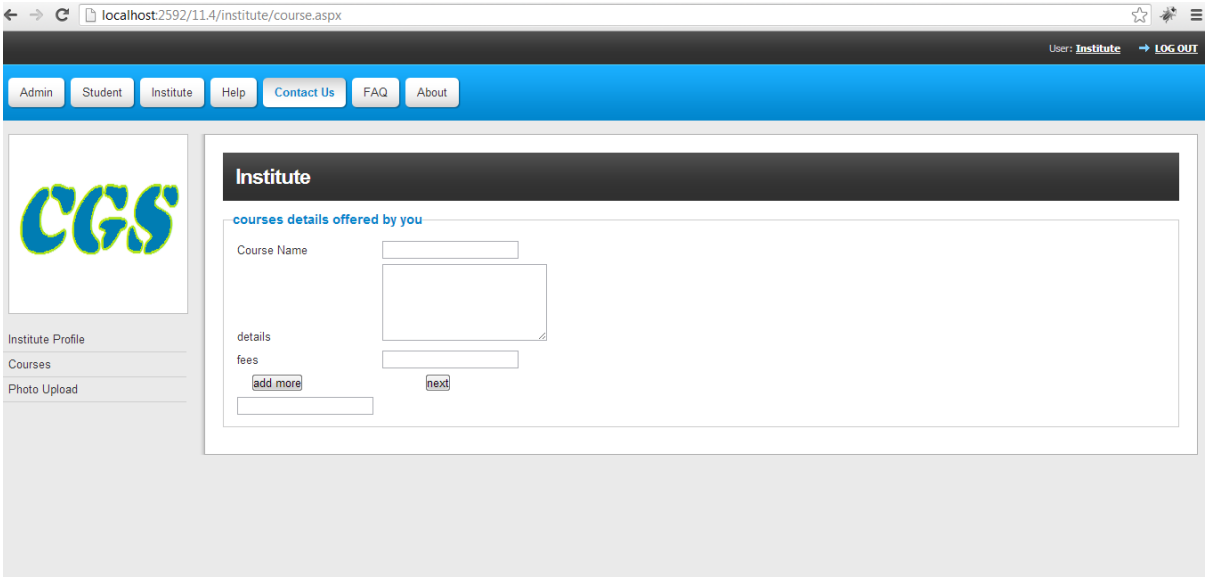


Fig 8.16: Institute Courses

This page lets the institute add the courses that are provided by them.

- **INSTITUTE COURSE VIEW**

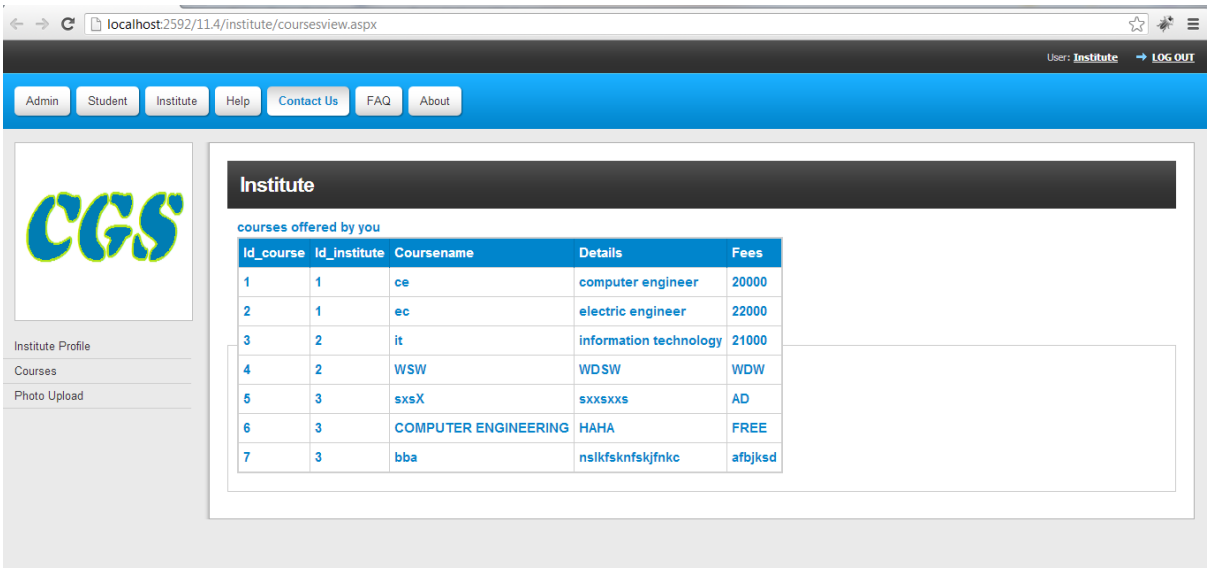


Fig 8.17: Institute Course View

This page displays the list of courses provided by the institute.

