



# AAVISHKAR INSTITUTE

DEVELOPED BY: TANU SHAMRA

## CHAPTER:1

### Introduction & Object of Project

#### Introduction

This project is specially designed for Students, because in this project students can take online Admission from Aavishkar Institute from their home which is registered in this project. Students can apply for admission through our project “Aavishkar Institute”. As well as students can also get online material.

The “Aavishkar Institute” project is a multi-user system. It has been developed in a way that allows user to perform the function smoothly and with proper accuracy.

The system is developed in ASP.NET and some other software use like SQL Server, and also used Java Script and CSS for designing purpose.

## How It Works?

In this project first of all a homepage is displayed for all visitors as well as registered students/users.

If user wants to take admission, then they have to Login first. If the user is not a Registered User, he/she will have to click on the Signup Button and fill all the fields in the registration form.

After registering you can see login page. Users will have to login with their registered mobile number and password.

After login, user will see User-Panel of ‘Aavishkar Institute’ Website. User can simply apply for admission by filling all the required details.

## CHAPTER:2

### Tools or Platform

#### Requirement Details

➤ Software:

- SQL Server
- Any browser like Google Chrome, Internet Explorer, etc.

➤ Hardware:

- Processor: Intel Core i3
- RAM: 8.00 GB
- Hard Disk: 1.00 TB

➤ Tools Used:

- Front End: HTML, CSS, JavaScript
- Back End: ASP.NET
- Operating System: Windows 8 or Higher

## CHAPTER:3

### System Analysis

3.1 Identification of Needs

3.2 Preliminary Investigation

3.3 Feasibility Study

3.4 Project Planning

3.5 Project Scheduling (PERT Chart and GANTT Chart)

3.6 Software Requirement Specification

3.7 Data Models

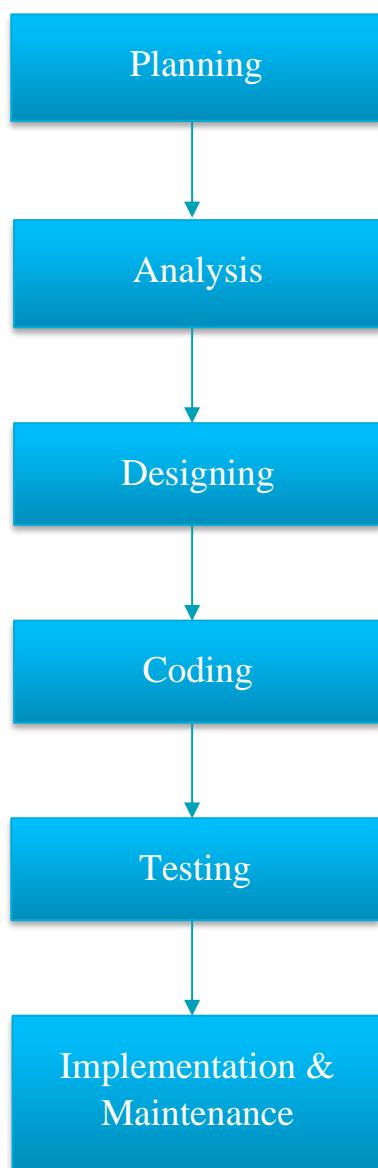
3.8 Normalization

#### 3.1 Identification of Needs

The viewer of our system does not need to login as long as User wants to browse information about Aavishkar Institute only. But if the User wants to apply for admission in Aavishkar Institute, then he/she must be registered to our website.

User needs to provide name, email, mobile number and password when creating an account.

## SDLC Model



## SDLC Steps

### ➤ Step 1: Requirement Gathering

In this phase of SDLC necessary information are collected. We collect the information in this phase through questionnaire and online websites also. We also note down the requirements of the User.

### ➤ Step 2: Requirement Specification & Analysis

The requirements are being specified in this phase. Analyst checks how the current system is working and then plans how to develop the proposed system and implement it.

### ➤ Step 3: Design

In this phase the design of both front-end and back-end is decided. In this phase we use HTML, CSS for designing, and designing of tables are also done properly in this phase.

## ➤ Step 4: Coding

After the completion of designing there is a turn for coding. Coding is completed in various files. In most of the cases coding takes much more time than designing.

## ➤ Step 5: Testing

In testing phase first unit level testing is done. In unit level testing the software is tested in an individual file. After the unit testing of each file, the integration testing done. In integration testing the files are tested into their own package. At last, in the system testing the whole system is run on the Internet Environment.

## ➤ Step 6: Maintenance

The proper maintenance is given as indicated in the SRS (Software Requirement Specification). It is a responsibility of Developer to maintain the software during its criteria mentioned in SRS document.

### 3.2 Preliminary Investigation

#### ➤ Our Objectives

- Admin can Add, Delete and Edit records.
- User can learn about various courses.
- User can handle all type of information easily.
- User can contact us at any time, for any query.
- Easy updating, time saving, quick process.
- User can update their sign-up details.
- User can simply apply for admission by filling the admission form.
- Students can get online material.

## ➤ Platform Specification

I have used windows based developing & graphical based applications; our software will provide support for GUI efficiently. It is must for our system because for our project.

It is very useful and supportive for ASP.NET and Java Script to control the components. There is use of CSS for menu creation and display web page in it. It's easy to use integrity database report with Data Report.

ASP.NET platform is mostly use for developing web-based application, regular version of operating system is used with it. Always need the back-end tool for developing web application with front-end tool.

When starting a new business, organizations have some requirement for it. There are many reasons for starting a new business or modify the old system.

Business requirement set depends on the top-level organization i.e., manager, administrator etc.

### 3.3 Feasibility Study

A feasibility analysis usually involves a thorough assessment of the operational(need), financial and technical aspects of a proposal.

Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints.

A feasibility study should be relatively cheap and done at the earliest possible time.

Depending on the study, the decision is made whether to go ahead with a more detailed analysis.

When a new project is proposed, it normally goes through feasibility assessment.

Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration.

Facts considered in the feasibility analysis are:

## 1. Technical Feasibility

A technical feasibility study for a website project like Aavishkar Institute, which allows users to take online admission, covers the following aspects:

- i. **Technical Requirements:** The technical requirements of the website are ASP.NET web framework that uses C#, and SQL Server database is used to build the website.
- ii. **User Interface:** The user interface is a crucial component of any website, particularly for online admission sites. Using CSS and JavaScript in the user interface of our website has helped us create an engaging and visually appealing user experience.
- iii. **Security:** Security is another critical aspect of any website that handles sensitive information such as personal and financial details of users. The information we take from user will never be shared to third party. The user credentials will be securely stored and authenticated when logging into the website, as well as how user roles and permissions will be managed.
- iv. **Performance:** The website can handle a large number of concurrent users and traffic spikes without affecting its performance. It covers the scalability and performance metrics required to meet the expected user traffic.
- v. **Compatibility:** The website is compatible with various devices, including laptops, tablets, and mobile devices.

- vi. **Maintenance:** Maintenance is the very important part of any website. The website will require a system and software upgrade within specific period of time to ensure it remains up to date with the latest technology and security standards.
- vii. **Integration:** If the website needs to integrate with other systems, such as payment gateways or learning management systems, the integration will be designed to ensure the security of user data and prevent any potential data breaches.

Hence, the proposed system is technically feasible.

## 2. Economical Feasibility Study

Economic justification is generally the “Bottom Line” consideration for most systems. It includes a broad range of concerns that includes cost benefit analysis.

The study takes into account the costs associated with the project, as well as the potential benefits and returns that the project may generate. Here are some key elements that can be included in an economic feasibility study:

### 1) Cost Analysis:

- **Capital Costs:** The cost of equipment, software, and other assets required to offer the courses, such as computers, online learning platforms, or classroom equipment.
- **Operating Costs:** The cost of ongoing expenses such as salaries for teachers and staff, utilities, maintenance, and marketing.
- **Overhead Costs:** The cost of indirect expenses such as rent, insurance, and taxes. The cost of hardware and software for the class of application being considered.

### 2) Revenue Analysis:

- **Course Fees:** The tuition fees charged to students for attending the courses.
- **Enrollment Projections:** The estimated number of students that will enroll in the courses over a given period of time.
- **Revenue Projections:** The estimated revenue generated from course fees based on the enrollment projections.

### 3) Profitability Analysis:

- **Break-even Analysis:** The analysis of how many students need to enroll in the courses for the project to break even financially.
- **Profit Margin Analysis:** The analysis of the profit margin on each course, which is calculated by subtracting the cost of offering the course from the revenue generated by course fees.

### 4) Sensitivity Analysis:

- **Risk Assessment:** Risk assessment is an important component of the economic feasibility study for the Aavishkar Institute website project. The primary risks identified include fluctuations in enrollment, unexpected expenses such as server downtime, and competition from other coaching institutes. These risks have been assessed for their likelihood and impact on the project's financial performance, and contingency plans have been developed to mitigate their potential impact.
- **Scenario analysis:** Scenario analysis has also been conducted to evaluate the project's financial viability under different assumptions and circumstances. Based on varying levels of enrollment and course fees. The results of the scenario analysis indicate that the project is financially feasible, although the level of profitability varies. This analysis has helped to identify potential risks and opportunities and ensure that the project is economically sustainable in different circumstances.

This feasibility checks whether the system can be developed with the available funds. This can be done economically if planned judicially, so it is economically feasible. The cost of project depends upon the number of man hours required.

### 3. Operational Feasibility Study

It is mainly related to human organizations and political aspects. The points to be considered are:

- i. **Organizational Structure:** The current organizational structure of Aavishkar Institute can support the development and maintenance of the proposed website. If necessary, I would consider changes to the organizational structure to ensure that it supports the project effectively.
- ii. **Resources:** The availability of resources such as personnel, equipment, and facilities are sufficient to support the development and maintenance of the website. If necessary, I would recommend changes or additional resources.
- iii. **Legal feasibility:** The proposed website complies with relevant laws, regulations, and policies. The potential legal and regulatory risks and requirements are associated with the website.

The system is operationally feasible as it is very easy for the end users to operate it. It only needs basic information about Windows platform.

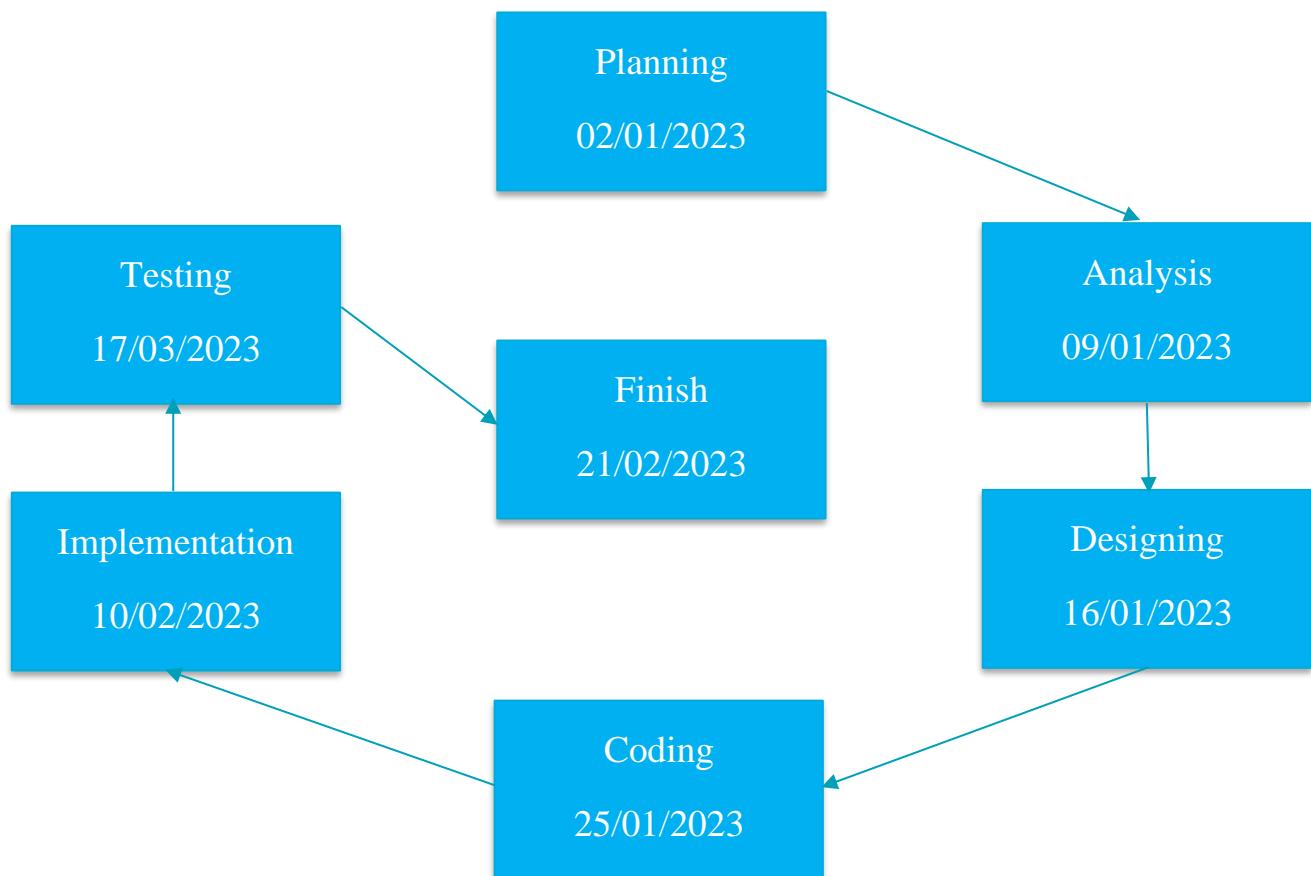
### 3.4 Project Planning

No.	Task Name	Start	Finish	Duration
1	Planning	02/01/2023	08/01/2023	1 Week
2	Analysis	09/01/2023	15/01/2023	1 Week
3	Designing	16/01/2023	24/01/2023	1.2 Week
4	Coding	25/01/2023	09/02/2023	2.2 Week
5	Implementation	10/02/2023	16/02/2023	1 Week
6	Testing	17/02/2023	21/02/2023	5 Days
<u>Total = 50 Days</u>				

### 3.5 Project Scheduling

#### ➤ Pert Chart

This chart represents the development of our system date wise.



## ➤ Gantt Chart

This is a graphical representation of the date wise development of our project system. It is very similar to the PERT chart except that it is represented graphically.

ID	Task Name	Gantt Chart from January to February
1	Planning	02 Jan – 08 Jan
2	Analysis	09 Jan - 15 Jan
3	Designing	16 Jan – 24 Jan
4	Coding	25 Jan - 09 Jan
5	Implementation	10 Feb - 16 Feb
6	Testing	17 Feb - 21 Feb

### 3.6 Software Requirement

#### ➤ ASP.NET

ASP.NET is a popular web application framework developed by Microsoft. It allows developers to build dynamic web applications and services using a variety of programming languages, including C#, Visual Basic, and F#.

ASP.NET offers a range of features that make it an ideal choice for developing web applications, including built-in security features, a rich set of controls and libraries, and support for a wide range of databases.

Some of the key components of ASP.NET include the .NET Framework, which provides a common language runtime and a rich set of libraries for building applications, and the ASP.NET runtime, which provides a set of services for handling requests and managing application state.

In addition, ASP.NET includes a range of tools for developing, debugging, and deploying applications, including Visual Studio, a powerful integrated development environment (IDE), and various command-line tools.

ASP.NET has better language support, a large set of new controls, XML based Components and better Student authentication.

ASP.NET provides increased performance by running compiled code.

ASP.NET code is not fully backward compatible with ASP.

ASP.NET Contains a large set of HTML Controls; almost all HTML elements on a page can be defined as ASP.NET Controls objects that can be controlled by scripts.

Overall, ASP.NET is a powerful and versatile framework that can be used to build a wide range of web applications, from small websites to large-scale enterprise applications.

## ➤ C#

C# is an object-oriented programming language from Microsoft that aims to combine the computing power of C++ with the programming ease of Visual Basic. C# is based on C++ and contains features similar to those of Java.

C# is designed to work with Microsoft's .Net platform. Microsoft's aim is to facilitate the exchange of information and services over the Web, and to enable developers to build highly portable applications.

C# simplifies programming through its use of Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP) which allow access to a programming object or method without requiring the programmer to write additional code for each step. Because programmers can build on existing code, rather than repeatedly duplicating it, C# is expected to make it faster and less expensive to get new products and services to market.

Microsoft is collaborating with ECMA (European Computer Manufacturers Association), the international standards body, to create a standard for C#. International Organization for Standardization (ISO) recognition for C# would encourage other companies to develop their own versions of the language. Companies that are already using C# include Apex Software, Bunka Orient, Component Source, devSoft, FarPoint Technologies, LEAD (Leadership for Environment & Development) Technologies, ProtoView, and Seagate Software.

## ➤ SQL Server

SQL Server is a relational database management system (RDBMS) from Microsoft that's designed for the enterprise environment. SQL Server runs on T-SQL (Transact -SQL), a set of programming extensions from Sybase and Microsoft that add several features to standard SQL, including transaction control, exception and error handling, row processing, and declared variables.

Code named Yukon in development; SQL Server 2005 was released in November 2005. The 2005 product is said to provide enhanced flexibility, scalability, reliability, and security to database applications, and to make them easier to create and deploy, thus reducing the complexity and tedium involved in database management. SQL Server 2005 also includes more administrative support.

The original SQL Server code was developed by Sybase; in the late 1980s, Microsoft, Sybase and Ashton-Tate collaborated to produce the first version of the product, SQL Server. Subsequently, both Sybase and Microsoft offered SQL Server products. Sybase has since renamed their product Adaptive Server Enterprise.

## ➤ Java Script

- **JavaScript Origins:**

JavaScript was released by Netscape and Sun Microsystems in 1995. However, JavaScript is not the same thing as Java.

- It is a programming language.
- It is an interpreted language.
- It is object-based programming.
- It is widely used and supported.
- It is accessible to the beginner.

- **Uses of JavaScript:**

Use it to add multimedia elements With JavaScript you can show, hide, change, resize images, and create image rollovers. You can create scrolling text across the status bar.

Create pages dynamically Based on the user's choices, the date, or other external data, JavaScript can produce pages that are customized to the user.

## ➤ HTML

HTML is the standard Mark-up language for creating web pages and CSS and JavaScript it forms a triad of cornerstone technologies for the WWW.

Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages.

HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

Hyper Text Mark-up Language (HTML) can embed programs written in a scripting language such as JavaScript which affect the behavior and content of web pages.

Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

## ➤ CSS

Cascading Style Sheet (CSS) is a Style sheet language used for describing the presentation of a document written in a Mark-up language.

Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media.