- **INSTITUTE UPLOAD PHOTO**

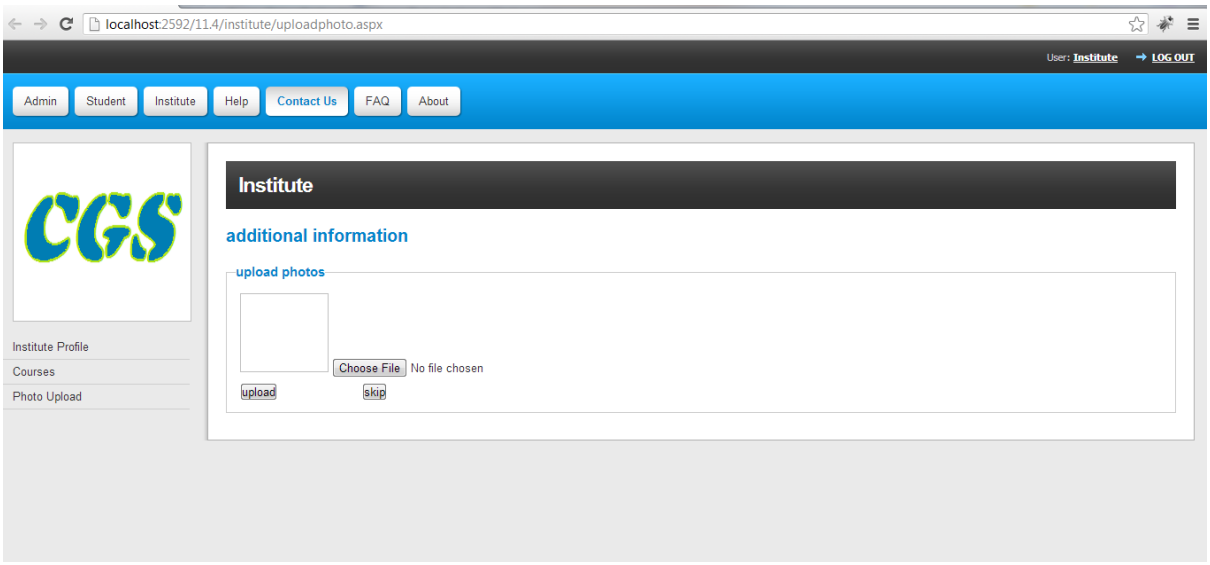


Fig 8.18: Institute Upload Photo

This page lets the institute upload images for their image gallery.

- **INSTITUTE FAQ**

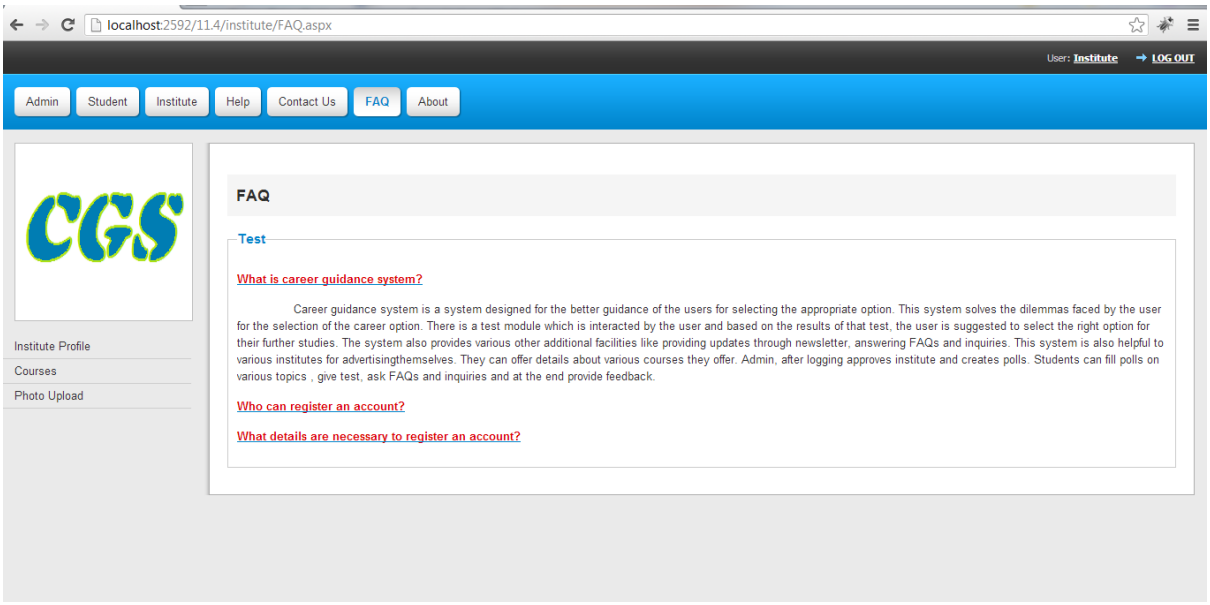


Fig 8.19: Institute FAQ

This page contains the Frequently Asked Questions by the different institutes.

- **INSTITUTE CONTACT**

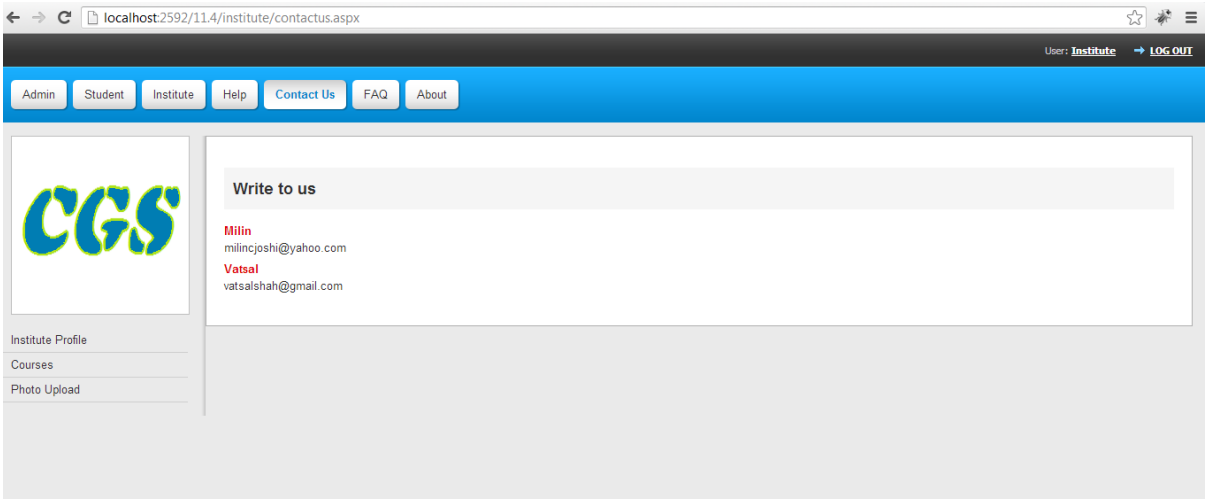


Fig 8.20: Institute Contact

This page provides the details for contacting the admin.

8.3 STUDENT

- **STUDENT REGISTER**

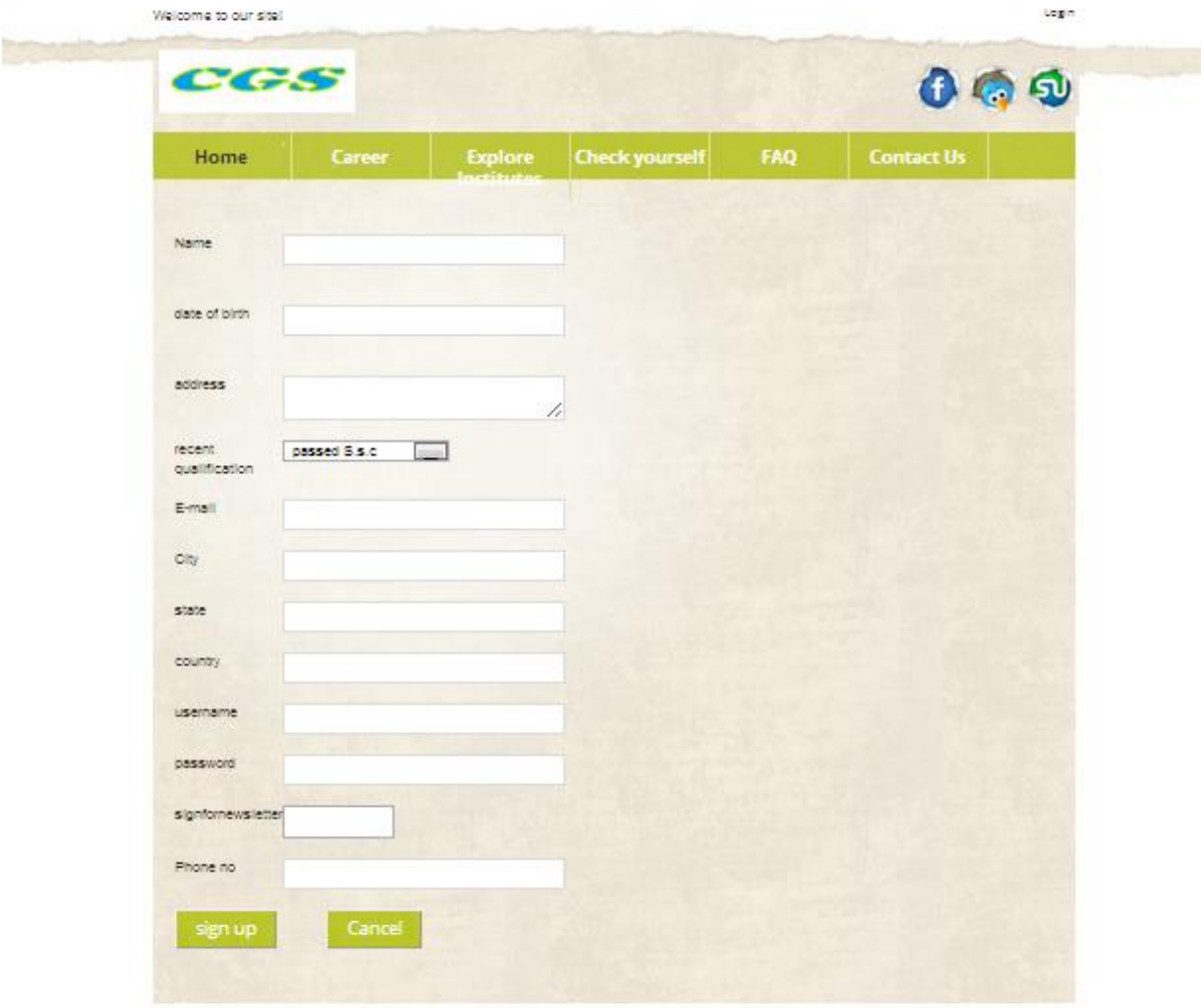


Fig 8.21: Student Register

This page lets the student register into our system by providing his/her details.