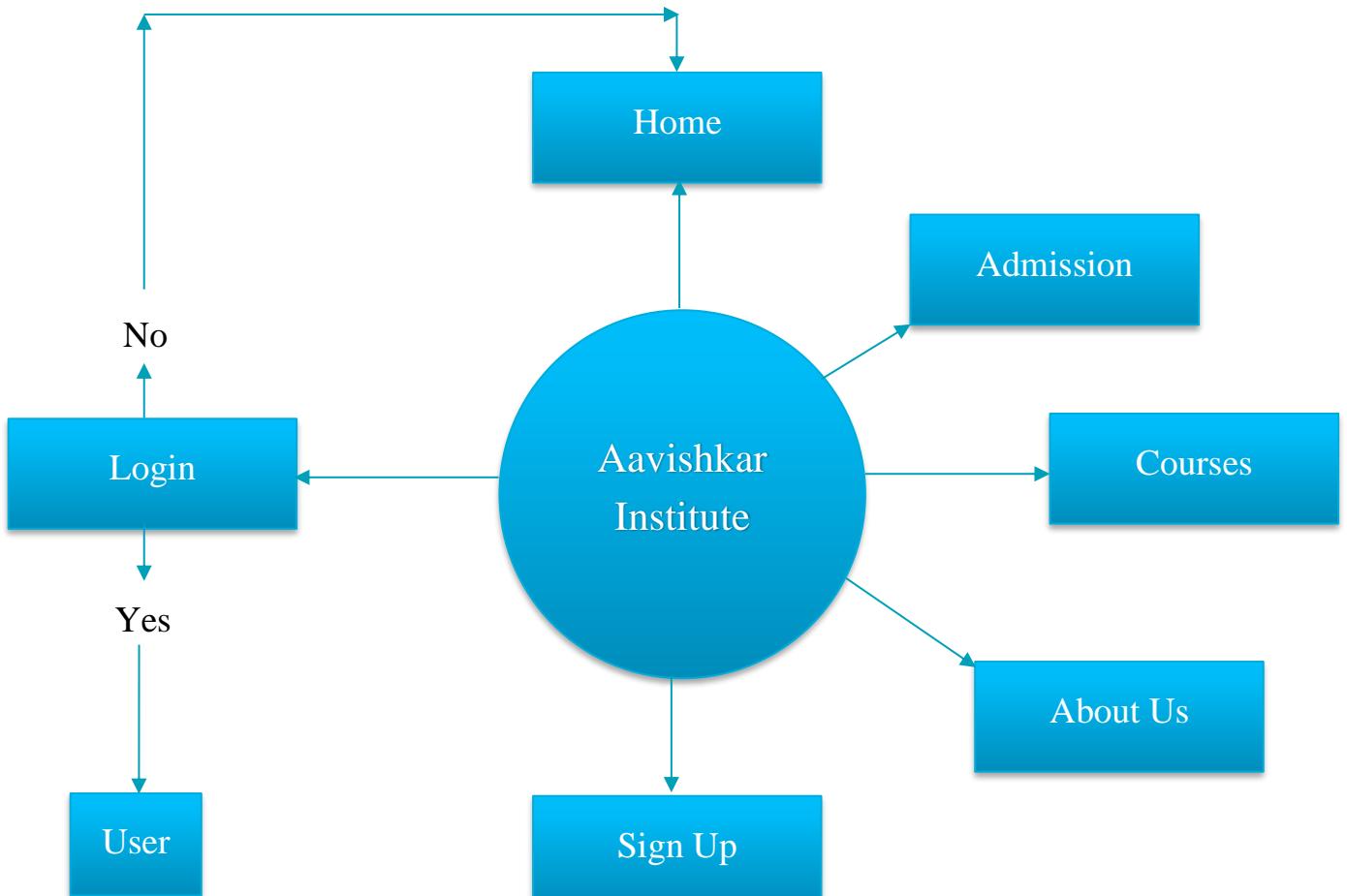
Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging web pages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colors, and fonts.

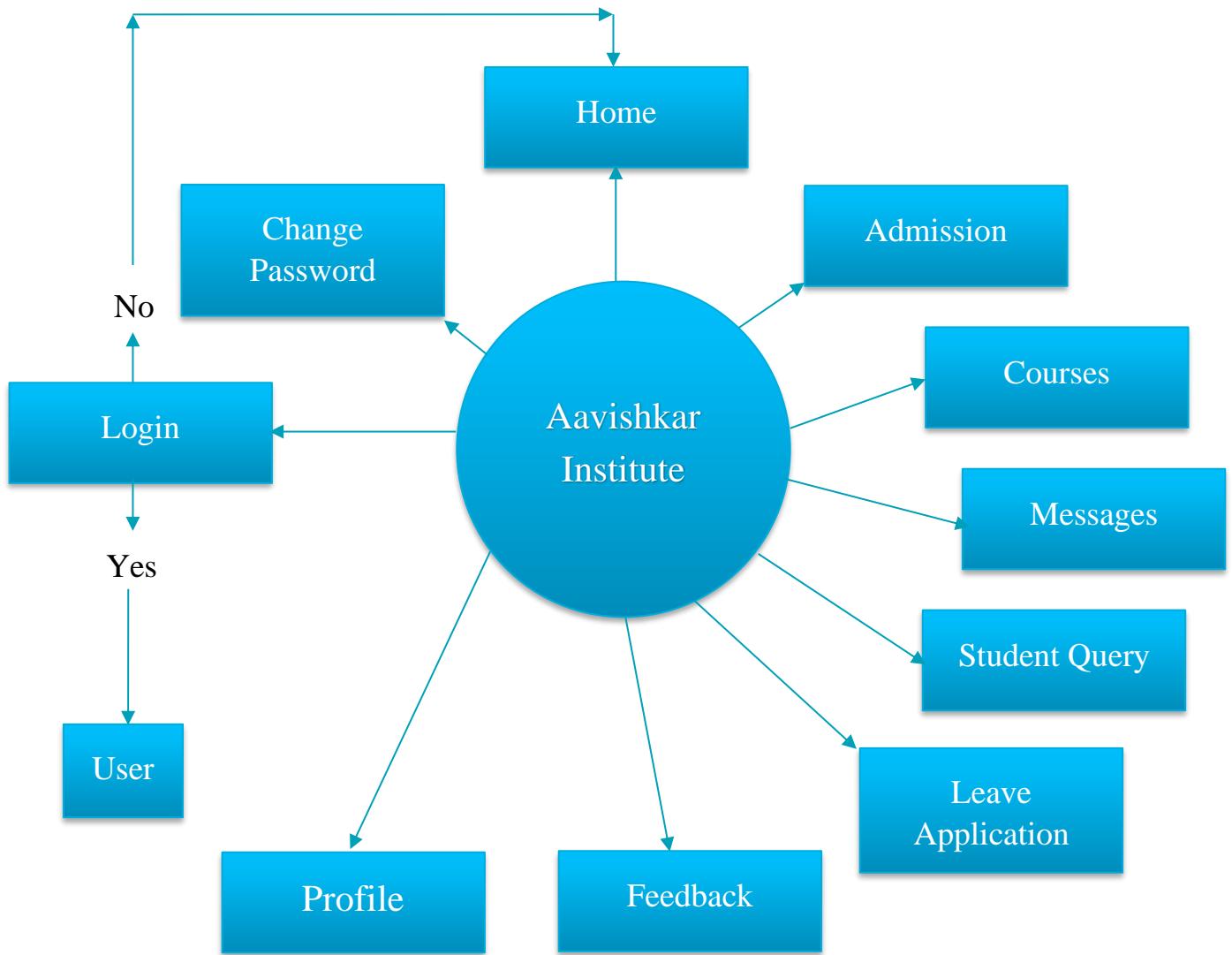
### 3.7 Data Models

#### ➤ Data Flow Diagram

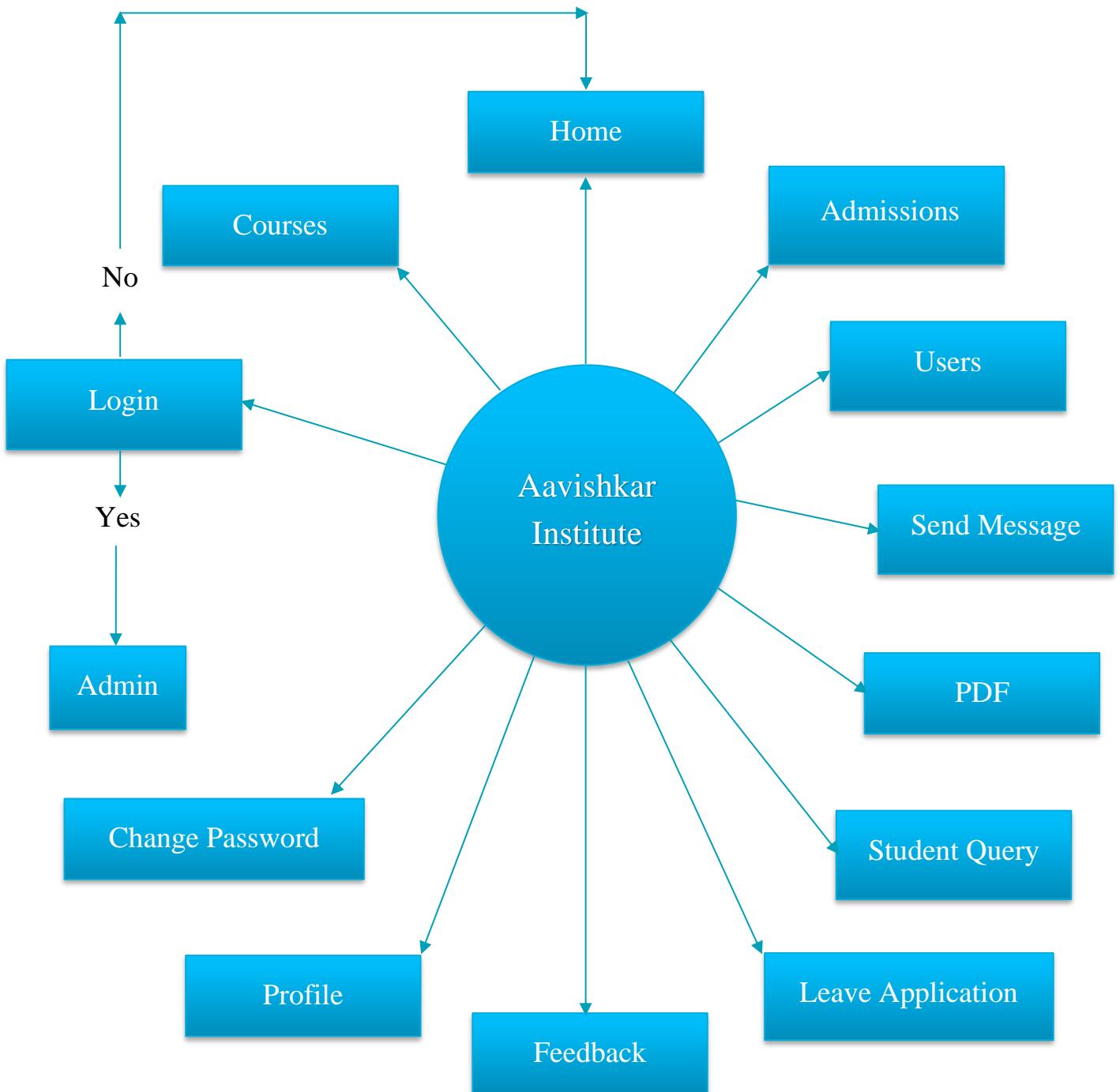
##### 1. 0-Level Visitor Data Flow Diagram:



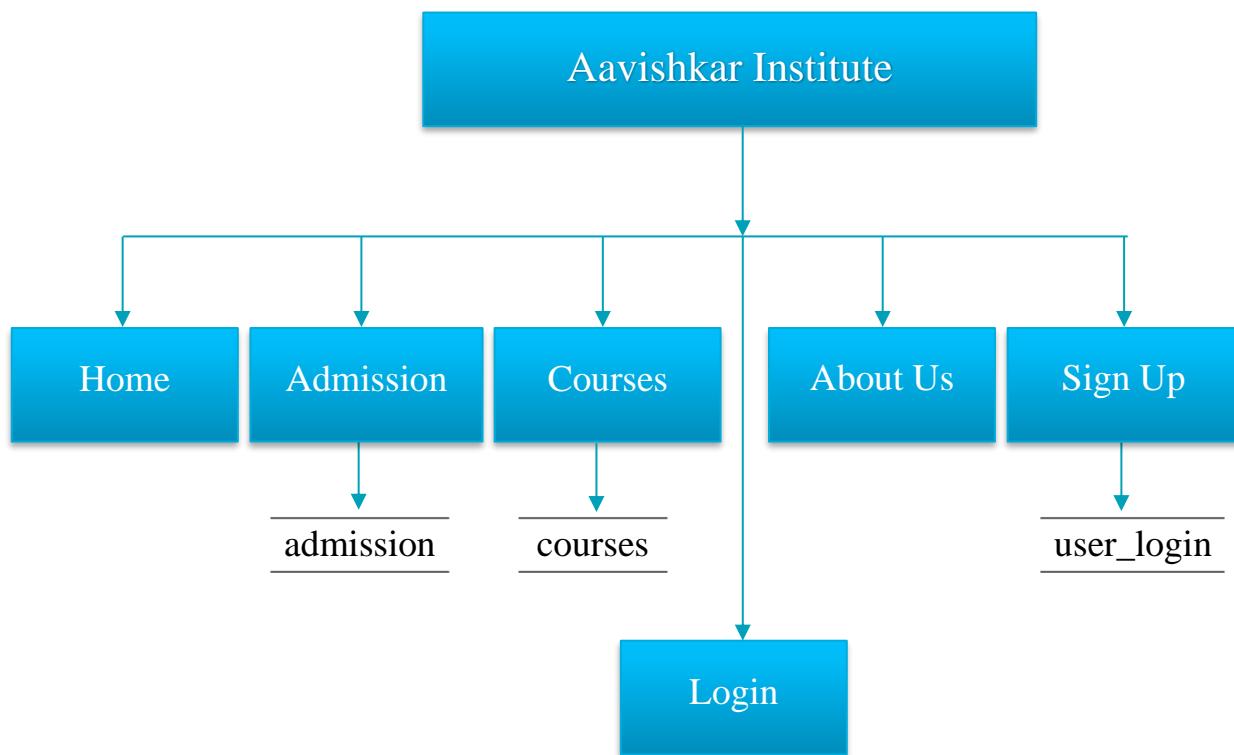
## 2. 0-Level User Data Flow Diagram:



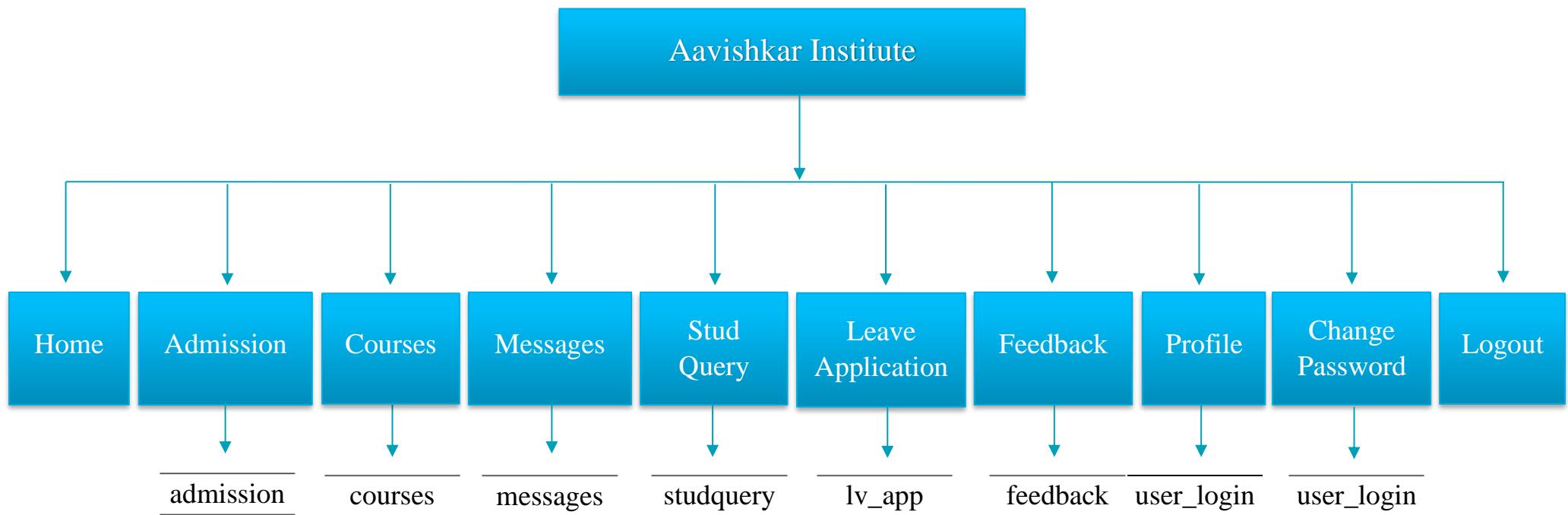
### 3. 0-Level Admin Data Flow Diagram:



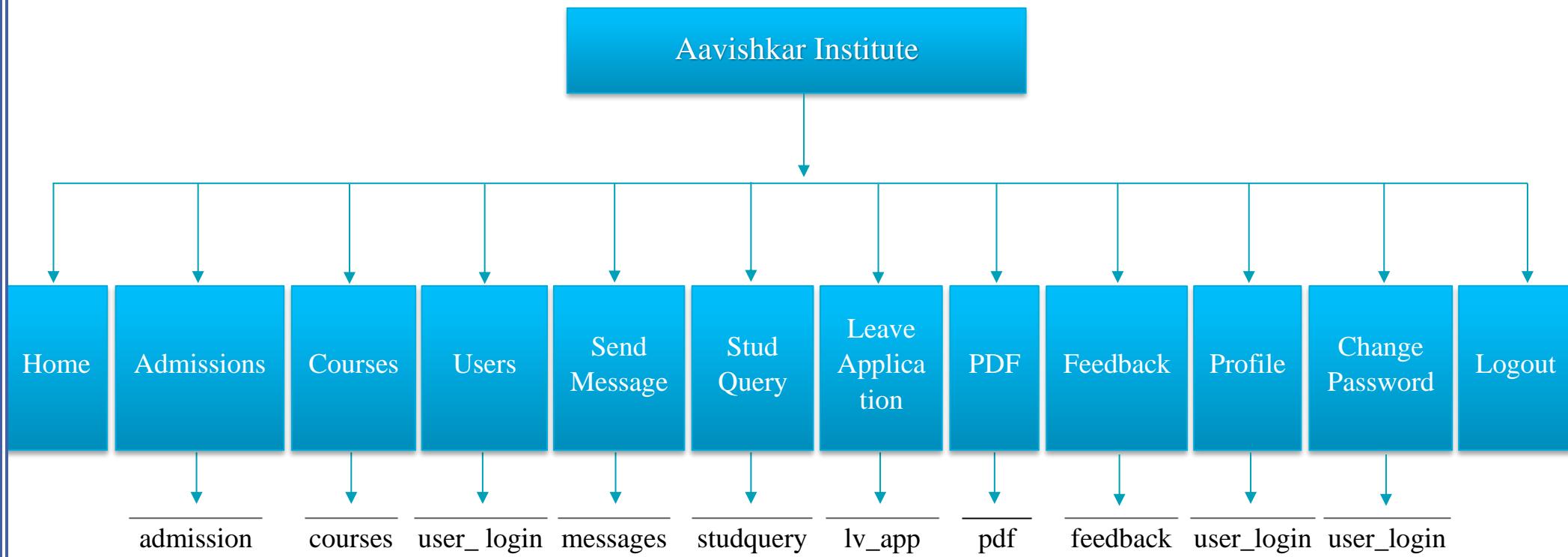
### 1. 1-Level Visitor Data Flow Diagram:



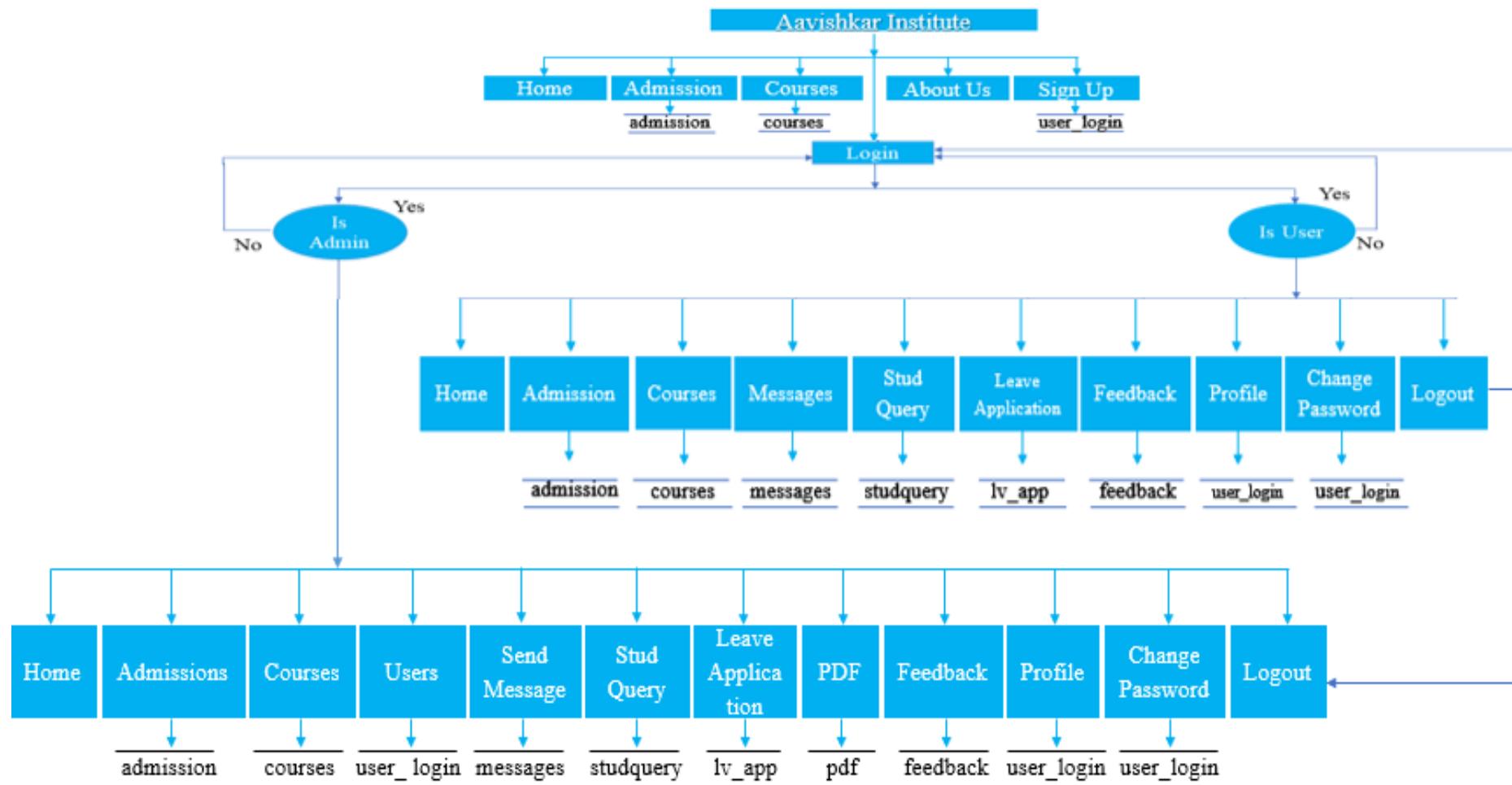
## 2. 1-Level User Data Flow Diagram:



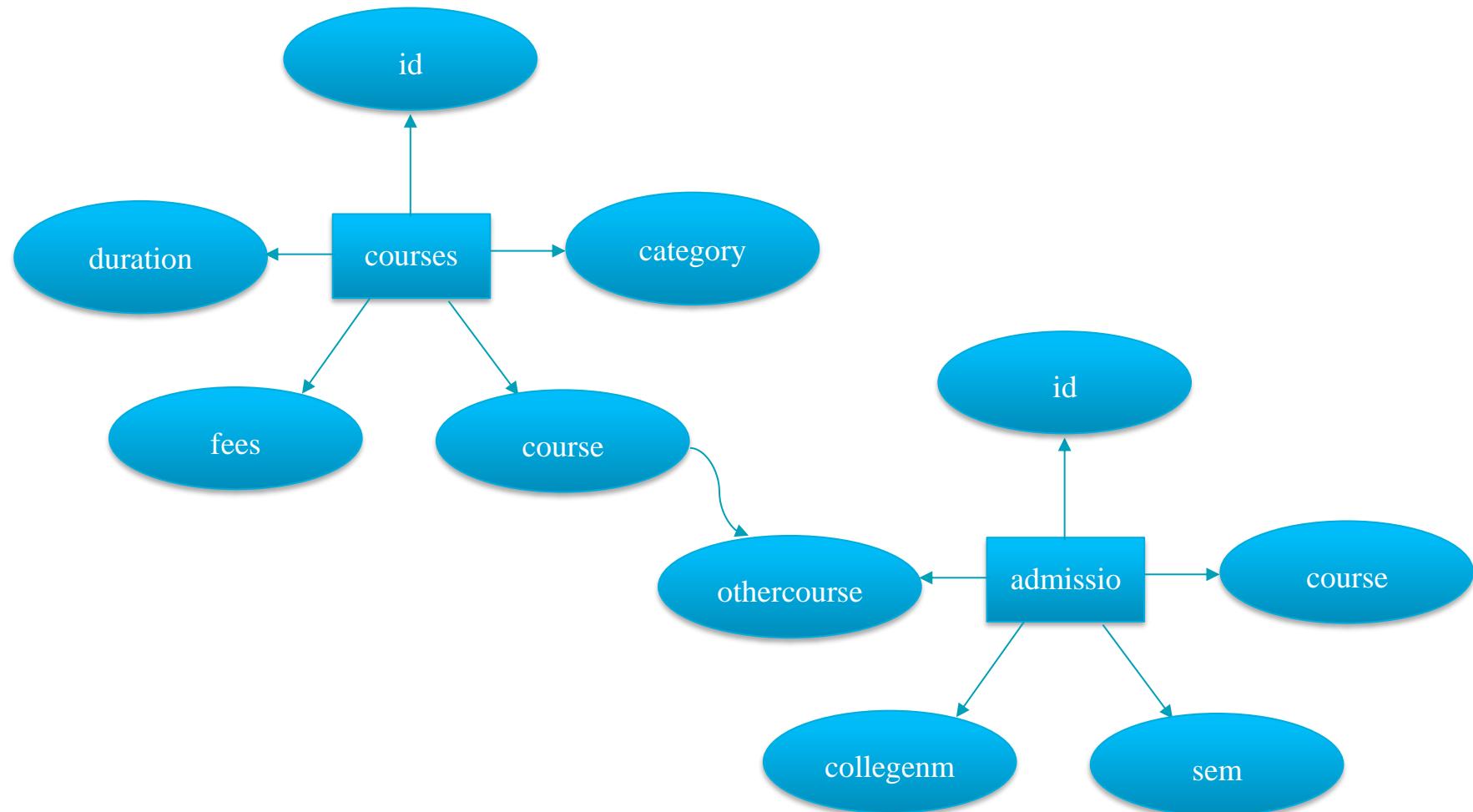
### 3. 1-Level Admin Data Flow Diagram:



## 1. Context-Level Visitor, User, Admin Data Flow Diagram:

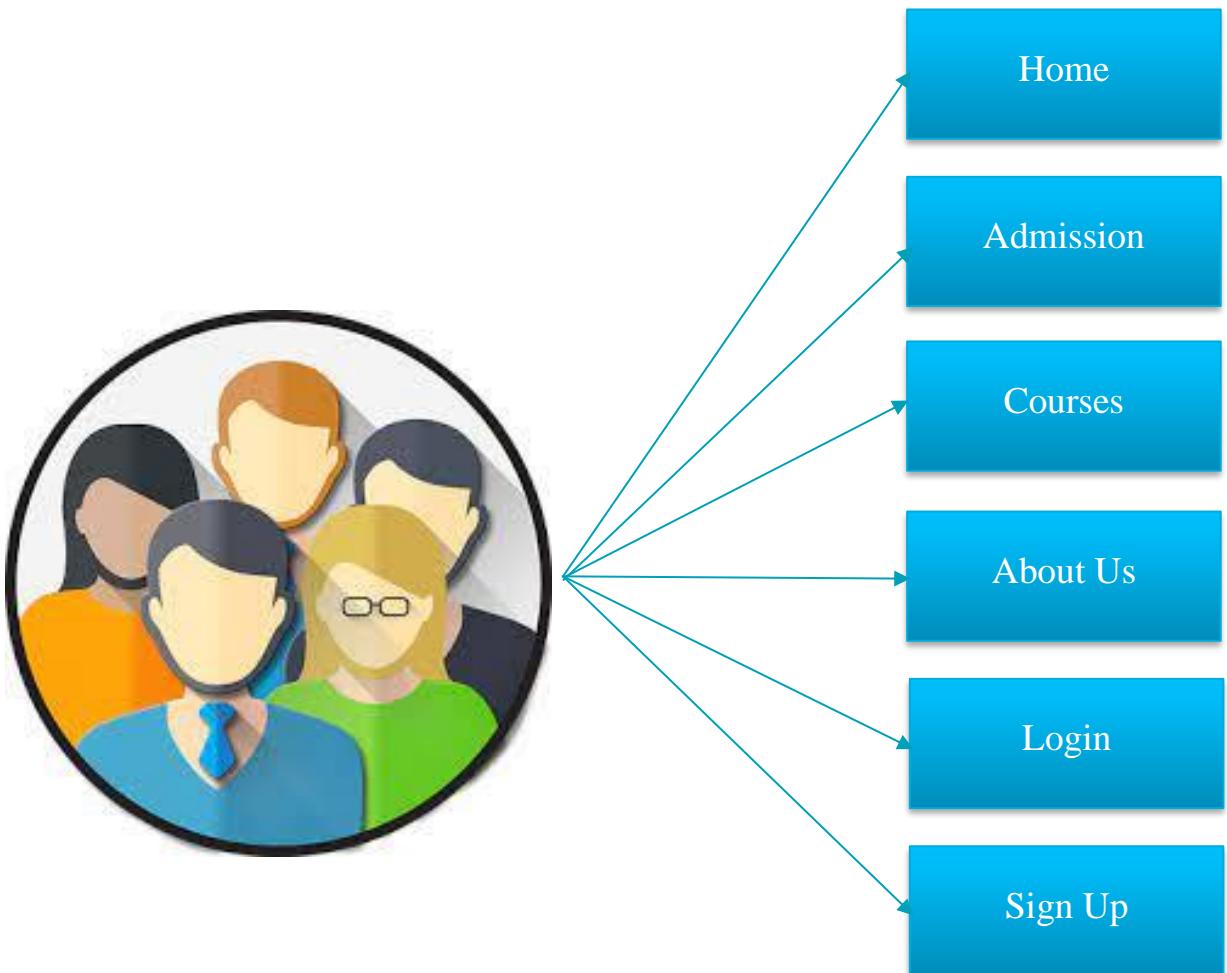


## ➤ ER Diagram

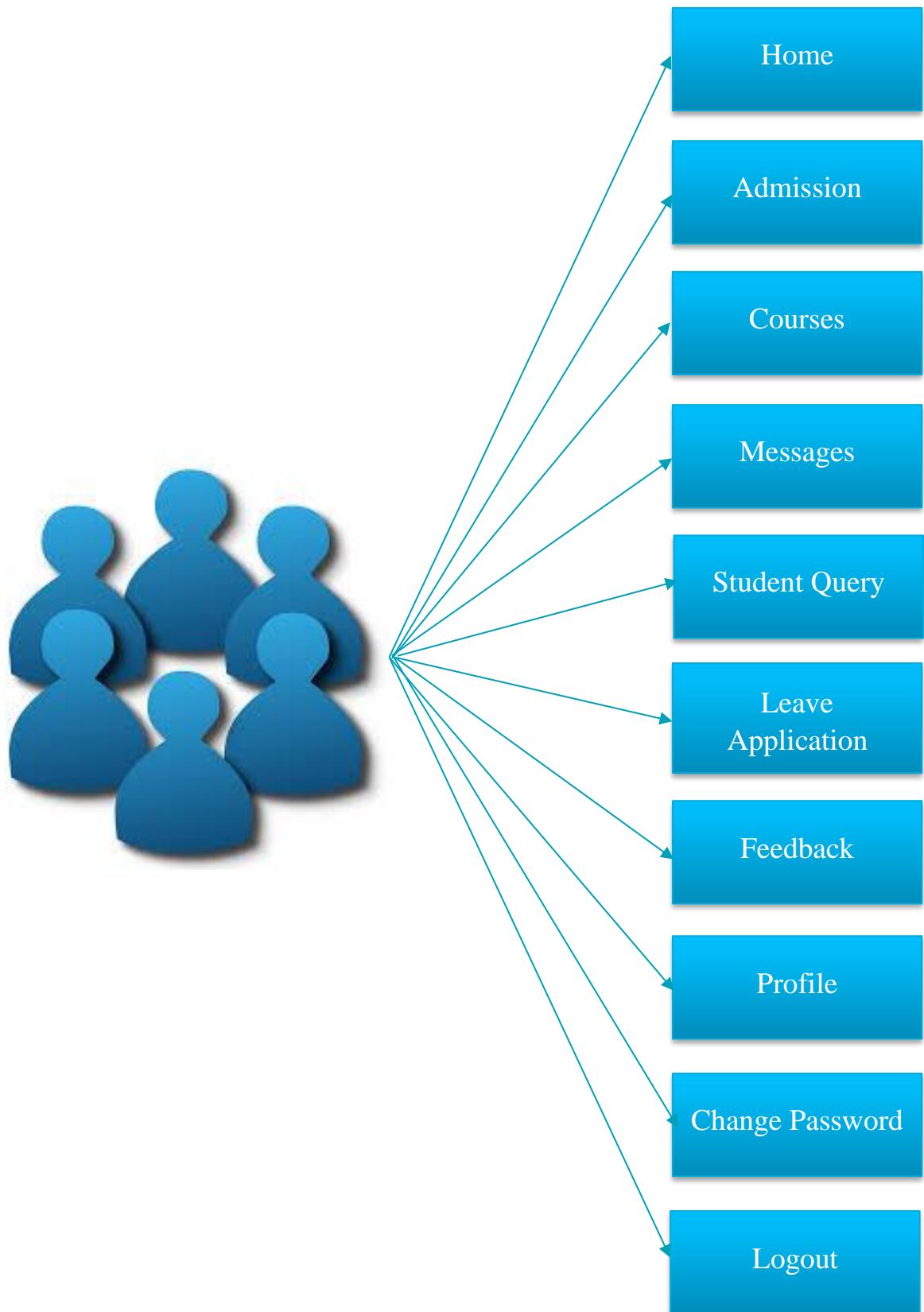


➤ Use Case Diagram

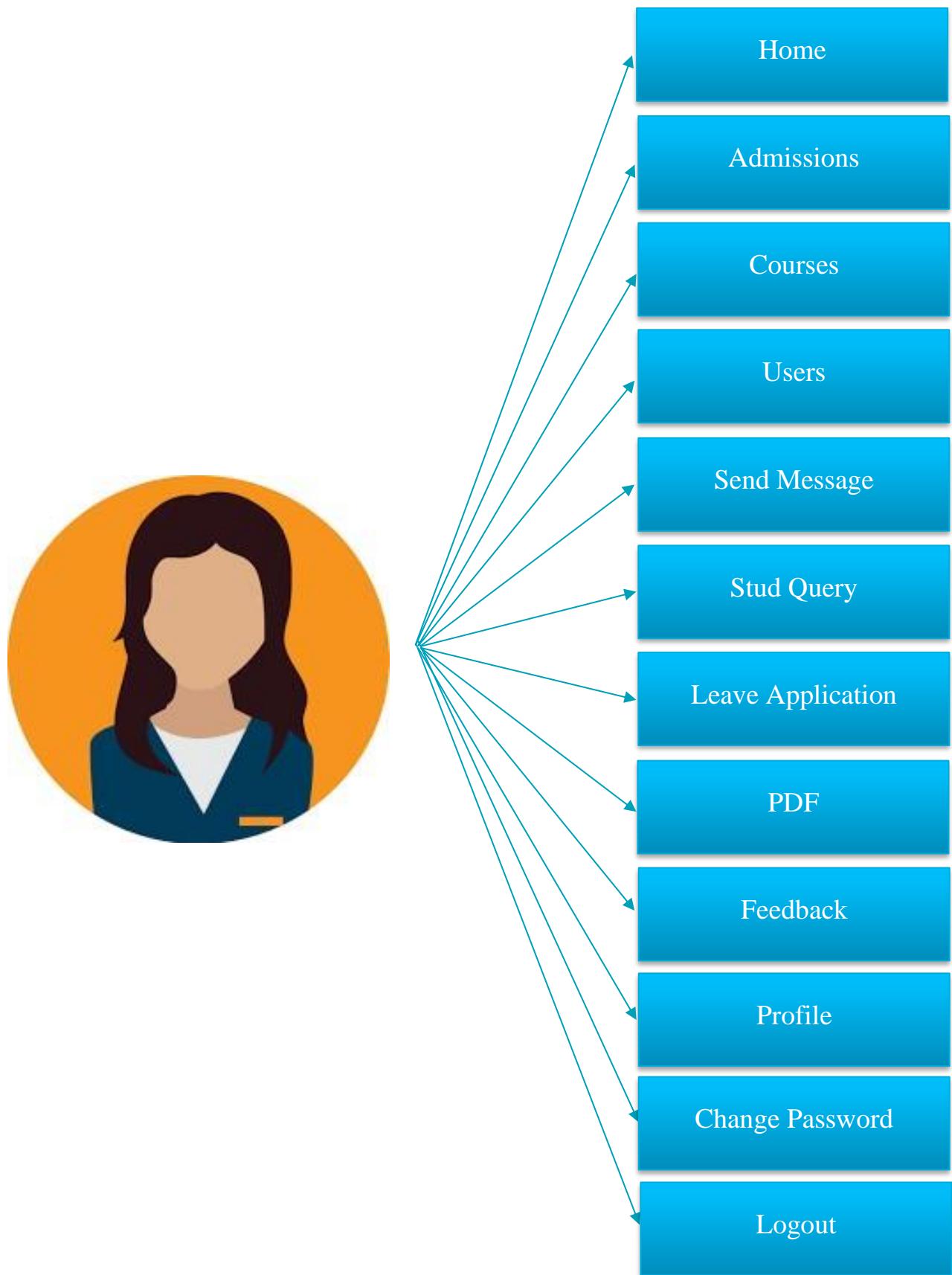
1. Visitor Use Case Diagram:



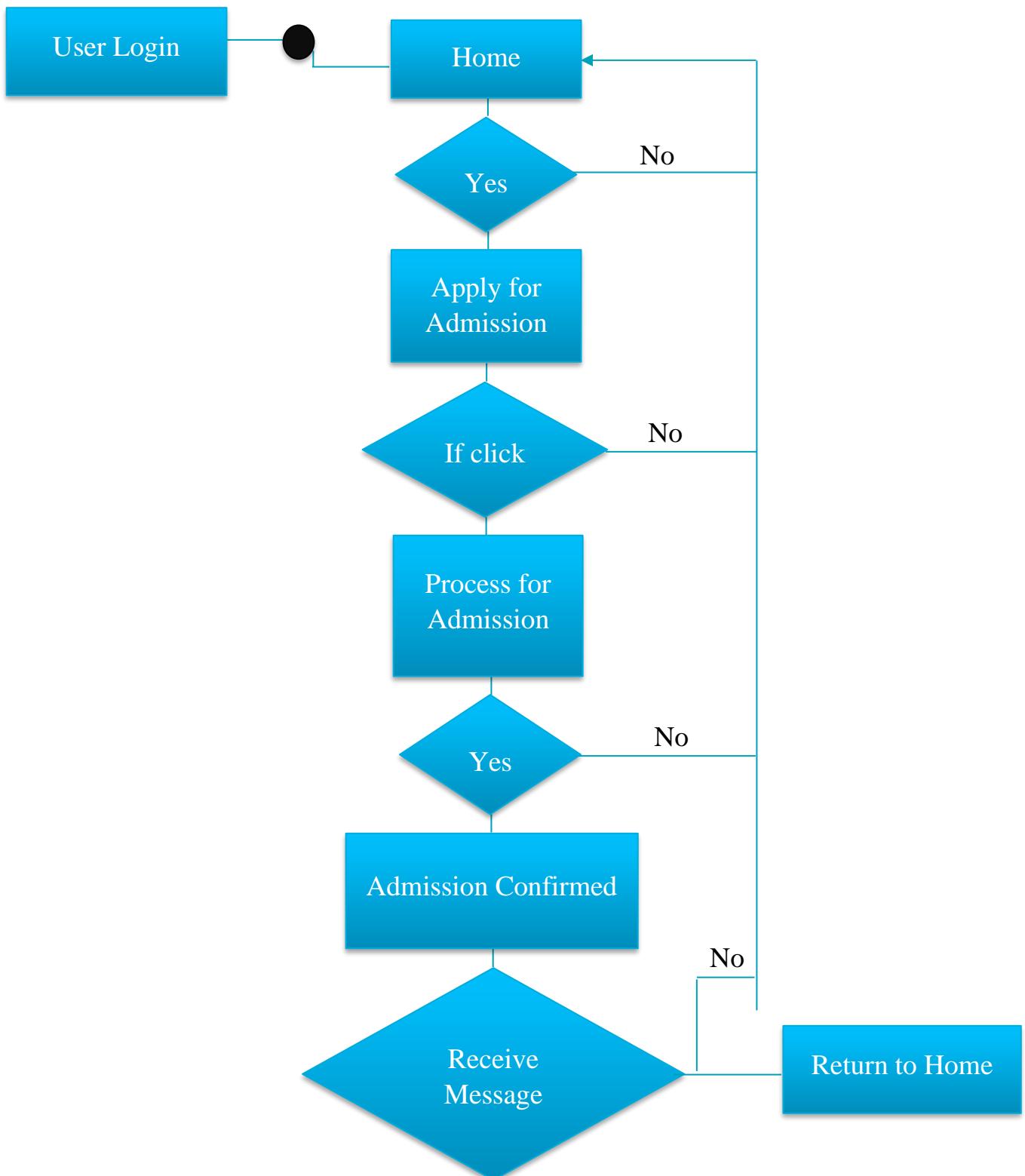
## 2. User Use Case Diagram:



### 3. Admin Use Case Diagram:



## ➤ Activity Diagram



## 3.8 Normalization

- Normalization rules are divided into following normal forms:

### 1. First Normal Form (1NF)

As per First Normal Form, no two rows of data can contain repeating group of information id, each set of columns must have a unique value, such that multiple columns cannot be used to fetch the same row. Each table should be organized into rows, and each row should have a primary key that distinguishes it as unique.