- **STUDENT LOGIN**

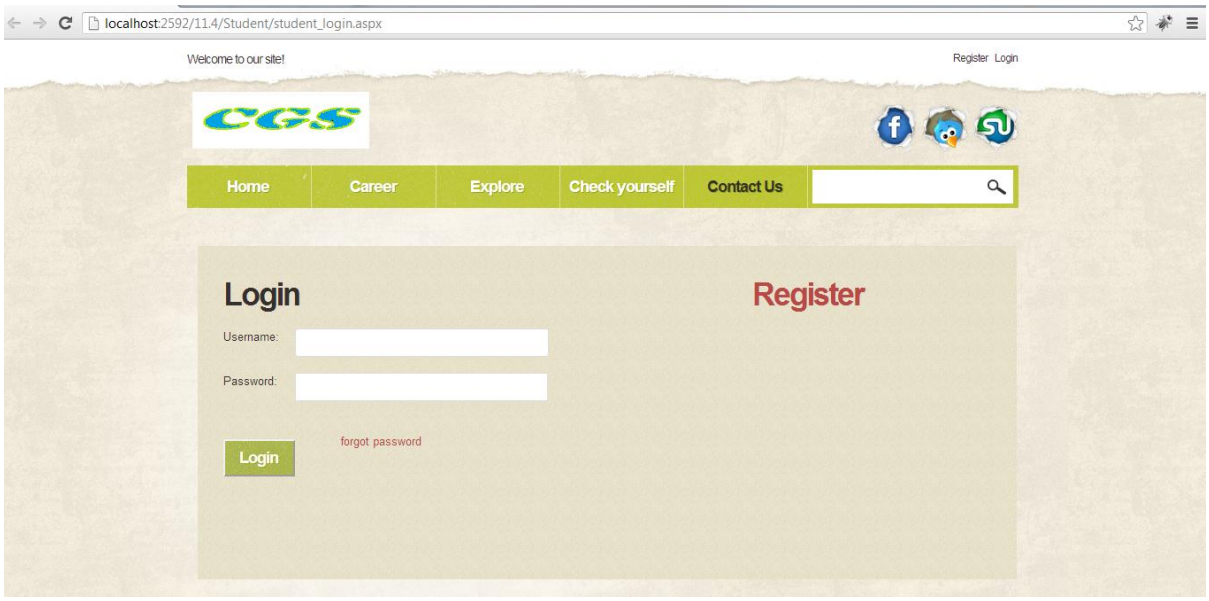


Fig 8.22: Student Login

By this page the student can login into our system by providing their username and password.

- **STUDENT HOME**

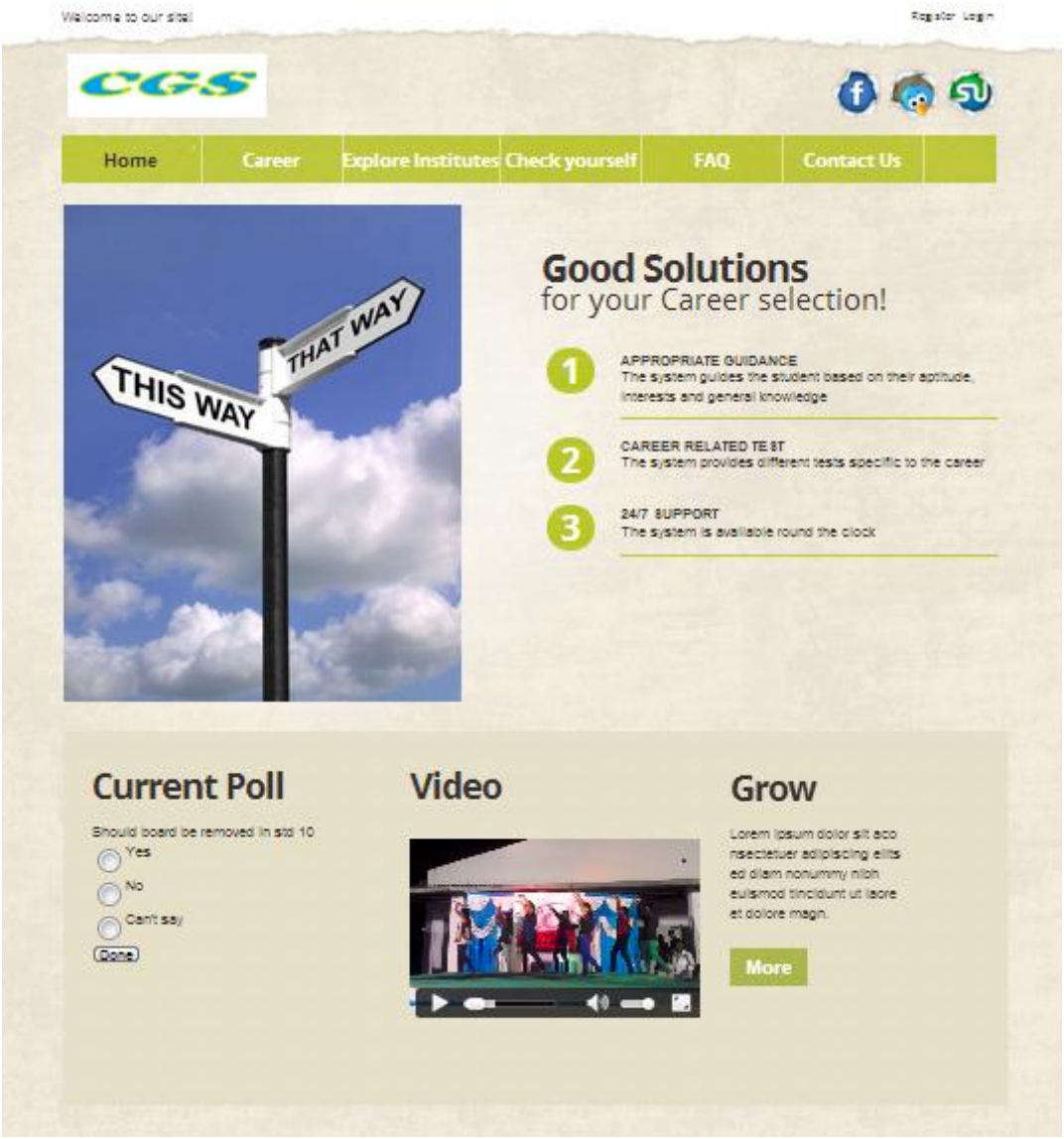


Fig 8.23: Student Home

After the student has logged in, this page is displayed to the student. It contains different activities that can be performed for career guidance.

- **STUDENT CHOOSE TEST**

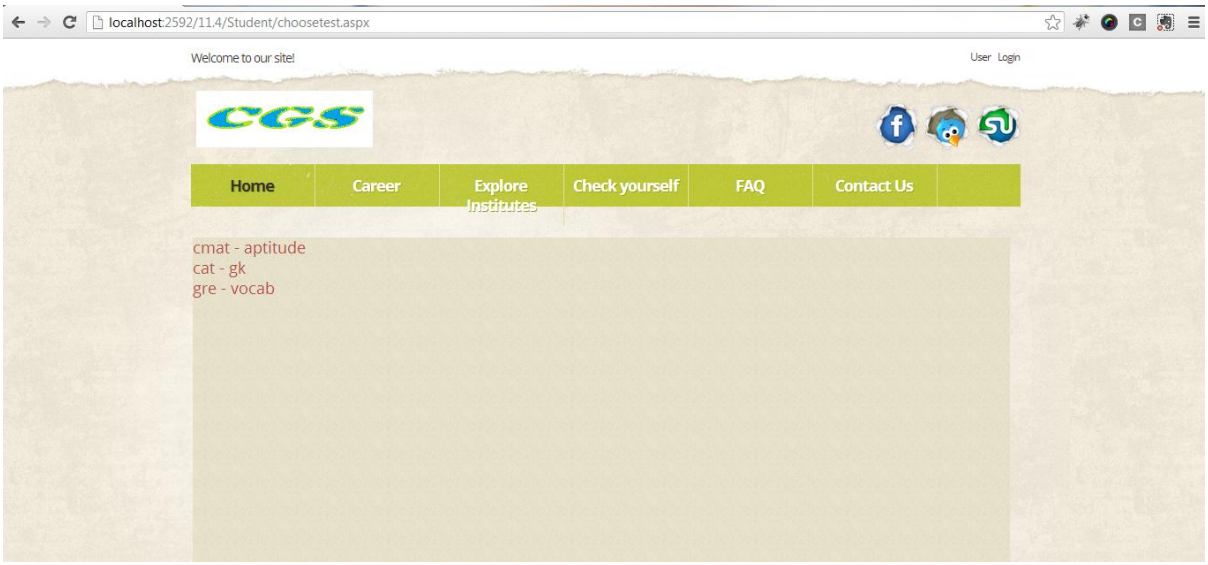


Fig 8.24: Student Choose Test

This page lets the student choose the test that they want to appear for.

- **STUDENT CHECK**

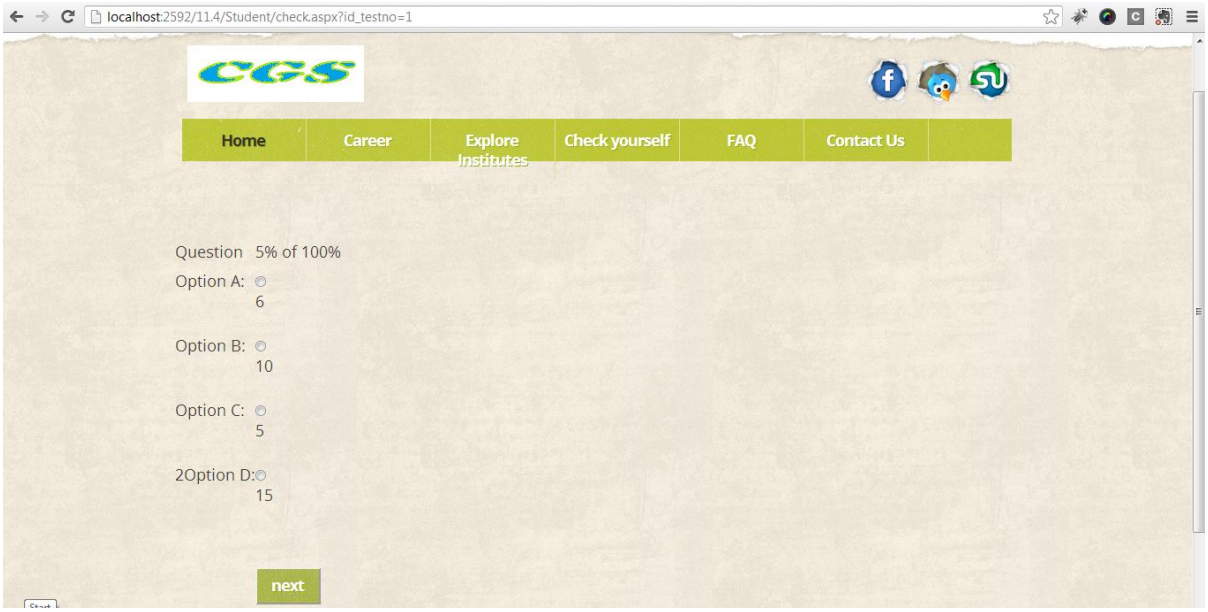


Fig 8.25: Student Check

This page contains the questions and their options for the tests. The student can choose any one option and proceed to the next one.