The Primary key is usually a single column, but sometimes more than one column can be combined to create a single primary key. For example, consider a table which is not in First normal form.

### 2. Second Normal Form (2NF)

As per the Second Normal Form there must not be any partial dependency of any column on primary key. It means that for a table that has concatenated primary key, each column in the table that is not part of the primary key must depend upon the entire concatenated key for its existence. If any column depends only on one part of the concatenated key, then the table fails Second normal form.

In User Table the candidate key will be column. Now, both the above tables qualify for Second Normal Form and will never suffer from update anomalies. Although there are a few complex cases in which table in Second Normal Form suffers update anomalies, and to handle those scenarios Third Normal Form is there.

### 3. Third Normal Form (3NF)

Third Normal form applies that every non-prime attribute of table must be dependent on primary key, or we can say that, there should not be the case that a non-prime attribute is determined by another non-prime attribute. So, this transitive functional dependency should be removed from the table and also the table must be in Second Normal form. For example, consider a table with following fields.

In this table User id is Primary key, but first name, last name and email depends upon mobile. The dependency between mobile and other fields is called transitive dependency. Hence to apply 3NF, we need to move the first name, last name and email to new table with mobile as primary key.

The advantage of removing transitive dependency is:

- Amount of data duplication is reduced.
- Data integrity is achieved.

## CHAPTER:4

### System Design

4.1 Tables Detail

4.2 Database Structure

4.3 Visitor Interface

4.4 User Interface

4.5 Admin Interface

4.6 Source Code

4.1 Tables Detail

➤ Visitor Side Table:

1. user\_login

	Name	Data Type	Allow Nulls
1	Id	int	<input type="checkbox"/>
	name	varchar(30)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mno	varchar(20)	<input checked="" type="checkbox"/>
	password	varchar(20)	<input checked="" type="checkbox"/>
	usertype	varchar(20)	<input checked="" type="checkbox"/>

## 2. courses

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	category	varchar(30)	<input checked="" type="checkbox"/>
	Course	varchar(50)	<input checked="" type="checkbox"/>
	fees	varchar(20)	<input checked="" type="checkbox"/>
	duration	varchar(20)	<input checked="" type="checkbox"/>

### ➤ User Side Table:

#### 1. user\_login

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	name	varchar(30)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mno	varchar(20)	<input checked="" type="checkbox"/>
	password	varchar(20)	<input checked="" type="checkbox"/>
	usertype	varchar(20)	<input checked="" type="checkbox"/>

## 2. courses

	Name	Data Type	Allow Nulls
1	Id	int	<input type="checkbox"/>
2	category	varchar(30)	<input checked="" type="checkbox"/>
3	Course	varchar(50)	<input checked="" type="checkbox"/>
4	fees	varchar(20)	<input checked="" type="checkbox"/>
5	duration	varchar(20)	<input checked="" type="checkbox"/>

## 3. studquery

	Name	Data Type	Allow Nulls
1	Id	int	<input type="checkbox"/>
2	name	varchar(30)	<input checked="" type="checkbox"/>
3	email	varchar(50)	<input checked="" type="checkbox"/>
4	mno	varchar(20)	<input checked="" type="checkbox"/>
5	query	varchar(MAX)	<input checked="" type="checkbox"/>
6	reply	varchar(MAX)	<input checked="" type="checkbox"/>

#### 4. admission

	Name	Data Type	Allow Nulls
1	Id	int	<input type="checkbox"/>
2	fnm	varchar(20)	<input checked="" type="checkbox"/>
3	lnm	varchar(20)	<input checked="" type="checkbox"/>
4	fathernm	varchar(20)	<input checked="" type="checkbox"/>
5	dob	varchar(20)	<input checked="" type="checkbox"/>
6	gender	varchar(10)	<input checked="" type="checkbox"/>
7	adhaarno	varchar(20)	<input checked="" type="checkbox"/>
8	photo	varchar(MAX)	<input checked="" type="checkbox"/>
9	parentprof	varchar(20)	<input checked="" type="checkbox"/>
10	codept	varchar(30)	<input checked="" type="checkbox"/>
11	designation	varchar(30)	<input checked="" type="checkbox"/>
12	address	varchar(100)	<input checked="" type="checkbox"/>
13	whatsappno	varchar(20)	<input checked="" type="checkbox"/>
14	email	varchar(50)	<input checked="" type="checkbox"/>
15	studmno	varchar(20)	<input checked="" type="checkbox"/>
16	pmno	varchar(20)	<input checked="" type="checkbox"/>
17	lastexam	varchar(50)	<input checked="" type="checkbox"/>
18	yrofpassing	varchar(10)	<input checked="" type="checkbox"/>
19	instnm	varchar(50)	<input checked="" type="checkbox"/>
20	pr	varchar(10)	<input checked="" type="checkbox"/>
21	std	varchar(10)	<input checked="" type="checkbox"/>
22	schoolnm	varchar(50)	<input checked="" type="checkbox"/>
23	course	varchar(50)	<input checked="" type="checkbox"/>
24	sem	varchar(10)	<input checked="" type="checkbox"/>
25	collegenm	varchar(50)	<input checked="" type="checkbox"/>
26	othercourse	varchar(50)	<input checked="" type="checkbox"/>
27	status	varchar(10)	<input checked="" type="checkbox"/>

## 5. lv\_app

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	name	varchar(30)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mno	varchar(20)	<input checked="" type="checkbox"/>
	subject	varchar(100)	<input checked="" type="checkbox"/>
	from_dt	varchar(20)	<input checked="" type="checkbox"/>
	to_dt	varchar(20)	<input checked="" type="checkbox"/>
	lv_reason	varchar(MAX)	<input checked="" type="checkbox"/>
	status	varchar(20)	<input checked="" type="checkbox"/>

## 6. feedback

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	name	varchar(30)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mno	varchar(20)	<input checked="" type="checkbox"/>
	feedback	varchar(MAX)	<input checked="" type="checkbox"/>

## 7. messages

	Name	Data Type	Allow Nulls
④	Id	int	<input type="checkbox"/>
	subject	varchar(100)	<input checked="" type="checkbox"/>
	message	varchar(MAX)	<input checked="" type="checkbox"/>
	date	varchar(20)	<input checked="" type="checkbox"/>

## 8. pdf

	Name	Data Type	Allow Nulls
④	Id	int	<input type="checkbox"/>
	course	varchar(10)	<input checked="" type="checkbox"/>
	sem	varchar(10)	<input checked="" type="checkbox"/>
	subject	varchar(50)	<input checked="" type="checkbox"/>
	pdf	varchar(MAX)	<input checked="" type="checkbox"/>
	path	varchar(MAX)	<input checked="" type="checkbox"/>

➤ Admin Side Table:

1. user\_login

	Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	name	varchar(30)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mno	varchar(20)	<input checked="" type="checkbox"/>
	password	varchar(20)	<input checked="" type="checkbox"/>
	usertype	varchar(20)	<input checked="" type="checkbox"/>

2. courses

	Name	Data Type	Allow Nulls
PK	Id	int	<input type="checkbox"/>
	category	varchar(30)	<input checked="" type="checkbox"/>
	Course	varchar(50)	<input checked="" type="checkbox"/>
	fees	varchar(20)	<input checked="" type="checkbox"/>
	duration	varchar(20)	<input checked="" type="checkbox"/>

### 3. messages

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	subject	varchar(100)	<input checked="" type="checkbox"/>
	message	varchar(MAX)	<input checked="" type="checkbox"/>
	date	varchar(20)	<input checked="" type="checkbox"/>

### 4. pdf

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	course	varchar(10)	<input checked="" type="checkbox"/>
	sem	varchar(10)	<input checked="" type="checkbox"/>
	subject	varchar(50)	<input checked="" type="checkbox"/>
	pdf	varchar(MAX)	<input checked="" type="checkbox"/>
	path	varchar(MAX)	<input checked="" type="checkbox"/>

## 5. admission

	Name	Data Type	Allow Nulls
1	Id	int	<input type="checkbox"/>
2	fnm	varchar(20)	<input checked="" type="checkbox"/>
3	lnm	varchar(20)	<input checked="" type="checkbox"/>
4	fathernm	varchar(20)	<input checked="" type="checkbox"/>
5	dob	varchar(20)	<input checked="" type="checkbox"/>
6	gender	varchar(10)	<input checked="" type="checkbox"/>
7	adhaarno	varchar(20)	<input checked="" type="checkbox"/>
8	photo	varchar(MAX)	<input checked="" type="checkbox"/>
9	parentprof	varchar(20)	<input checked="" type="checkbox"/>
10	codept	varchar(30)	<input checked="" type="checkbox"/>
11	designation	varchar(30)	<input checked="" type="checkbox"/>
12	address	varchar(100)	<input checked="" type="checkbox"/>
13	whatsappno	varchar(20)	<input checked="" type="checkbox"/>
14	email	varchar(50)	<input checked="" type="checkbox"/>
15	studmno	varchar(20)	<input checked="" type="checkbox"/>
16	pmno	varchar(20)	<input checked="" type="checkbox"/>
17	lastexam	varchar(50)	<input checked="" type="checkbox"/>
18	yrofpassing	varchar(10)	<input checked="" type="checkbox"/>
19	instnm	varchar(50)	<input checked="" type="checkbox"/>
20	pr	varchar(10)	<input checked="" type="checkbox"/>
21	std	varchar(10)	<input checked="" type="checkbox"/>
22	schoolnm	varchar(50)	<input checked="" type="checkbox"/>
23	course	varchar(50)	<input checked="" type="checkbox"/>
24	sem	varchar(10)	<input checked="" type="checkbox"/>
25	collegenm	varchar(50)	<input checked="" type="checkbox"/>
26	othercourse	varchar(50)	<input checked="" type="checkbox"/>
27	status	varchar(10)	<input checked="" type="checkbox"/>

## 6. studquery

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	name	varchar(30)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mno	varchar(20)	<input checked="" type="checkbox"/>
	query	varchar(MAX)	<input checked="" type="checkbox"/>
	reply	varchar(MAX)	<input checked="" type="checkbox"/>

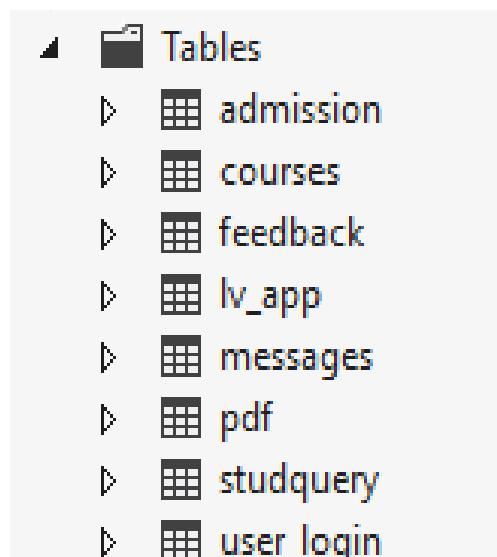
## 7. lv\_app

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	name	varchar(30)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mno	varchar(20)	<input checked="" type="checkbox"/>
	subject	varchar(100)	<input checked="" type="checkbox"/>
	from_dt	varchar(20)	<input checked="" type="checkbox"/>
	to_dt	varchar(20)	<input checked="" type="checkbox"/>
	lv_reason	varchar(MAX)	<input checked="" type="checkbox"/>
	status	varchar(20)	<input checked="" type="checkbox"/>

## 8. feedback

	Name	Data Type	Allow Nulls
10	Id	int	<input type="checkbox"/>
	name	varchar(30)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mno	varchar(20)	<input checked="" type="checkbox"/>
	feedback	varchar(MAX)	<input checked="" type="checkbox"/>

## 4.2 Database Structure



## 4.3 Visitor Interface

### 1. Home Page: This is home page for visitor.

**Aavishkar Institute**

Home Admission Courses About Us Login Sign Up

**Welcome To Aavishkar Institute**

**WHY US**

One step ahead in technology.  
Complete syllabus coverage and best faculties.  
Weekly exam program.  
Personal guidance by experts.  
Disciplined atmosphere.  
Using state of the art of the infrastructure and visualization facilities with AC class rooms.  
We provide curriculums and co-curriculums activities for overall development.  
Providing institute topper is our motto.  
Above all we provide best education in reasonable fees.

**OUR TEAM**

**Mrs. Ankita Dave**  
Director  
ankita12@gmail.com

**Mr. Pravin Varma**  
Head Of Department  
pravin88.varma@gmail.com

**Mrs. Anjali Bajpai**  
Faculty  
anjali.bajpai@gmail.com

**Mr. Kishor Soni**  
Faculty  
kishoresoni@gmail.com

**Mr. Vivek Bhatt**  
Faculty  
vivekb568@gmail.com

**Mrs. Miral Sharma**  
Faculty  
sharma03.ms@gmail.com

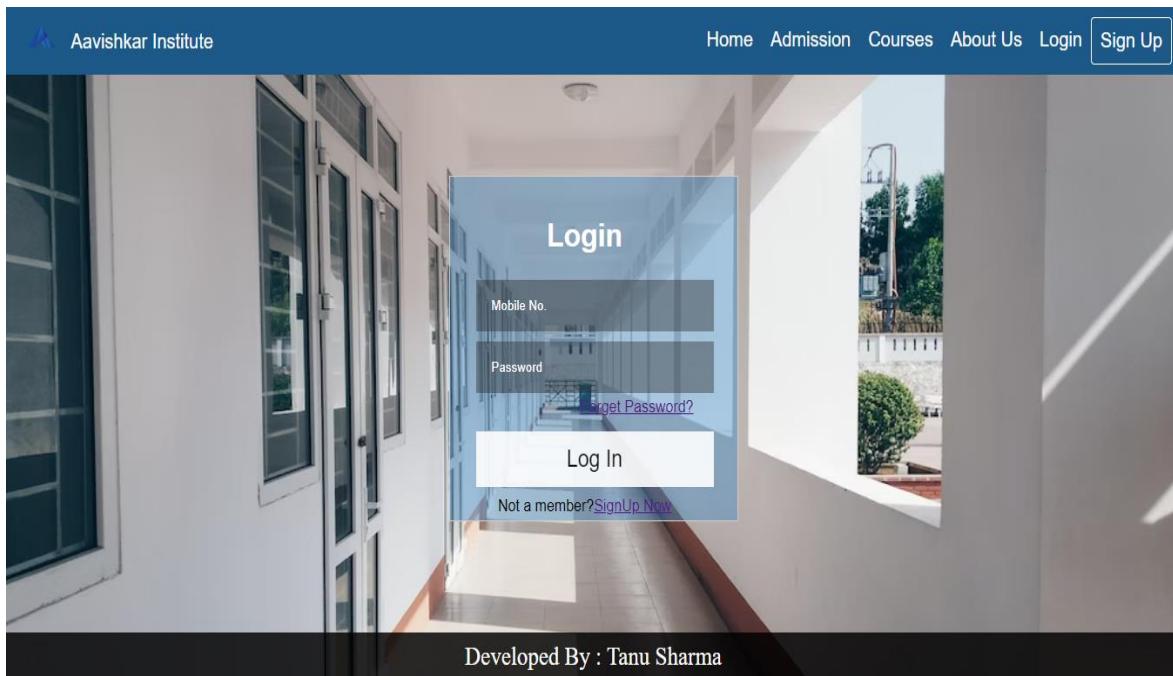
**FREQUENTLY ASKED QUESTIONS**

1. Is Aavishkar an English medium, co-educational institute?  
Yes, Aavishkar is an English medium co-educational institute.
2. What is the medium of instruction?  
English is the medium of instruction at Aavishkar. We also respect everyone's mother tongue.
3. What are the various mode of payment accepted here?  
You can make payment via Cash, Cheques.
4. Which is the nearest landmark?  
You can easily locate the establishment as it is in close proximity at 2nd floor, JP Mall, Mahatma Gandhi Road, Indore.
5. What are its hours of operation?  
The establishment is functional on Monday to Saturday 8:00 AM - 8:00 PM and on Sunday 9:00 AM - 12:00 PM.

**CONTACT US**

Mobile: 9996348876  
Email: sharma11.ts@gmail.com  
Address: 2nd floor, JP Mall, Mahatma Gandhi Road, Indore, Madhya Pradesh-452001.

2. Admission Page: User must login to apply for admission.

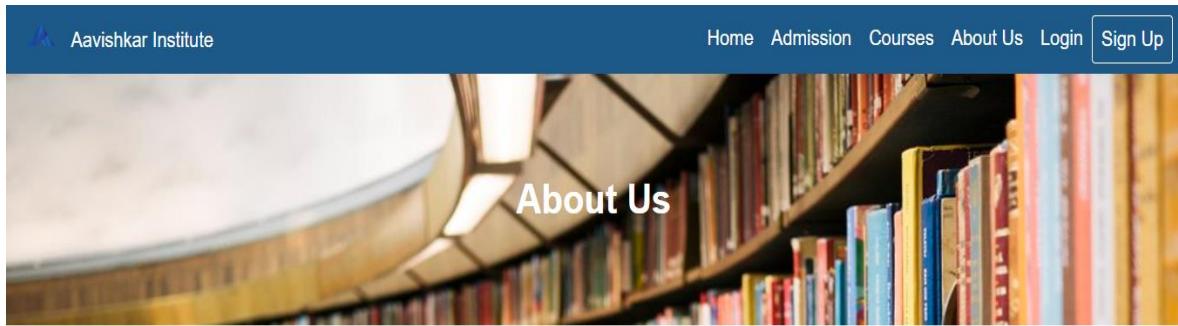


### 3. Courses Page: Visitor can see courses information.

Category	Course	Fees	Duration
HSC (General)	11th & 12th	40,000	2 years
UG	BCA	90,000	3 years
PG	MCA	1,20,000	2 years
Basic Course	MS-Office	6,000	2 months
Graphics Designing	PageMaker, CorelDRAW, Photoshop	10,000	6 months
Web Designing	HTML	3,000	1 month
Web Designing	CSS	8,000	5 months
Web Designing	JavaScript	10,000	2 months
Programming Language	C Language	4,000	1.5 months
Programming Language	C++	5,000	2 months
Programming Language	C#	4,000	1 month
Programming Language	Java	6,000	4 months
Programming Language	Python	2,000/pm	3-6 months
Programming Language	Android	7,000	3 months
Programming Language	Node.js	6,000	3 months
Programming Language	React.js	8,000	4 months
Programming Language	Angular.js	6,000	2.5 months
Database	Oracle	5,000	1 month
Database	MySQL	5,000	2-3 months
Database	SQL Server	5,000	1.5 months
Database	MongoDB	5,000	1 month
Web Programming	PHP with MySQL	6,000	2-3 months
Web Programming	ASP.Net	5,000	1 month
Web Programming	JSP	5,000	1 month

Developed By : Tanu Sharma

#### 4. About Us Page: This page describes about “Aavishkar Institute”.



We established in the year 2002 with a purpose of various length and breadth in Education at Indore, Madhya Pradesh, India. We have academic excellence in Commerce and Computer Education. We are providing high-class standards for Education. At present Aavishkar Institute is considered as one of the "Esteemed Centers" of learning. We are running long term projects and pilot plans in various Section of Commerce and Computers. Our efforts are always towards the better Carrier for students. To colorized it we are arranging various Seminars, Conferences, Lectures of experts and various tours. We are specialized to built Carrier in standard 11th and 12th Commerce (English Medium), B.C.A., M.C.A., various Computer Courses like Financial Accounting - Tally(Authorized Tally Learning Center), Graphics, Web Designing, etc.. Our number of the students by completing their Academic Carrier and Computer Courses achieved higher post in various section of the society.