- **STUDENT RESULT**

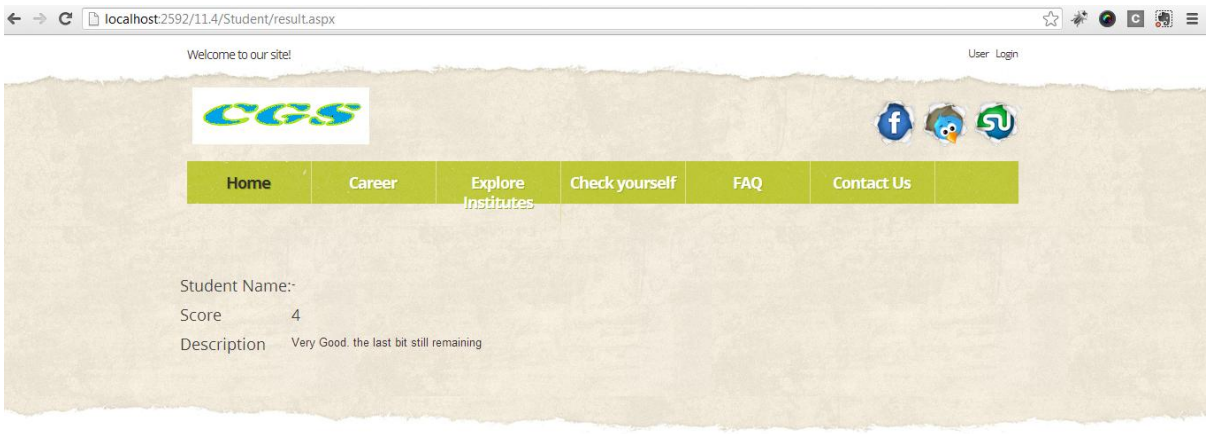


Fig 8.26: Student Result

This page displays the result of the test that the student has attempted. it contains the score of the student and description for choosing their career.

- **STUDENT EXPLORE INDIAN INSTITUTE**

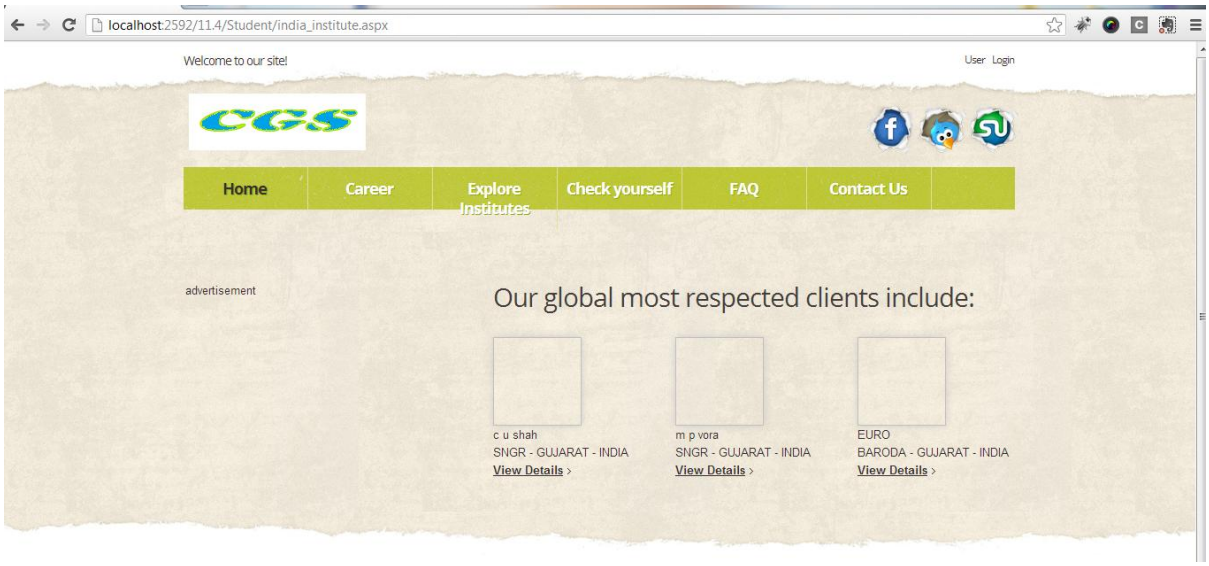


Fig 8.27: Student Explore Indian Institute

This page lets the student explore all the Indian institutes registered in our system.

- **STUDENT EXPLORE ABROAD INSTITUTES**

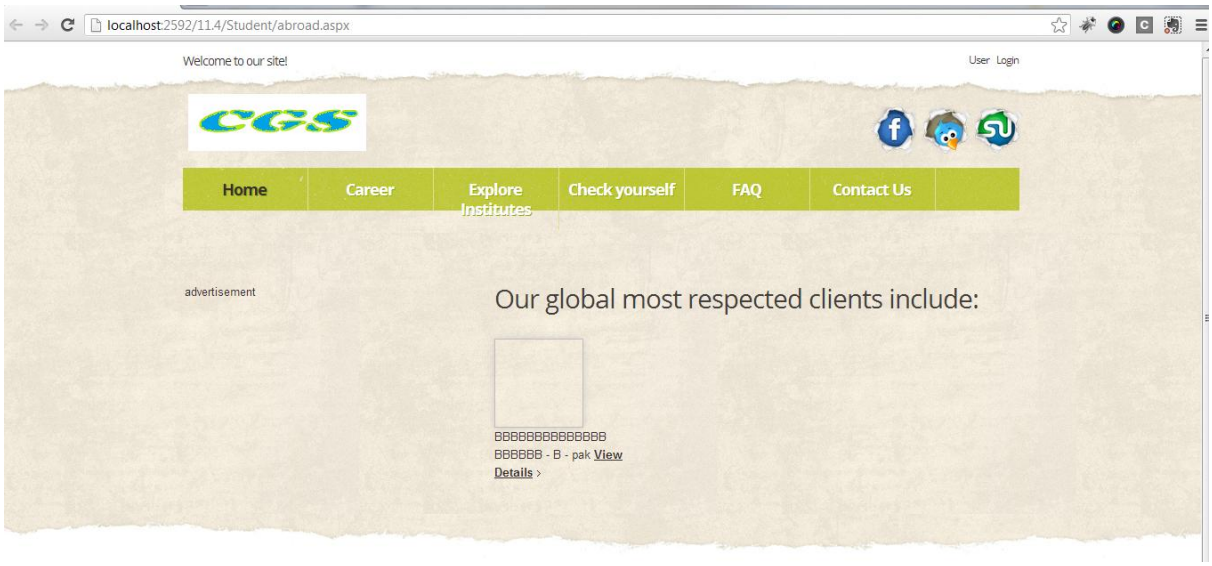


Fig 8.28: Student Explore Abroad Institutes

This page lets the student explore all the non-Indian institutes registered in our system.

- **STUDENT FORGOT PASSWORD**

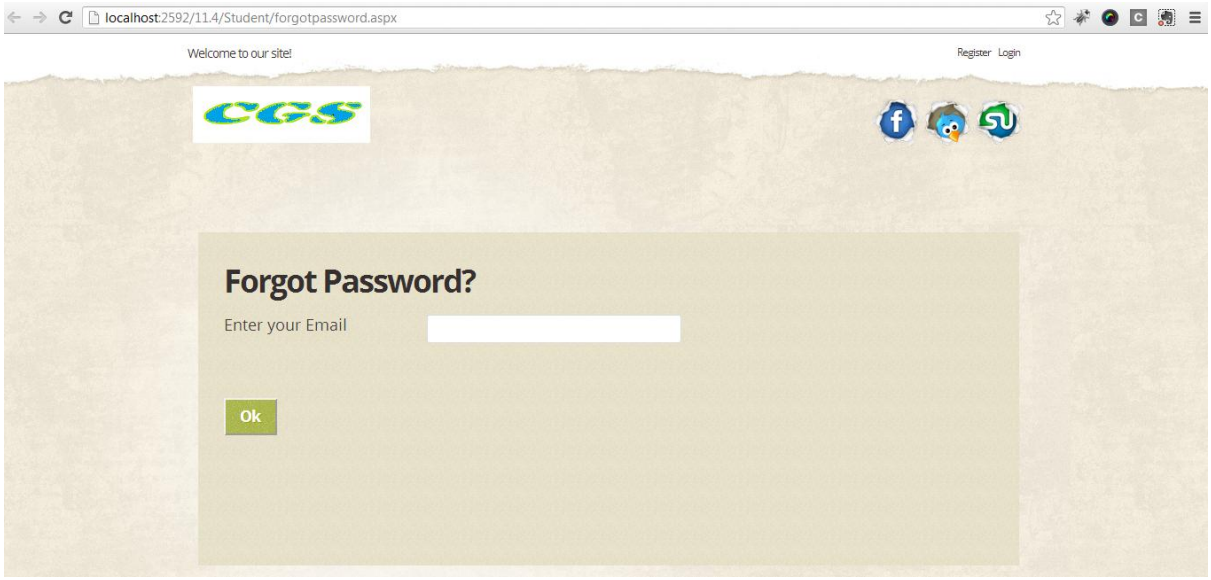


Fig 8.29: Student Forgot Password

If the student has forgot his/her password, this page takes the E-mail of the student and sends him/her the password through E-mail.

- **STUDENT CAREER**

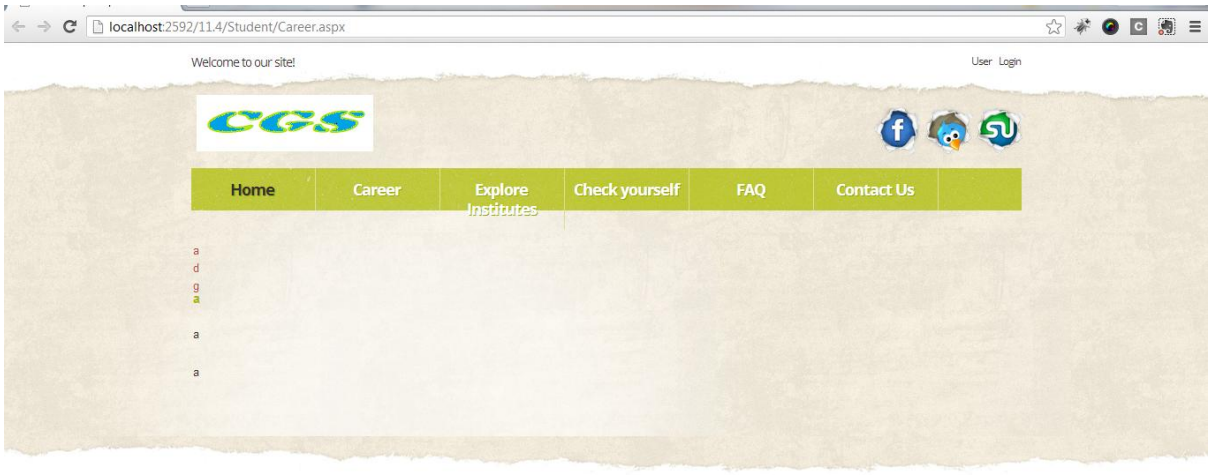


Fig 8.30: Student Career

This page displays the details about different courses according to their category.