##### **Our Vision**

- To Be Most Trustable And Admired Brand in the Education Sector.
- Empowering our Students to thrive in life.
- To Promote the Students to follow their Dreams and Help them to Achieve the same.

##### **Our Mission**

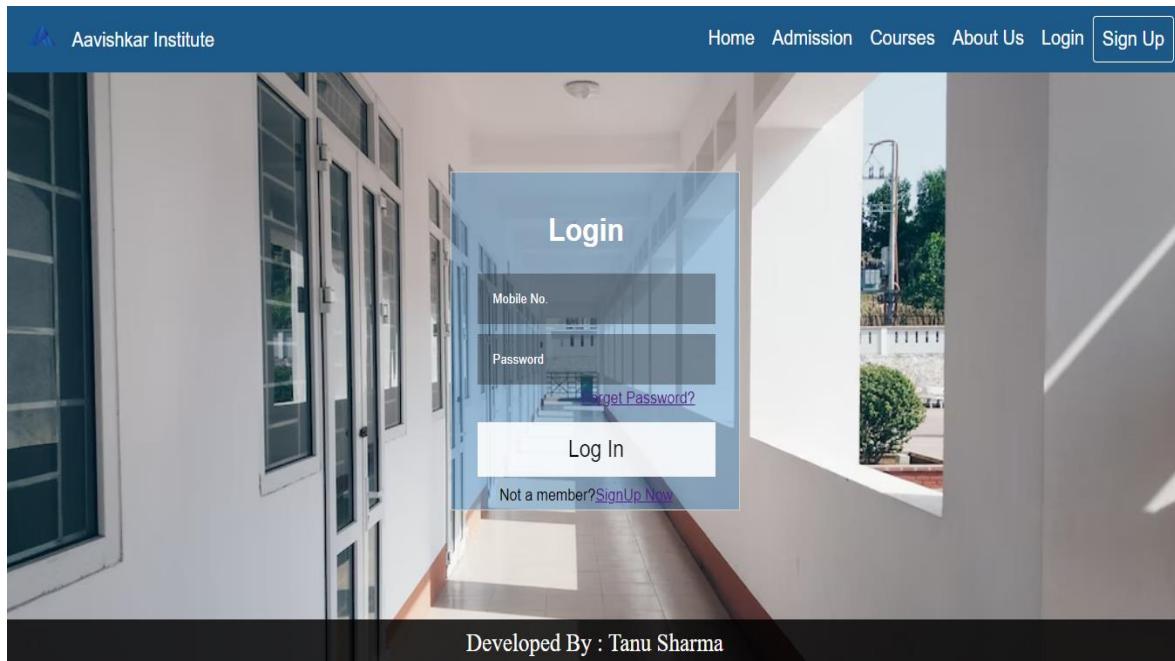
- Honesty Towards Students, Parents, Society, and Nation.
- To provide Quality Education.
- The Competence of our Methodologies and Systems.

##### **Our Values**

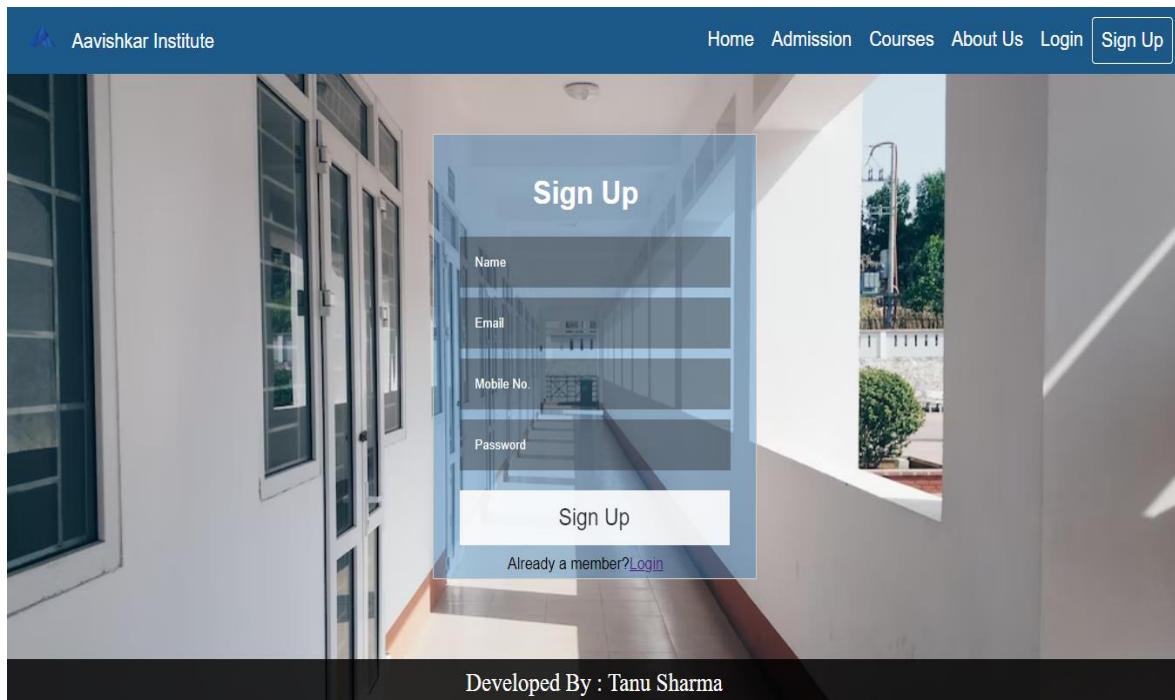
- Pursue Growth and Learning.
- Moral Development.
- Foster Collaboration and Innovation.
- Do More With Less.

Developed By : Tanu Sharma

5. Login Page: Existing user can simply login with his mobile no. and password.

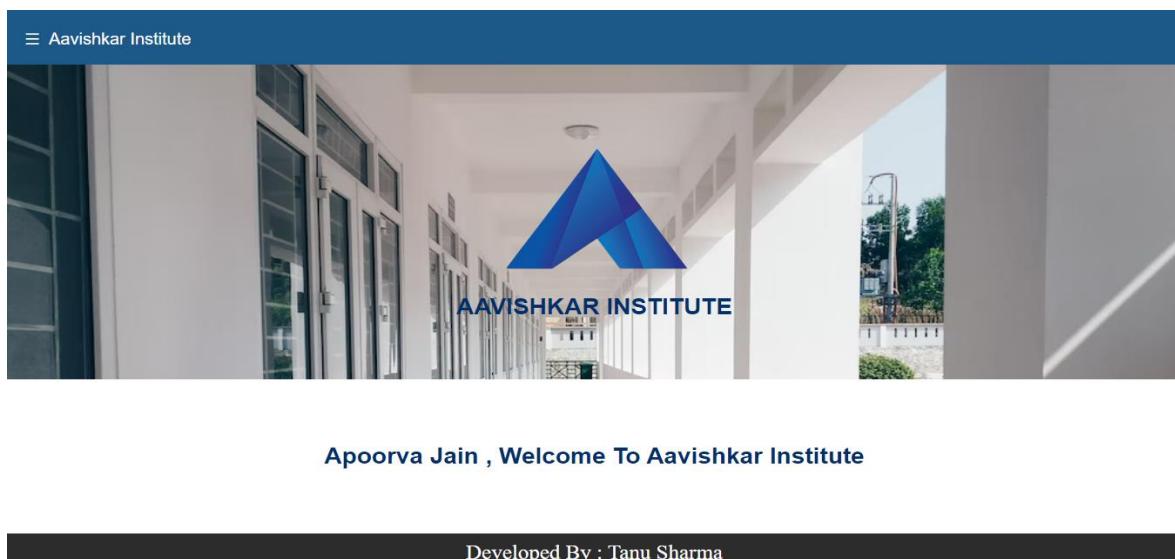


6. Sign Up Page: New user can simply sign up by providing required details.

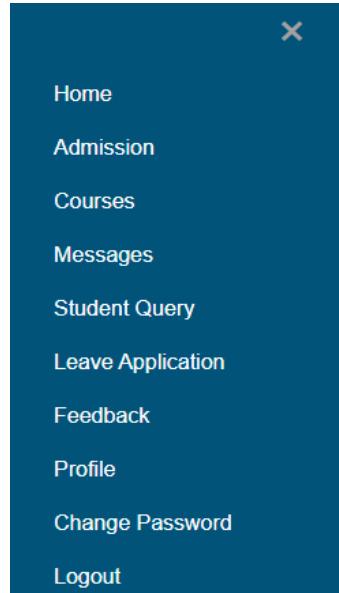


## 4.4 User Interface

1. Home Page: After user login, a home page is displayed.



On header ,there's a menu button. User can open it and go to the particular page.

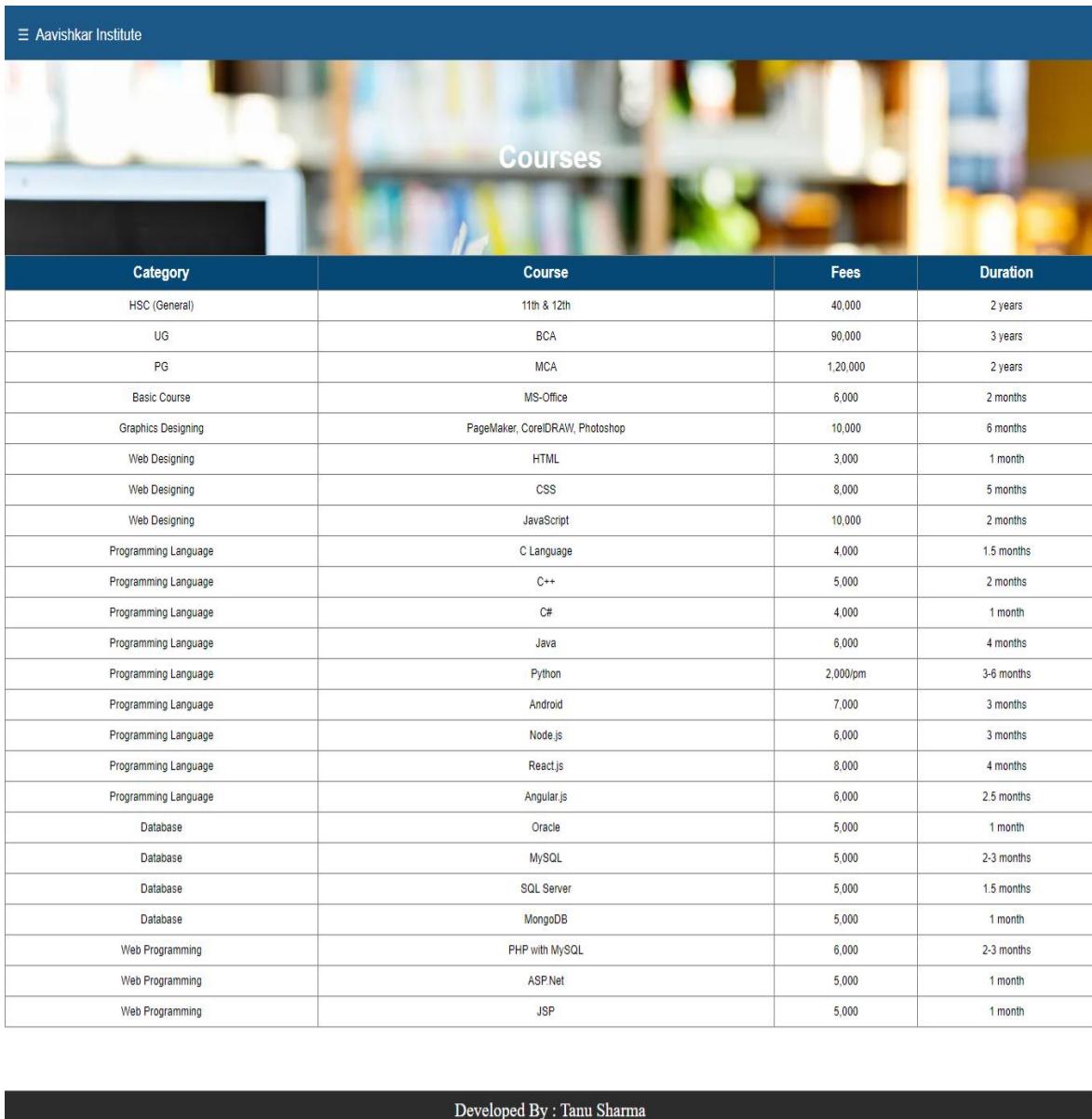


Developed By: Tanu Sharma

## 2. Admission Page: Here user can apply for admission.

The screenshot shows a web-based admission form for Aavishkar Institute. The header includes the institute's logo and name. The main section is titled "Admission Form". It contains various input fields for personal information, such as First Name, Last Name, DOB, Photo, Parent's Profession, Designation, Email, Parent's Mobile No., Year of Passing, Percentage, and Address. There are also fields for Father's Name, Gender, Adhaar Number, Company/Dept. Name, WhatsApp No., Student's Mobile No., Last/Appeared/Passed Exam, and Name of Institute. A "Commerce" section includes fields for Standard (with radio buttons for 11<sup>th</sup> & 12<sup>th</sup> and 12<sup>th</sup>), School Name, Other Courses (with a dropdown menu), and Modes of Payment (with options like Cash and Cheque). A "Computer" section includes fields for Course (with radio buttons for BCA and MCA), Semester (with a dropdown menu), and College Name. At the bottom, there is a "Submit" button and a footer note stating "Developed By : Tanu Sharma".

### 3. Courses Page: User can see courses information.



Category	Course	Fees	Duration
HSC (General)	11th & 12th	40,000	2 years
UG	BCA	90,000	3 years
PG	MCA	1,20,000	2 years
Basic Course	MS-Office	6,000	2 months
Graphics Designing	PageMaker, CorelDRAW, Photoshop	10,000	6 months
Web Designing	HTML	3,000	1 month
Web Designing	CSS	8,000	5 months
Web Designing	JavaScript	10,000	2 months
Programming Language	C Language	4,000	1.5 months
Programming Language	C++	5,000	2 months
Programming Language	C#	4,000	1 month
Programming Language	Java	6,000	4 months
Programming Language	Python	2,000/pm	3-6 months
Programming Language	Android	7,000	3 months
Programming Language	Node.js	6,000	3 months
Programming Language	React.js	8,000	4 months
Programming Language	Angular.js	6,000	2.5 months
Database	Oracle	5,000	1 month
Database	MySQL	5,000	2-3 months
Database	SQL Server	5,000	1.5 months
Database	MongoDB	5,000	1 month
Web Programming	PHP with MySQL	6,000	2-3 months
Web Programming	ASP Net	5,000	1 month
Web Programming	JSP	5,000	1 month

Developed By : Tanu Sharma

Developed By: Tanu Sharma

#### 4. Messages Page: Students can see messages send by admin.

Apoorva Shekhar Your admission is confirmed.

Your Query: I am learning ASP.Net, can I also apply for another course?  
Answer: Yes, you can apply.

Apoorva Your Leave is granted.

Date: 2023-01-26  
Subject: Republic Day Holiday  
Message: Tomorrow is Republic Day, so there will be a holiday.

Date: 2023-01-28  
Subject: Seminar  
Message: Tomorrow there will be a motivational seminar for everyone.

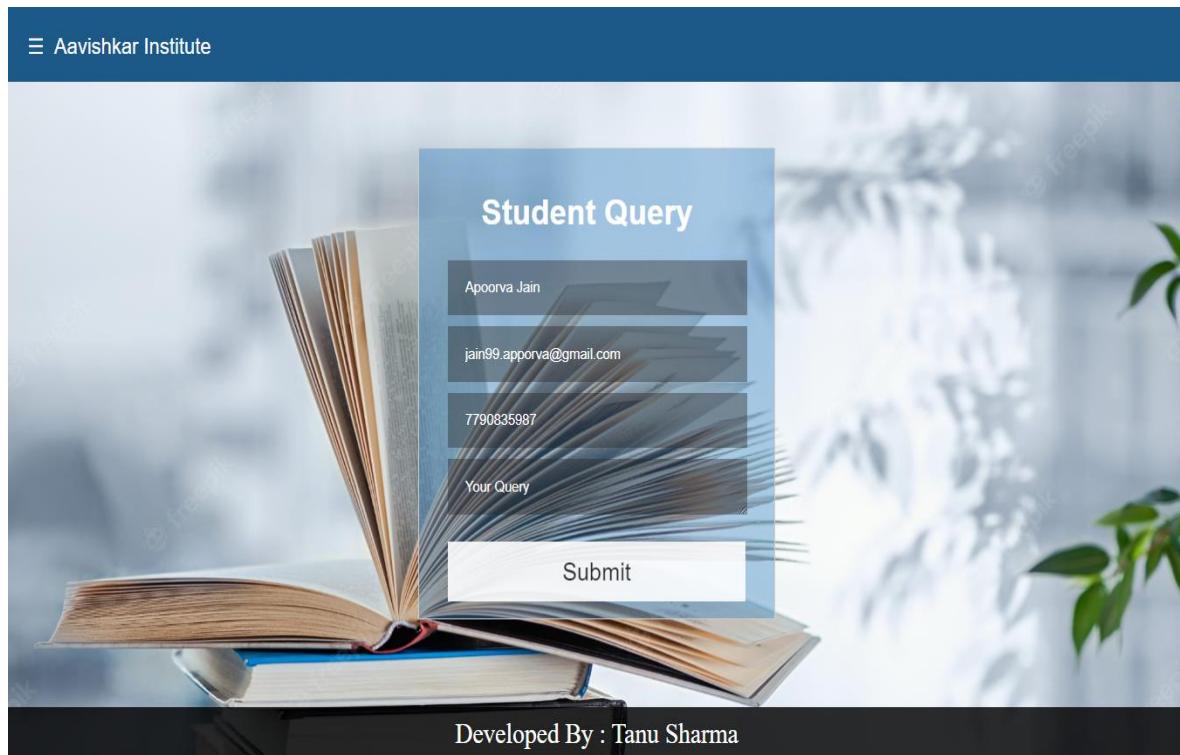
Date: 2023-02-18  
Subject: Mahashivratri Holiday  
Message: Tomorrow there will be a holiday for Mahashivratri Holiday.

Date: 2023-02-19  
Subject: Webinar  
Message: Tomorrow there will be a webinar for all of you. Kindly visit the link <https://tech.webinar.com> tomorrow.