- **STUDENT CONTACT**

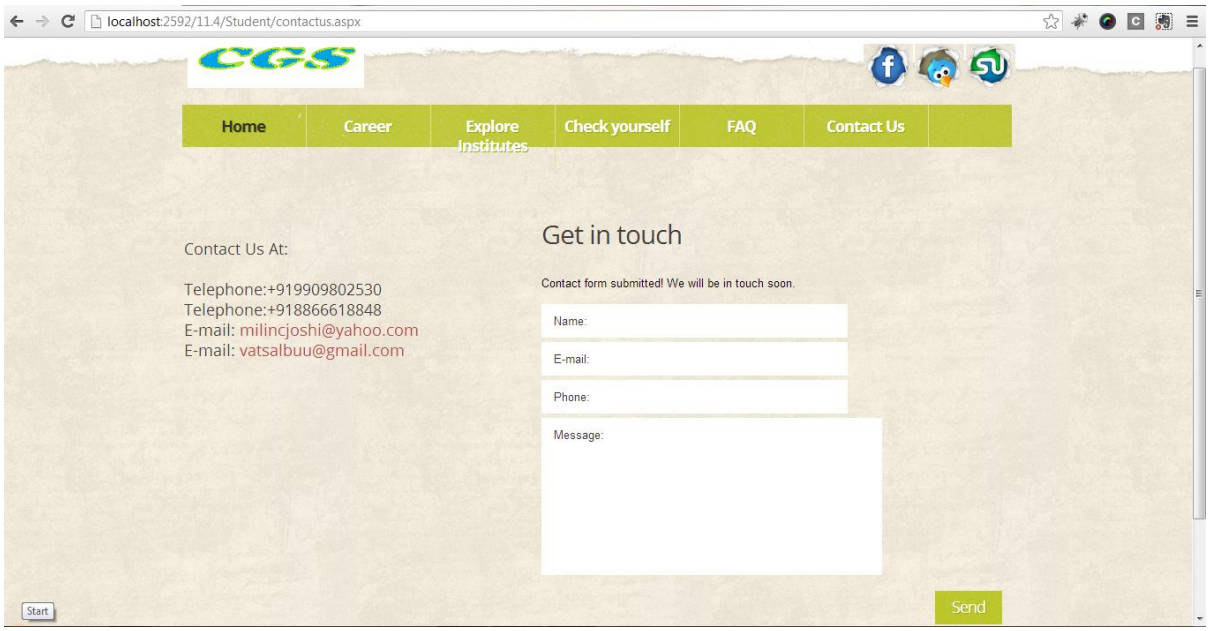


Fig 8.31: Student Contact

Through this page the student can contact the admin for various inquiries.

- **STUDENT FAQ**

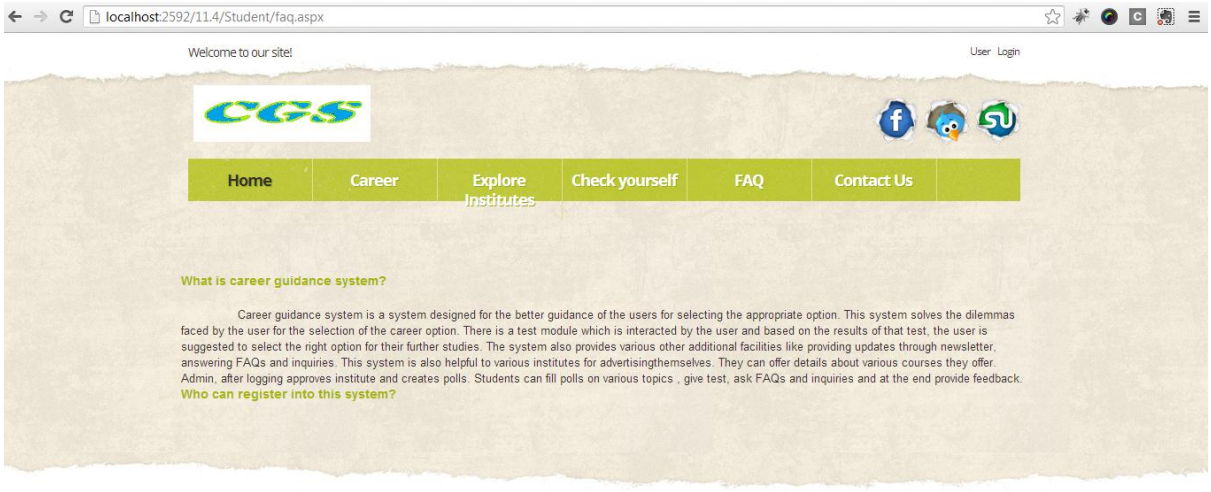


Fig 8.32: Student FAQ

This page contains all the Frequently Asked Questions by the students.

9.0 LIMITATIONS AND FUTURE ENHANCEMENTS

9.1 LIMITATIONS

Career guidance system offers guidance to students for S.S.C, H.S.C and Bachelors. Thus system is limited to the students for above education span only. Furthermore the system does not provide guidance for all available education field. Also it provides information about institutes up to certain level only. The system also does not provide any reference materials needed for exam preparation. The institute can only obtain a list of students that have inquired about them but it cannot contact the students directly through our website. The system does not provide any kind of instant messaging facility.

9.2 FUTURE ENHANCEMENTS

Currently the system is provided free of cost to all the users, future enhancements may include paid registration in the system. It can also include chat facility between admin and student. Future enhancements can increase scope of education fields like guidance for Masters, PhDs, etc. Future modules can consist of discussion module for students. Future scopes also include a specific design for mobile devices like cell phones and tablets. Future student facilities includes of profile pages for student user and a record of all the tests attempted.

10.0 CONCLUSION

Thus from this we conclude that E-Catalogue provides a cheaper and faster way to manage and modify catalogs. It provides a better option for manufacturer instead of printing manual catalogs which reduces paper work and cost. At the same time wholesalers and consumers will get the advantage of receiving the updates about new product. In this way, E-catalogue is a practical approach for designing and displaying the catalogs.

REFERENCES

www.jagranjosh.com

www.ebscohost.com

www.careerdirectonline.org

www.careerdots.com

www.google.com

www.msdn.microsoft.com

www.asp.net

www.aspfree.com