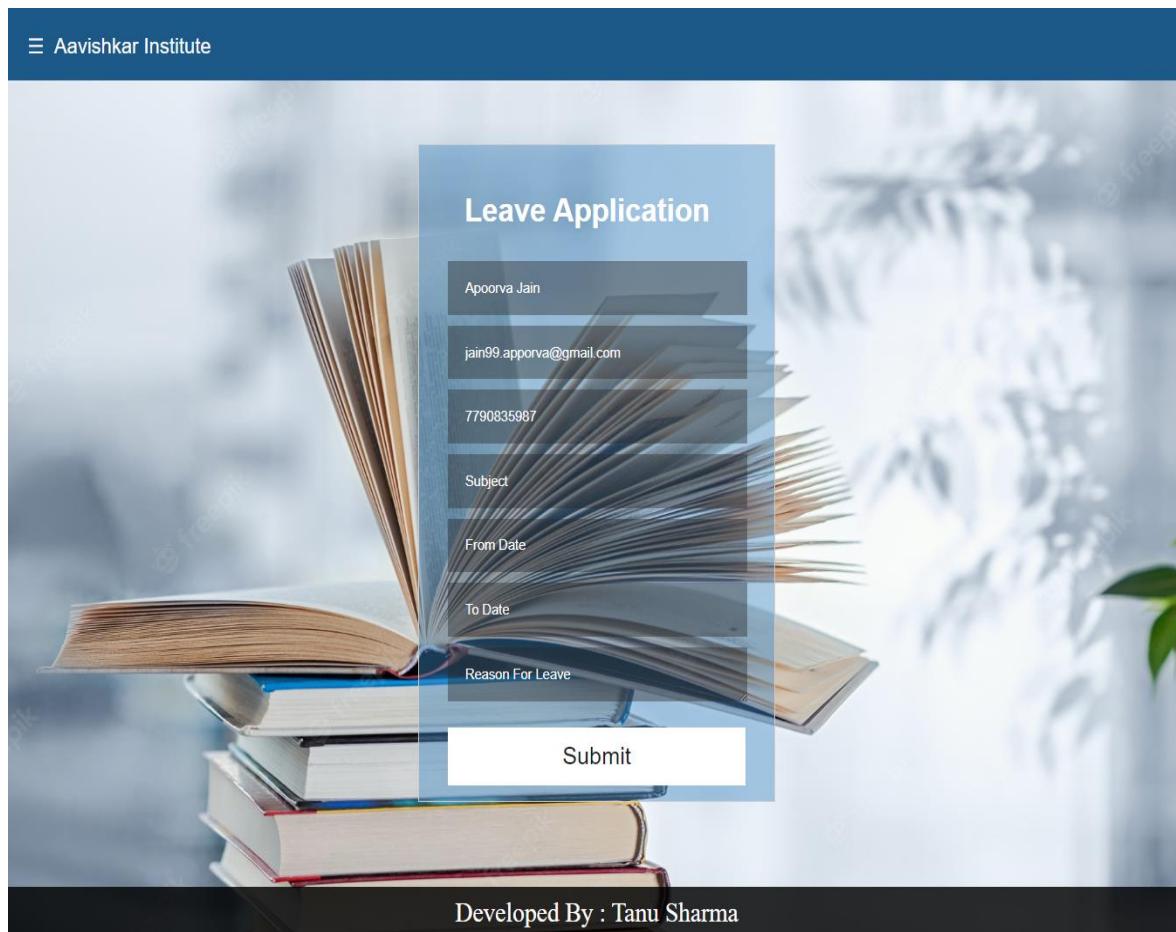
Date: 2023-02-20  
Subject: Reading holiday  
Message: We would like to inform you that there will be a 8 days reading holiday for all of you starting from February 21nd to February 28th. Regular classes will resume on March 1st. Thank you for your attention, and we wish you happy reading!

Course: BCA  
Semester: Sem-1  
Subject: C Language  
PDF: C Language.pdf  
[Click To Download](#)

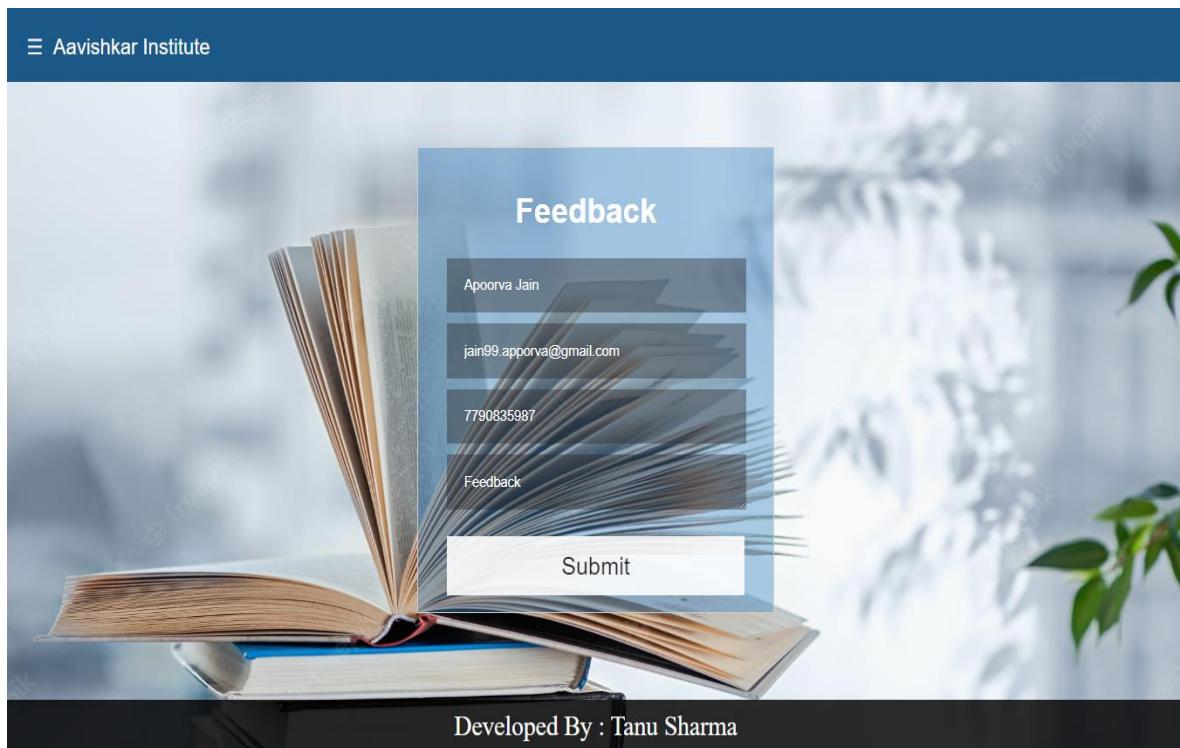
## 5. Student Query Page: Here students can ask their queries.



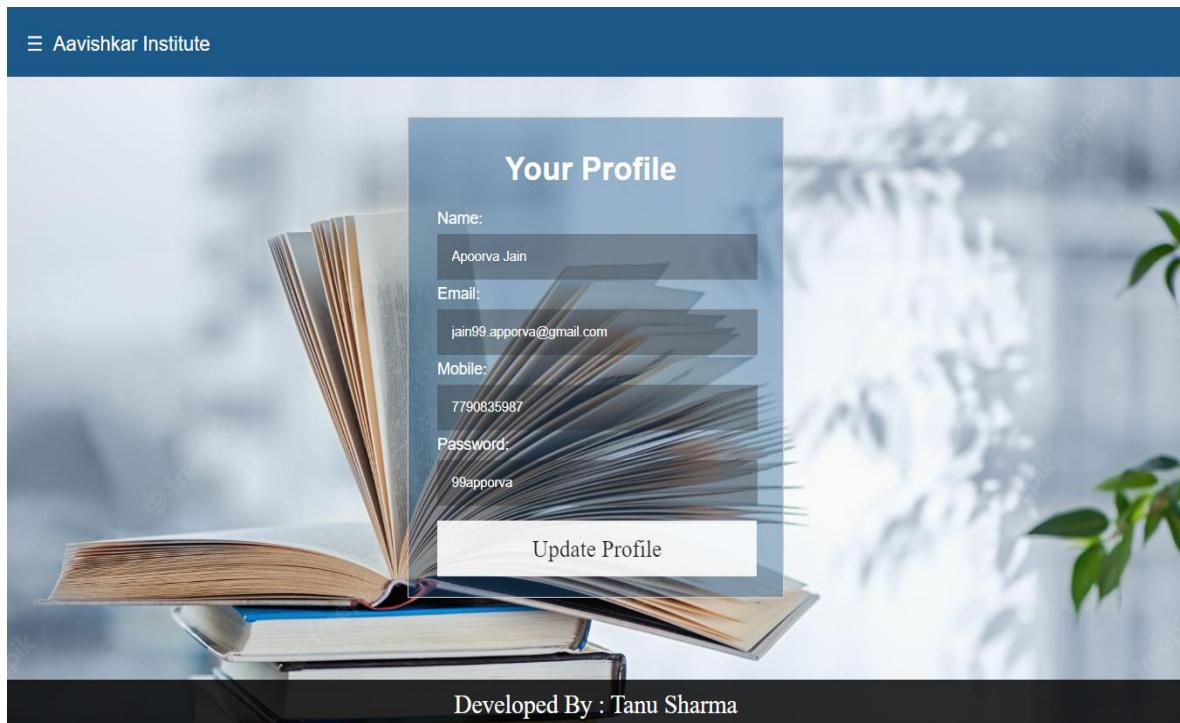
## 6. Leave Application Page: Here students can apply for leave.



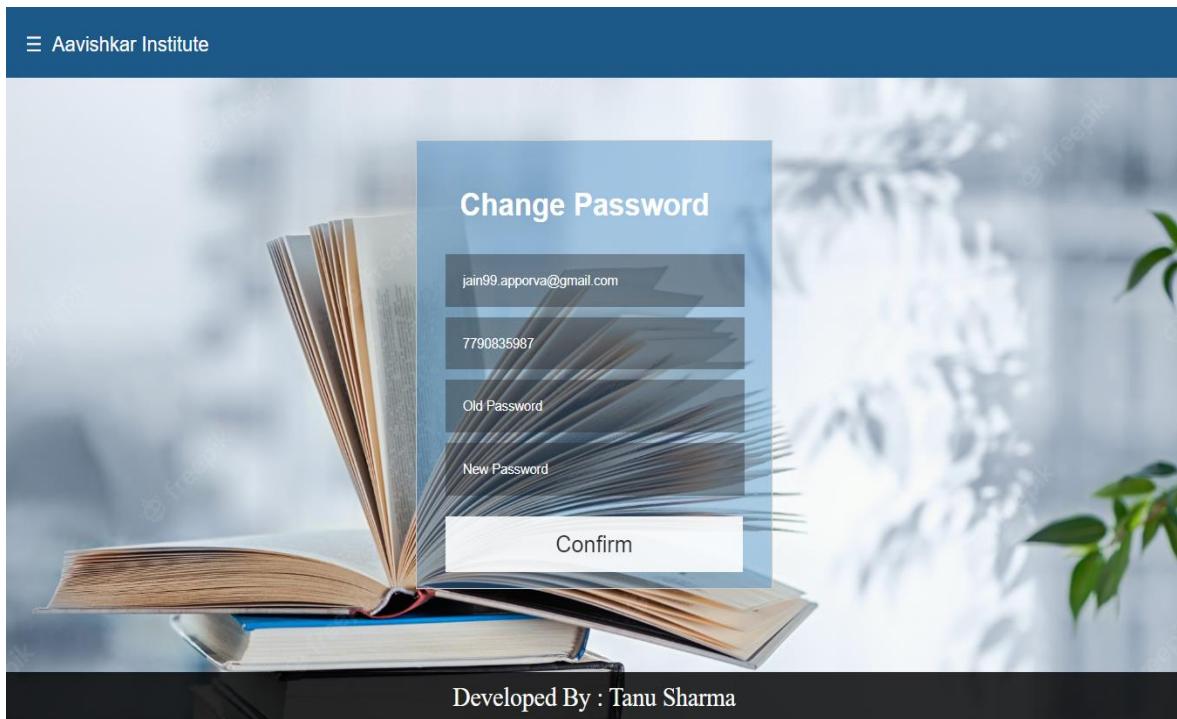
## 7. Feedback Page: Here user can give their feedback.



## 8. Profile Page: Here user can see and update his profile.

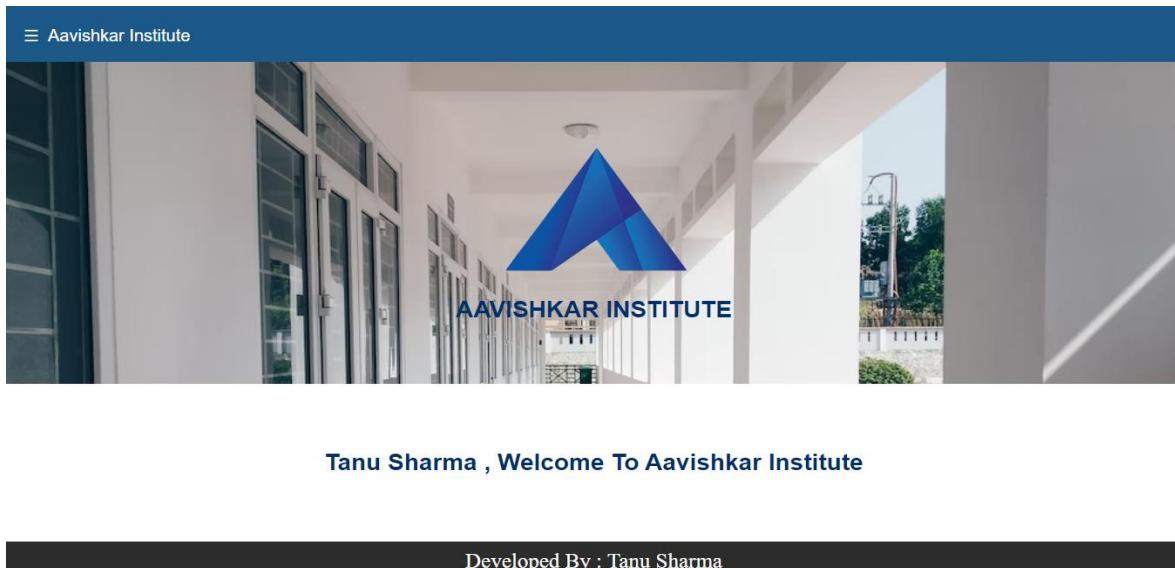


9. Change Password Page: Here user can easily change password.

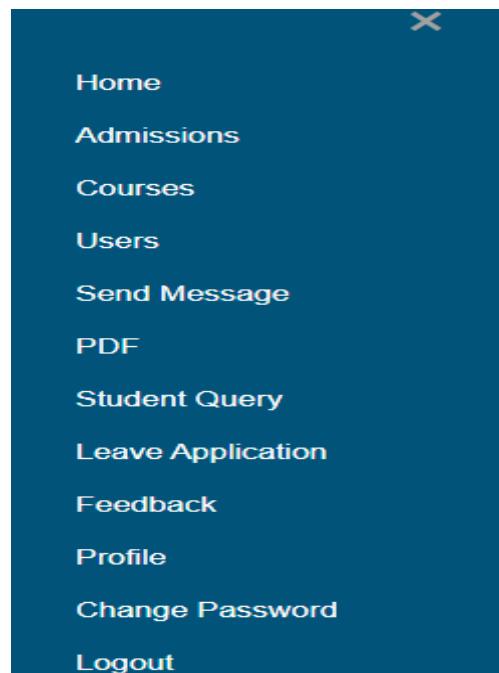


## 4.4 Admin Interface

1. Home Page: After admin login, a home page is displayed.

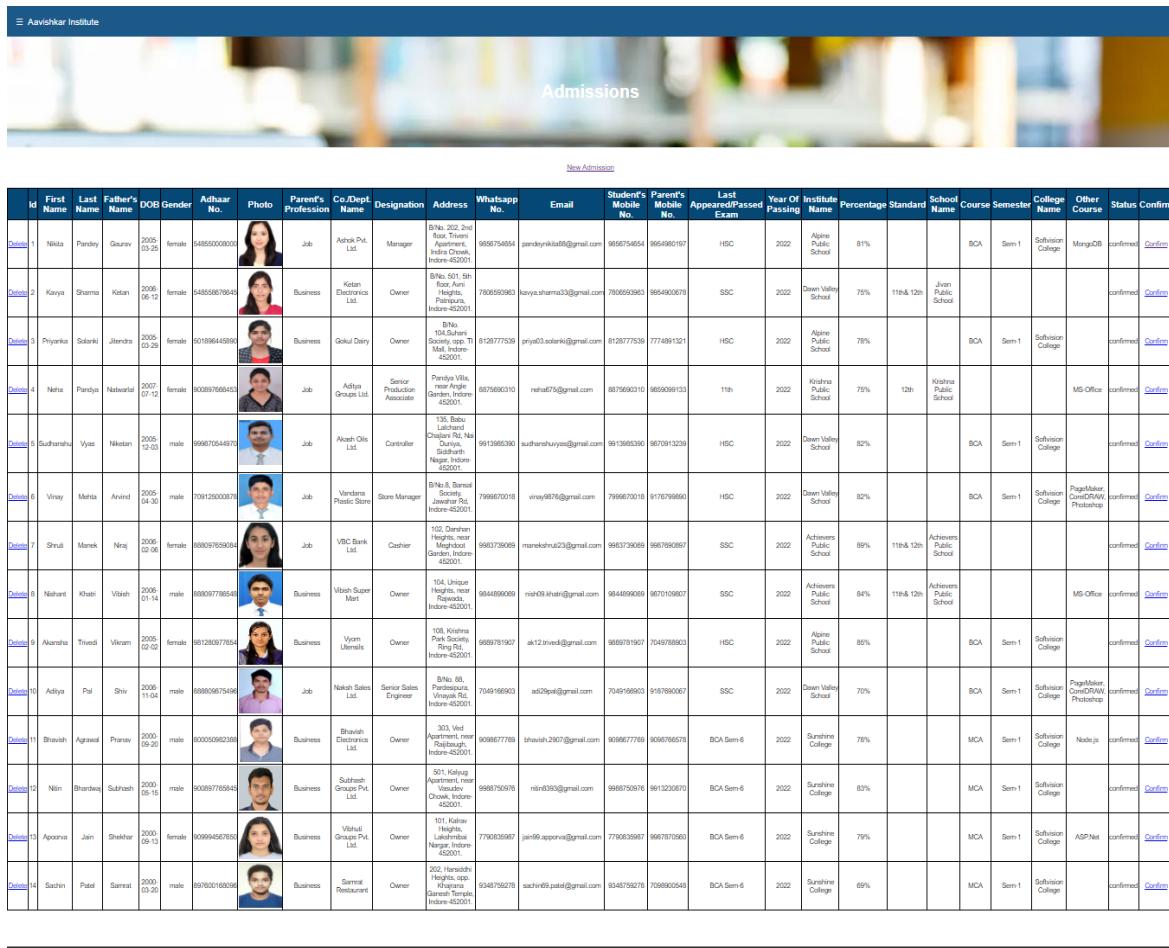


On header ,there's a menu button. Admin can open it and go to the particular page.



Developed By: Tanu Sharma

## 2. Admission Page: Here admin can add new student and confirm or delete the admission.



The screenshot shows a web-based admissions management system. At the top, there's a header bar with the Aavishkar Institute logo and a navigation menu. Below the header is a banner with the word "Admissions". The main content area contains a table listing student information. The columns include: Id, First Name, Last Name, Father's Name, DOB, Gender, Adhaar No., Photo, Parent's Profession, Co/Dept Name, Designation, Address, WhatsApp No., Email, Student's Mobile No., Parent's Mobile No., Last Appeared/Passed Exam, Year Of Passing, Institute Name, Percentage, Standard, School Name, Course, Semester, College Name, Other Course, Status, and Confirm. Each row represents a student record with their details filled in.

	<b>ID</b>	<b>First Name</b>	<b>Last Name</b>	<b>Father's Name</b>	<b>DOB</b>	<b>Gender</b>	<b>Adhaar No.</b>	<b>Photo</b>	<b>Parent's Profession</b>	<b>Co/Dept Name</b>	<b>Designation</b>	<b>Address</b>	<b>WhatsApp No.</b>	<b>Email</b>	<b>Student's Mobile No.</b>	<b>Parent's Mobile No.</b>	<b>Last Appeared/Passed Exam</b>	<b>Year Of Passing</b>	<b>Institute Name</b>	<b>Percentage</b>	<b>Standard</b>	<b>School Name</b>	<b>Course</b>	<b>Semester</b>	<b>College Name</b>	<b>Other Course</b>	<b>Status</b>	<b>Confirm</b>
1	1	Nikita	Pandey	Gaurav	2005-03-25	female	5489500000000000		Job	Akash Pvt. Ltd	Manager	B/Ho. 222, 2nd Floor, Apartment, Sector-10, Noida, Indore-452001	9858754654	pandeynikita8@gmail.com	9858754654	9954900197	HSC	2022	Alpine Public School	81%			BCA	Sem-1	Sofvision College	MongoDB	confirmed	<a href="#">Confirm</a>
2	2	Kavya	Sharma	Ketan	2008-06-12	female	5489500079542		Business	Ketan Electronics Ltd	Owner	B/Ho. 501, 5th floor, Auri Residency, Patel Nagar, Patparganj, New Delhi-110026	7806959503	kavya.sharma03@gmail.com	7806959503	9954900678	SSC	2022	Dawn Valley School	75%	11th&12th	Jain Public School					confirmed	<a href="#">Confirm</a>
3	3	Priyanka	Sohanki	Jitendra	2005-03-26	female	50196844599		Business	Gokul Dairy	Owner	104, Saheli Society, opp. T.I. College, Sector-10, Noida, Indore-452001	9128777539	priy03.sohanki@gmail.com	9128777539	7774891321	HSC	2022	Alpine Public School	78%			BCA	Sem-1	Sofvision College		confirmed	<a href="#">Confirm</a>
4	4	Neha	Pandya	Natwarlal	2007-07-12	female	60099766643		Job	Aditya Groups Ltd	Senior Production Associate	Punjabi Villa, near Angle Garden, Indore-452001	8875690310	nehat75@gmail.com	8875690310	9859009133	11th	2022	Krishna Public School	75%	12th	Krishna Public School				MS-Office	confirmed	<a href="#">Confirm</a>
5	5	Sucharshu	Vyas	Nikatan	2005-12-03	male	69980754497		Job	Akash Oils Ltd	Controller	135, Babu Latchand Bhagat Rd, Opp. Duniya, Sudharth Nath, Sector-10, Noida, Indore-452001	9913685390	sudharshuyas@gmail.com	9913685390	9670913239	HSC	2022	Dawn Valley School	82%			BCA	Sem-1	Sofvision College		confirmed	<a href="#">Confirm</a>
6	6	Vineay	Maria	Anvind	2005-04-30	male	70912800027		Job	Vardena Plastic Store	Store Manager	B/Ho. 8, Bansil Society, Janta Rd, Indore-452001	7999870218	vineay0707@gmail.com	7999870218	9178796960	HSC	2022	Dawn Valley School	82%			BCA	Sem-1	Sofvision College	PageMaker, CorelDRAW, Photoshop	confirmed	<a href="#">Confirm</a>
7	7	Shruti	Manek	Niraj	2008-02-06	female	69909765904		Job	VIBC Bank Ltd	Cashier	102, Deenbaran Heights, Sector-10, Noida, Indore-452001	9860796049	manek01m22@gmail.com	9860796049	9860796049	SSC	2022	Achievers Public School	86%	11th&12th	Achievers Public School					confirmed	<a href="#">Confirm</a>
8	8	Nishant	Khetri	Vtash	2008-01-14	male	69909778954		Business	Vtash Super Mart	Owner	104, Unique Height, Sector-10, Noida, Indore-452001	9844899009	nish09.khetri@gmail.com	9844899009	9670109907	SSC	2022	Achievers Public School	84%	11th&12th	Achievers Public School				MS-Office	confirmed	<a href="#">Confirm</a>
9	9	Akansha	Tripathi	Vikram	2005-02-02	female	68126977654		Business	Vijay Utensils	Owner	108, Ananya Park Society, Ring Rd, Indore-452001	9885781907	ak12.ved@gmail.com	9885781907	7049786900	HSC	2022	Alpine Public School	85%			BCA	Sem-1	Sofvision College		confirmed	<a href="#">Confirm</a>
10	10	Aditya	Patel	Shiv	2008-11-04	male	68880967549		Job	Naksh Sales Ltd	Senior Sales Engineer	B/Ho. 88, Panchayat Bhawan, Vinayak Rd, Indore-452001	7049166903	ad12patel@gmail.com	7049166903	9157890067	SSC	2022	Dawn Valley School	70%			BCA	Sem-1	Sofvision College	PageMaker, CorelDRAW, Photoshop	confirmed	<a href="#">Confirm</a>
11	11	Bhavish	Agrawal	Pranav	2000-09-20	male	80005098209		Business	Bhavish Electronics Ltd	Owner	303, Verdi Apartment, near Patel Nagar, Sector-10, Noida, Indore-452001	9098677789	bhavish2907@gmail.com	9098677789	9060766578	BCA Sem-6	2022	Sunshine College	78%			MCA	Sem-1	Sofvision College	Node.js	confirmed	<a href="#">Confirm</a>
12	12	Nitin	Bhandarkar	Subhash	2000-05-15	male	60089776545		Business	Subhash Groups Pvt. Ltd.	Owner	501, Mahaveer Apartments, Chaudhary Chandra Shekhar Marg, Sector-10, Noida, Indore-452001	99687505976	nitin393@gmail.com	99687505976	9913230870	BCA Sem-6	2022	Sunshine College	85%			MCA	Sem-1	Sofvision College		confirmed	<a href="#">Confirm</a>
13	13	Aparna	Jain	Shweta	2000-09-13	female	60999456769		Business	Vishal Groups Pvt. Ltd.	Owner	101, Kalika Heights, Mayur Vihar, Phase-1, New Delhi-110061	7790635987	parn69.approve@gmail.com	7790635987	9967870660	HSC	2022	Sunshine College	79%			MCA	Sem-1	Sofvision College	ASP-.Net	confirmed	<a href="#">Confirm</a>
14	14	Sachin	Patel	Samrat	2000-03-20	male	60160016634		Business	Sarmat Restaurant	Owner	202, Hemlata Heights, opp. Aranya Apartments, Gomti Nagar, Sector-10, Noida, Indore-452001	9349795076	sachin91patel@gmail.com	9349795076	7038900545	BCA Sem-6	2022	Sunshine College	69%			MCA	Sem-1	Sofvision College		confirmed	<a href="#">Confirm</a>

Developed By : Tanu Sharma

Developed By: Tanu Sharma

3. Add New Student Page: By clicking on Add New Student an admission form is displayed where admin can add a new student.

**Admission Form**

First Name

Father's Name

Gender  Male  Female  Other

Adhaar No.

Company/ Dept. Name

Address

Whatsapp No.

Student's Mobile No.

Last(Appeared/Passed Exam)

Name of Institute

**Commerce**

Standard  11<sup>th</sup> & 12<sup>th</sup>  12<sup>th</sup>

School Name

Other Course

Modes of Payment: Cash and Cheque

**Computer**

Course  BCA  MCA

Semester

College Name

Email

Parent's Mobile No.

Year of Passing

Percentage

Submit

Developed By : Tanu Sharma

#### 4. Courses Page: Here admin can view, add or delete courses.

The screenshot shows a web application interface for managing courses. At the top, there's a header bar with the Aavishkar Institute logo. Below it, the main title is "Courses". In the center, there's a search/filter form with fields for Category (dropdown), Course (text input), Fees (text input), Duration (text input), and an "Add Course" button. Below the form is a table listing 24 courses. The table has columns for Id, Category, Course, Fees, and Duration. Each row contains a "Delete" link. The categories listed include HSC (General), UG, PG, Basic Course, Graphics Designing, Web Designing, Programming Language, Database, and Web Programming. The fees range from 3,000 to 12,000, and durations range from 1 month to 3 years.

	<b>Id</b>	<b>Category</b>	<b>Course</b>	<b>Fees</b>	<b>Duration</b>
<a href="#">Delete</a>	1	HSC (General)	11th & 12th	40,000	2 years
<a href="#">Delete</a>	2	UG	BCA	90,000	3 years
<a href="#">Delete</a>	3	PG	MCA	12,000	2 years
<a href="#">Delete</a>	4	Basic Course	MS-Office	8,000	2 months
<a href="#">Delete</a>	5	Graphics Designing	PageMaker, CorelDRAW, Photoshop	10,000	6 months
<a href="#">Delete</a>	6	Web Designing	HTML	3,000	1 month
<a href="#">Delete</a>	7	Web Designing	CSS	8,000	5 months
<a href="#">Delete</a>	8	Web Designing	JavaScript	10,000	2 months
<a href="#">Delete</a>	9	Programming Language	C Language	4,000	1.5 months
<a href="#">Delete</a>	10	Programming Language	C++	5,000	2 months
<a href="#">Delete</a>	11	Programming Language	C#	4,000	1 month
<a href="#">Delete</a>	12	Programming Language	Java	8,000	4 months
<a href="#">Delete</a>	13	Programming Language	Python	2,000/pm	3-6 months
<a href="#">Delete</a>	14	Programming Language	Android	7,000	3 months
<a href="#">Delete</a>	15	Programming Language	Node.js	6,000	3 months
<a href="#">Delete</a>	16	Programming Language	React.js	8,000	4 months
<a href="#">Delete</a>	17	Programming Language	Angular.js	6,000	2.5 months
<a href="#">Delete</a>	18	Database	Oracle	5,000	1 month
<a href="#">Delete</a>	19	Database	MySQL	5,000	2-3 months
<a href="#">Delete</a>	20	Database	SQL Server	5,000	1.5 months
<a href="#">Delete</a>	21	Database	MongoDB	5,000	1 month
<a href="#">Delete</a>	22	Web Programming	PHP with MySQL	6,000	2-3 months
<a href="#">Delete</a>	23	Web Programming	ASP.Net	5,000	1 month
<a href="#">Delete</a>	24	Web Programming	JSP	5,000	1 month

Developed By : Tanu Sharma

## 5. Users Page: Here admin can view and delete users.

	<b>Id</b>	<b>Name</b>	<b>Email</b>	<b>Mobile No.</b>	<b>Usertype</b>
<a href="#">Delete</a>	1	Tanu Sharma	sharma11.ts@gmail.com	9996348876	admin
<a href="#">Delete</a>	2	Nikita Pandey	pandeynikita88@gmail.com	9856754654	user
<a href="#">Delete</a>	3	Akansha Trivedi	ak12.trivedi@gmail.com	9889781907	user
<a href="#">Delete</a>	4	Priyanka Solanki	priya03.solanki@gmail.com	8128777539	user
<a href="#">Delete</a>	5	Neha Pandya	neha675@gmail.com	8875690310	user
<a href="#">Delete</a>	6	Sudhanshu Vyas	sudhanshuvyas@gmail.com	9913985390	user
<a href="#">Delete</a>	7	Vinay Mehta	vinay9876@gmail.com	7999870018	user
<a href="#">Delete</a>	8	Shruti Manek	manekshruti23@gmail.com	9983739069	user
<a href="#">Delete</a>	9	Nishant Khatri	nish09.khatri@gmail.com	9844899069	user
<a href="#">Delete</a>	10	kavya Sharma	kavya.sharma33@gmail.com	7806593963	user
<a href="#">Delete</a>	11	Aditya Pal	adi29pal@gmail.com	7049166903	user
<a href="#">Delete</a>	12	Bhavish Agrawal	bhavish.2907@gmail.com	9098677769	user
<a href="#">Delete</a>	13	Nitin Bhardwaj	nitin8393@gmail.com	9988750976	user
<a href="#">Delete</a>	14	Apoorva Jain	jain99.apoorva@gmail.com	7790835987	user
<a href="#">Delete</a>	15	Sachin Patel	sachin69.patel@gmail.com	9348759278	user

Developed By : Tanu Sharma

**6. Messages Page:** Here all messages will be displayed that are sent by admin to user and also admin can send new messages.



<b>Id</b>	<b>Subject</b>	<b>Message</b>	<b>Date</b>
1	Republic Day Holiday	Tomorrow is Republic Day, so there will be a holiday.	2023-01-26
2	Seminar	Tomorrow there will be a motivational seminar for everyone.	2023-01-28
3	Mahashivratri Holiday	Tomorrow there will be a holiday for Mahashivratri Holiday	2023-02-18
4	Webinar	Tomorrow there will be a webinar for all of you. Kindly visit the link <a href="https://tech.webinar.com">https://tech.webinar.com</a> tomorrow.	2023-02-19
5	Reading holiday	We would like to inform you that there will be a 8 days reading holiday for all of you starting from February 21nd to February 28th. Regular classes will resume on March 1st. Thank you for your attention, and we wish you happy reading!	2023-02-20

Developed By : Tanu Sharma

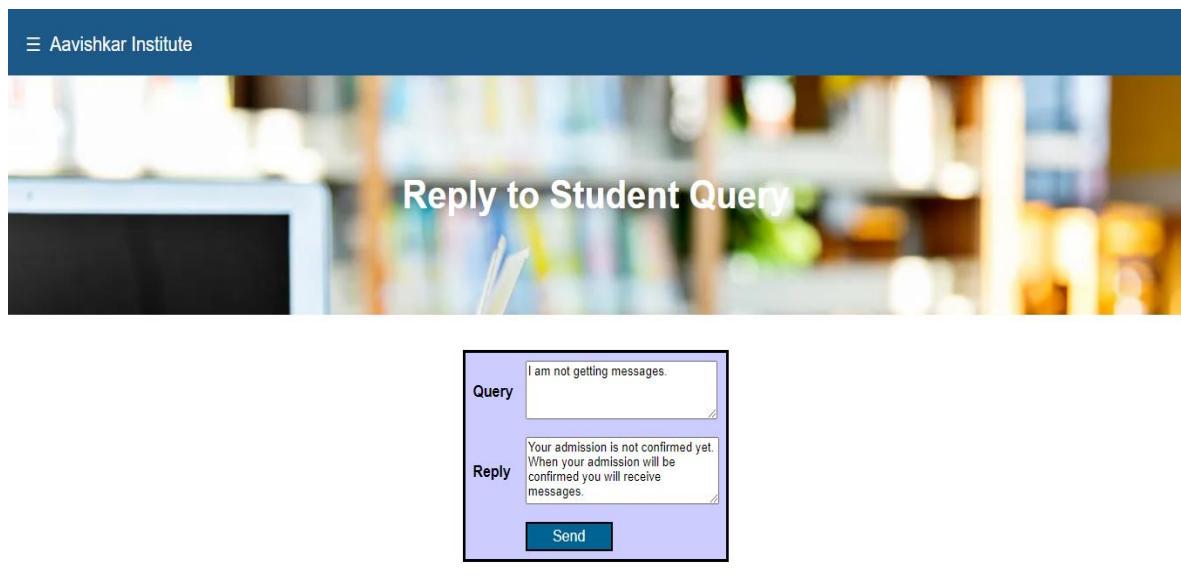
7. Student Query Page: Here admin can view queries asked by students and can also reply to them.

The screenshot shows a web application titled "Student Query". At the top, there is a navigation bar with the text "Aavishkar Institute". Below the title, there is a blurred background image of a library or study area. The main content area has a table with the following columns: Id, Name, Email, Mno, Query, Reply, and Reply Query. The table contains five rows of data:

Id	Name	Email	Mno	Query	Reply	Reply Query
1	Kavya Sharma	kavya.sharma33@gmail.com	7806593963	I am not getting messages.	Your admission is not confirmed yet. When your admission will be confirmed you will receive messages.	<a href="#">Reply</a>
2	Apoorva Jain	jain99.apoorva@gmail.com	7790835987	I am learning ASP.Net, can I also apply for another course?	Yes, you can apply.	<a href="#">Reply</a>
3	Vinay Mehta	vinay9876@gmail.com	7999870018	I have doubts in communication skills, so can I come to understand them this evening?	Yes, you can come.	<a href="#">Reply</a>
4	Sachin Patel	sachin69.patel@gmail.com	9348759278	I went out today with my family, so can I submit my assignments tomorrow.	pending	<a href="#">Reply</a>
5	Priyanka Solanki	priya03.solanki@gmail.com	8128777539	My parents want to meet Mr. Pravin Varma, so can we come this evening?	pending	<a href="#">Reply</a>

Developed By : Tanu Sharma

8. Reply To Student Query Page: By clicking on reply a reply to student query page is displayed where admin can reply to student's query.



Developed By : Tanu Sharma

**9. Leave Application Page:** Here admin can view all leave applications submitted by students and can grant them.



<b>Id</b>	<b>Name</b>	<b>Email</b>	<b>Mobile No.</b>	<b>Subject</b>	<b>From Date</b>	<b>To Date</b>	<b>Leave Reason</b>	<b>Status</b>	<b>Grant</b>
1	Neha Pandya	neha675@gmail.com	8875690310	Leave application for a week.	29-08-2022	05-01-2023	My grandmother is not well from last few days and my family has planned a visit to our hometown Kanpur for a week. So, I won't be able to attend classes. Thus, I request you to grant 8 days off from institute.	granted	<a href="#">Grant</a>
2	Bhavish Agrawal	bhavish.2907@gmail.com	9098677769	Application for sick leave.	09-01-2023	11-01-2023	I am not well because of fever. I request for leave of 3 days, or till recovery. I will update you in any other case.	granted	<a href="#">Grant</a>
3	Nishant Khatri	nish09.khatri@gmail.com	9844899069	Sick leave application.	12-01-2023	15-01-2023	I am extremely ill since last two days and doctor diagnosed malaria. Due to this condition of mine, I am not be able to attend classes till I get recover. Kindly grant me leave for 4 days.	granted	<a href="#">Grant</a>
4	Apoorva Jain	jain99.apoorva@gmail.com	7790835987	Application for leave in institute.	18-01-2023	22-01-2023	I have my cousin's marriage to attend which is scheduled in Delhi. Therefore, I am unable to attend classes. I request you to Kindly grant me leave for 5 days so I can attend my family function.	granted	<a href="#">Grant</a>
5	Shruti Manek	manekshruti23@gmail.com	9983739069	Leave application for personal reason.	06-02-2023	08-02-2023	I will not be able to attend classes because of urgent work at home. Therefore, I request to kindly grant me leave for 3 days.	granted	<a href="#">Grant</a>

Developed By : Tanu Sharma

10. PDF Page: Here admin can view, delete and send pdf to students.

	<b>Id</b>	<b>Course</b>	<b>Semester</b>	<b>Subject</b>	<b>Pdf</b>	<b>Path</b>
<a href="#">Delete</a>	1	BCA	Sem-1	C Language	C Language.pdf	D:\BCA\Sem-6 Project\pdf\C Language.pdf
<a href="#">Delete</a>	2	BCA	Sem-1	Computer Fundamentals	Computer Fundamentals.pdf	D:\BCA\Sem-6 Project\pdf\Computer Fundamentals.pdf
<a href="#">Delete</a>	3	BCA	Sem-1	Communication Skills	Communication Skills.pdf	D:\BCA\Sem-6 Project\pdf\Communication Skills.pdf
<a href="#">Delete</a>	4	BCA	Sem-1	Networking Fundamentals	Networking Fundamentals.pdf	D:\BCA\Sem-6 Project\pdf\Networking Fundamentals.pdf
<a href="#">Delete</a>	5	MCA	Sem-1	Computer Organization and Architecture	Computer Organization and Architecture.pdf	D:\BCA\Sem-6 Project\pdf\Computer Organization and Architecture.pdf
<a href="#">Delete</a>	6	MCA	Sem-1	Probability and Statistical Methods	Probability and Statistical Methods.pdf	D:\BCA\Sem-6 Project\pdf\Probability and Statistical Methods.pdf

Developed By : Tanu Sharma

Developed By: Tanu Sharma

## 11. Feedback Page: Here admin can view and delete feedbacks.

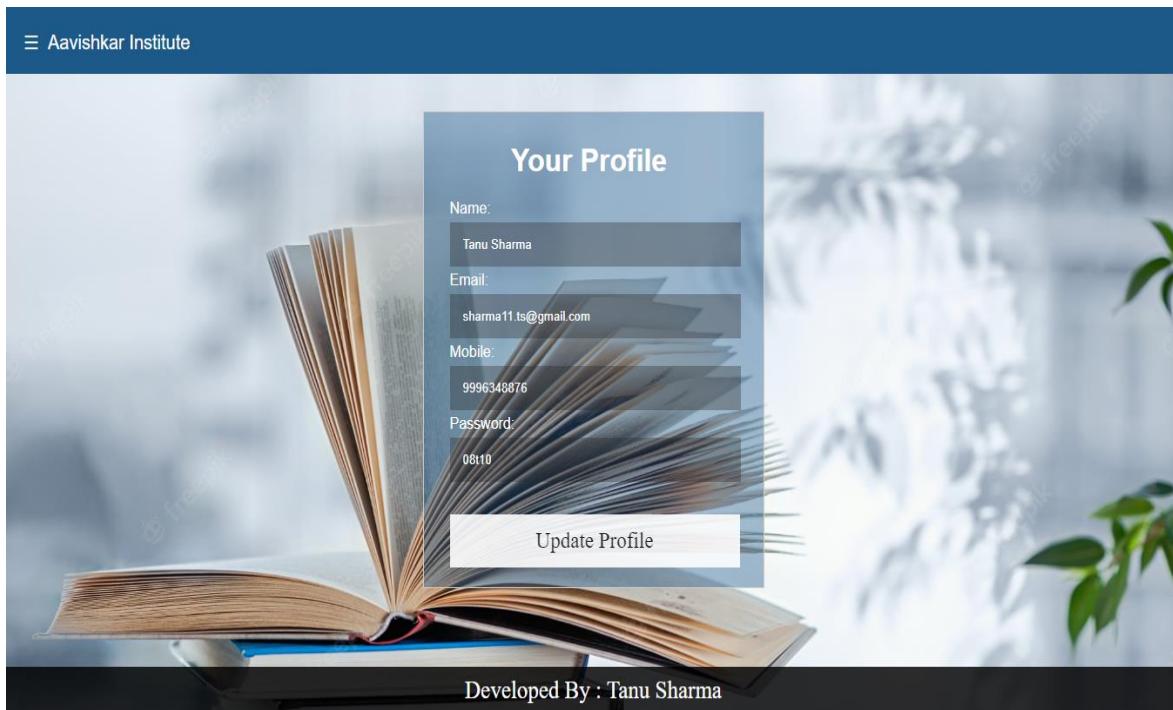


The screenshot shows a web application for managing feedback. At the top, there's a blue header bar with the text "Aavishkar Institute". Below the header, the word "Feedback" is prominently displayed in large white letters. The main content area is a table listing 10 feedback entries. Each entry includes columns for Id, Name, Email, Mobile No., and Feedback. The "Feedback" column contains short snippets of positive comments from students. Each row also features a "Delete" link under the "Id" column.

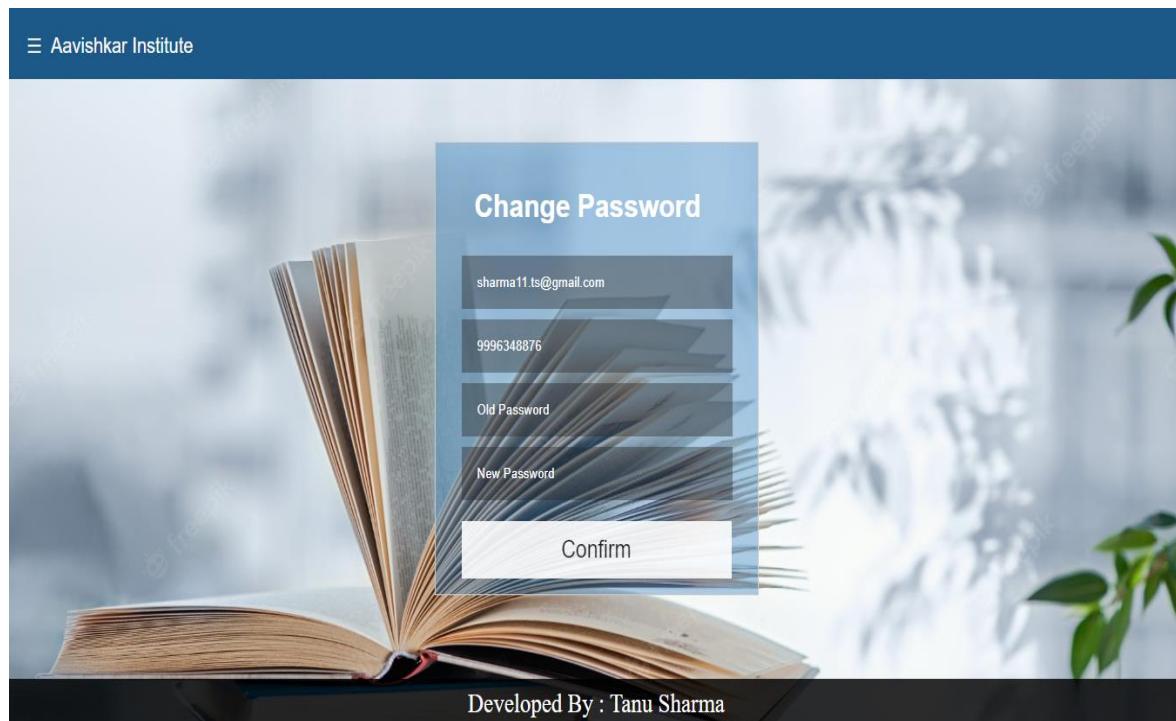
	<b>Id</b>	<b>Name</b>	<b>Email</b>	<b>Mobile No.</b>	<b>Feedback</b>
<a href="#">Delete</a>	1	Akansha Trivedi	ak12.trivedi@gmail.com	9889781907	The institute is awesome. I am so impressed with the institute.
<a href="#">Delete</a>	2	Sudhanshu Vyas	sudhanshuvyas@gmail.com	9913985390	The faculties here are so good and supportive.
<a href="#">Delete</a>	3	Aditya Pal	adi29pal@gmail.com	7049166903	A fantastic institute! The faculties are very good...
<a href="#">Delete</a>	4	Vinay Mehta	vinay9876@gmail.com	7999870018	It is one of the best institute.
<a href="#">Delete</a>	5	Bhavish Agrawal	bhavish 2907@gmail.com	9098677769	The teachers are very friendly. They are just wonderful.
<a href="#">Delete</a>	6	Nikita Pandey	pandeynikita88@gmail.com	9856754654	Good faculty, excellent teaching, good education environment.
<a href="#">Delete</a>	7	Nishant Khatri	nish09.khatri@gmail.com	9844899069	Best institute with best faculty.
<a href="#">Delete</a>	8	Shruti Manek	manekshruti23@gmail.com	9983739069	The teachers are amazing, we learn so much from them.
<a href="#">Delete</a>	9	Neha Pandya	neha675@gmail.com	8875690310	The teachers are amazing, we learn so much from them.
<a href="#">Delete</a>	10	Nitin Bhardwaj	nitin8393@gmail.com	9988750976	Very nice experience being here...quality of teaching is just wonderful.

Developed By : Tanu Sharma

## 12. Profile Page: Here admin can view and update his profile.



13. Change Password Page: Here admin can easily change password by providing required details.



## 4.6 Source Code

➤ Visitor Side Code:

✓ Connection String:

- ```
<connectionStrings>
    <add name="cn" connectionString="Data
        Source=(LocalDB)\v11.0;AttachDbFilename='D:\BCA\Sem-6
        Project\App_Data\aaavishkar_institute.mdf';Integrated
        Security=True" providerName="System.Data.SqlClient"/>
</connectionStrings>
```

1. Sign Up File:

- ```
protected void signin_Click (object sender, EventArgs e)
{
    string str =
WebConfigurationManager.ConnectionStrings["cn"].ToString();

    string check = "select mno from user_login where email =
@email or mno = @mno or password = @password";
    string query = "insert into user_login (name, email, mno,
password, usertype) values (@name, @email, @mno, @password,
'user')";

    using (SqlConnection cn = new SqlConnection(str))
    {
        using (SqlCommand cmd = new SqlCommand(check, cn))
        {
            cmd.Parameters.AddWithValue("@email", email.Text);
            cmd.Parameters.AddWithValue("@mno", mno.Text);
            cmd.Parameters.AddWithValue("@password",
password.Text);

            cn.Open();
        }
    }
}
```

```
using (SqlDataReader dr = cmd.ExecuteReader())
{
    if (dr.HasRows)
    {
        Response.Write("<script>alert('The user
already exists')</script>");
    }
    else
    {
        dr.Close();

        using (SqlCommand sqlcmd = new
SqlCommand(query, cn))
        {

            sqlcmd.Parameters.AddWithValue("@name",
name.Text);

            sqlcmd.Parameters.AddWithValue("@email"
, email.Text);
            sqlcmd.Parameters.AddWithValue("@mno",
mno.Text);
            sqlcmd.Parameters.AddWithValue("@passwo
rd", password.Text);

            sqlcmd.ExecuteNonQuery();
            Response.Write("<script>alert('Register
ed as User...')</script>");
            Response.Write("<script>setTimeout(func
tion() { window.location.href =
'login.aspx' }, 0000);</script>");
        }
    }
}

cn.Close();
}
```

➤ User Side Code:

1. Login File:

```

• protected void login_Click(object sender, EventArgs e)
{
    string str =
WebConfigurationManager.ConnectionStrings["cn"].ToString();

    using (SqlConnection cn = new SqlConnection(str))
    {
        string query = "select * from user_login where mno =
@mno and password = @password";

        using (SqlCommand cmd = new SqlCommand(query, cn))
        {
            cmd.Parameters.AddWithValue("@mno", mno.Text);
            cmd.Parameters.AddWithValue("@password",
password.Text);

            cn.Open();

            using (SqlDataReader reader = cmd.ExecuteReader())
            {
                if (reader.HasRows)
                {
                    reader.Read();

                    if (reader["usertype"].ToString() ==
"admin")
                    {
                        Session["admin"] = mno.Text;

                        Response.Redirect("admin/Default.aspx");
                    }
                }
                else if (reader["usertype"].ToString() ==
"user")
                {
                    Session["user"] = mno.Text;

                    Response.Redirect("user/Default.aspx");
                }
            }
        else
        {
    
```

```
        Response.Write("<script>alert('Please enter  
a valid mobile no. or password')</script>");  
    }  
}  
cn.Close();  
}  
}  
}
```

## 2. Forgot Password File:

- ```
• protected void confirmmpswd_Click(object sender, EventArgs e)
{
    string str =
    WebConfigurationManager.ConnectionStrings["cn"].ToString();
    string mno = this.mno.Text;
    string email = this.email.Text;
    string password = this.password.Text;
    string confirmpassword = this.confirmpassword.Text;

    using (SqlConnection cn = new SqlConnection(str))
    {
        using (SqlCommand cmd = new SqlCommand())
        {
            cmd.Connection = cn;
            cmd.CommandType = CommandType.Text;

            cmd.CommandText = "select mno from user_login
where email = @email and mno = @mno";

            cmd.Parameters.AddWithValue("@email", email);
            cmd.Parameters.AddWithValue("@mno", mno);

            cn.Open();
            SqlDataReader reader = cmd.ExecuteReader();

            if (reader.HasRows)
            {
                reader.Close();

                cmd.CommandText = "update user_login set
password = @password where mno = @mno";
                cmd.Parameters.AddWithValue("@password",
password);
```

```
cmd.ExecuteNonQuery();

        Response.Write("<script>alert('Your Password  
Updated Successfully...')</script>");
        Response.Write("<script>setTimeout(function()  
{ window.location.href = 'login.aspx' },  
0000);</script>");
    }
    else
    {
        Response.Write("<script>alert('Please enter  
a valid email or mobile no.')</script>");
    }
    cn.Close();
}
}
```

### 3. Admission File:

- ```
• protected void Page_Load(object sender, EventArgs e)
{
    othercourse.DataSource = SqlDataSource1;
    othercourse.DataBind();
}
protected void submit_Click(object sender, EventArgs e)
{
    string str =
    WebConfigurationManager.ConnectionStrings["cn"].ToString()
    ;

    string check = "SELECT adhaarno FROM admission WHERE
adhaarno=@adhaarno";
    string query = "INSERT INTO
admission(fnm,lnm,fathernm,dob,gender,adhaarno,photo,paren
tprof,codept,designation,address,whatsappno,email,studmno,
pmno,lastexam,yrofpassing,instrnm,pr,stud,schoolnm,course,se
m,collegenm,othercourse,status)
VALUES(@fnm,@lnm,@fathernm,@dob,@gender,@adhaarno,@photo,@
parentprof,@codept,@designation,@address,@whatsappno,@ema
il,@studmno,@pmno,@lastexam,@yrofpassing,@instrnm,@pr,@std,@
schoolnm,@course,@sem,@collegenm,@othercourse,'pending')";

    using (SqlConnection cn = new SqlConnection(str))
    {
        using (SqlCommand sqlcmd = new SqlCommand(check, cn))
        {
            cn.Open();
            sqlcmd.ExecuteNonQuery();
        }
    }
}
```

```
sqlcmd.Parameters.AddWithValue("@adhaarno",
adhaarno.Text);

cn.Open();

using (SqlDataReader dr = sqlcmd.ExecuteReader())
{
    if (dr.HasRows)
    {
        Response.Write("<script>alert('The adhaarno. already exists')</script>");
    }
    else
    {
        dr.Close();

        using (SqlCommand cmd = new
SqlCommand(query, cn))
        {
            cmd.Parameters.AddWithValue("@fnm",
fnm.Text);
            cmd.Parameters.AddWithValue("@lnm",
lnm.Text);
            cmd.Parameters.AddWithValue("@fathernm",
fathernm.Text);
            cmd.Parameters.AddWithValue("@dob",
dob.Text);
            cmd.Parameters.AddWithValue("@gender",
gender.SelectedValue);
            cmd.Parameters.AddWithValue("@adhaarno",
adhaarno.Text);
            cmd.Parameters.AddWithValue("@photo",
Path.GetFileName(photo.PostedFile.FileName));
            cmd.Parameters.AddWithValue("@parentprof",
parentprof.Text);
            cmd.Parameters.AddWithValue("@codept",
codept.Text);
            cmd.Parameters.AddWithValue("@designation",
designation.Text);
            cmd.Parameters.AddWithValue("@address",
address.Text);
            cmd.Parameters.AddWithValue("@whatsappno",
whatsappno.Text);
            cmd.Parameters.AddWithValue("@email",
email.Text);
            cmd.Parameters.AddWithValue("@studmno",
studmno.Text);
        }
    }
}
```

```
        cmd.Parameters.AddWithValue("@pmno",  
        pmno.Text);  
        cmd.Parameters.AddWithValue("@lastexam"  
        , lastexam.Text);  
        cmd.Parameters.AddWithValue("@yrofpassing",  
        yrofpassing.Text);  
        cmd.Parameters.AddWithValue("@instnm",  
        instnm.Text);  
        cmd.Parameters.AddWithValue("@pr",  
        pr.Text);  
        cmd.Parameters.AddWithValue("@std",  
        std.Text);  
        cmd.Parameters.AddWithValue("@schoolnm",  
        schoolnm.Text);  
        cmd.Parameters.AddWithValue("@course",  
        course.SelectedValue);  
        cmd.Parameters.AddWithValue("@sem",  
        sem.SelectedValue);  
        cmd.Parameters.AddWithValue("@collegenm",  
        collegenm.Text);  
        cmd.Parameters.AddWithValue("@othercourse",  
        othercourse.SelectedValue);  
        cmd.ExecuteNonQuery();  
  
        string filePath =  
        Server.MapPath("/images/") +  
        Path.GetFileName(photo.PostedFile.FileName);  
        photo.SaveAs(filePath);  
  
        Response.Write("<script>alert('Applied  
        for admission  
        successfully...')</script>");  
  
        Response.Write("<script>window.location  
        ='messages.aspx'</script>");  
    }  
}
```

#### 4. Courses File:

- ```
<asp:GridView ID="GridView1" runat="server"
    AutoGenerateColumns="False" DataSourceID="SqlDataSource1"
    CellPadding="10" Width="100%" HeaderStyle-CssClass="h"
    RowStyle-CssClass="r">
    <Columns>
        <asp:BoundField DataField="category"
            HeaderText="Category" SortExpression="category" />
        <asp:BoundField DataField="course"
            HeaderText="Course" SortExpression="course" />
        <asp:BoundField DataField="fees" HeaderText="Fees"
            SortExpression="fees" />
        <asp:BoundField DataField="duration"
            HeaderText="Duration" SortExpression="duration" />
    </Columns>
</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:cn %>"
    SelectCommand="SELECT [category], [course], [fees],
    [duration] FROM [courses]"></asp:SqlDataSource>
```

#### 5. Messages File:

- ```
<div>
    <h1>Messages</h1>
    <div class="container">
        <asp:Repeater ID="admission" runat="server"
            DataSourceID="SqlDataSource4">
            <ItemTemplate>
                <%#Eval("fnm") %>
                <%#Eval("fathernm") %>
                <%#Eval("lnm") %>
                your admission is <%#Eval("status") %>
            </ItemTemplate>
        </asp:Repeater>
    <asp:SqlDataSource ID="SqlDataSource4"
        runat="server"
        ConnectionString="<%$ ConnectionStrings:cn %>"
        SelectCommand="SELECT [fnm], [lnm], [fathernm],
        [status] FROM [admission] WHERE (([pmno] = @mno)
        OR ([studmno] = @mno))">
        <SelectParameters>
            <asp:SessionParameter Name="mno"
                SessionField="user" Type="String" />
        </SelectParameters>
    </asp:SqlDataSource>
</div>
```

```

</asp:SqlDataSource>
</div>

<div class="container">
<asp:Repeater ID="studquery" runat="server"
DataSourceID="SqlDataSource3">
    <ItemTemplate>
        Your Query: <%#Eval("query") %><br />
        Answer:<%#Eval("reply") %><br />
    </ItemTemplate>
</asp:Repeater>
<asp:SqlDataSource ID="SqlDataSource3"
runat="server"
ConnectionString="<%$ ConnectionStrings:cn %>"
SelectCommand="SELECT [query], [reply] FROM
[studquery] WHERE ([mno] = @mno)">
    <SelectParameters>
        <asp:SessionParameter Name="mno"
            SessionField="user" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
</div>

<div class="container">
<asp:Repeater ID="lvapp" runat="server"
DataSourceID="SqlDataSource1">
    <ItemTemplate>
        <%# Eval("name") %> Your Leave is <%#
        Eval("status") %>.
    </ItemTemplate>
</asp:Repeater>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server"
ConnectionString="<%$ ConnectionStrings:cn %>"
SelectCommand="SELECT [name], [status] FROM
[lv_app] WHERE ([mno] = @mno)">
    <SelectParameters>
        <asp:SessionParameter Name="mno"
            SessionField="user" Type="String" />
    </SelectParameters>
</asp:SqlDataSource>
</div>

<div class="container">
<asp:Repeater ID="message" runat="server"
DataSourceID="SqlDataSource2">
    <ItemTemplate>

```

```

        <b>Date:</b> <%#Eval("Date") %><br />
        <b>Subject:</b>
        <%#Eval("subject") %><br />
        <b>Message:</b>
        <%#Eval("message") %><br />
        <br /><br />
    </ItemTemplate>
</asp:Repeater>
<asp:SqlDataSource ID="SqlDataSource2"
runat="server"
ConnectionString="<%$ ConnectionStrings:cn %>"
SelectCommand="SELECT [date], [subject], [message]
FROM [messages]"></asp:SqlDataSource>
</div>

<div class="container">
<asp:Repeater ID="pdf" runat="server"
DataSourceID="SqlDataSource5">
    <ItemTemplate>
        <b>Course:</b> <%#Eval("course") %><br />
        <b>Semester:</b> <%#Eval("sem") %><br />
        <b>Subject:</b> <%#Eval("subject") %><br />
        <br />
        <asp:Label ID="pdf1" runat="server"
Text="PDF: " Font-Bold="True" Font-
Size="Medium"></asp:Label>
        <asp:Label ID="pdf" runat="server"
Text='<%#Eval("pdf") %>' 
ReadOnly="True" Font-Names="Arial"
Font-Size="Medium"></asp:Label>
        <br /><asp:Button ID="Button1"
runat="server" Text="Click To Download"
OnClick="btnDownload_Click"
BackColor="#003366" ForeColor="White"
Height="25" BorderColor="White" Font-
Size="Medium" style="margin-top:5px;">
        </asp:Button>
        <br /><br />
    </ItemTemplate>
</asp:Repeater>
<asp:SqlDataSource ID="SqlDataSource5"
runat="server"
ConnectionString="<%$ ConnectionStrings:cn %"
> SelectCommand="SELECT [course], [sem],
[subject], [pdf] FROM
[pdf]"></asp:SqlDataSource>
</div>
</div>

```

- ```

protected void btnDownload_Click(object sender, EventArgs e)
{
    Button button = (Button)sender;
    RepeaterItem item =
    (RepeaterItem)button.NamingContainer;
    Label pdf = (Label)item.FindControl("pdf");
    string fileName = pdf.Text;
    string connectionString =
    WebConfigurationManager.ConnectionStrings["cn"].ConnectionString;
    string query = "SELECT path FROM pdf WHERE pdf = @pdf";
    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        using (SqlCommand cmd = new SqlCommand(query, conn))
        {
            cmd.Parameters.AddWithValue("@pdf", fileName);
            conn.Open();
            object result = cmd.ExecuteScalar();
            if (result != null)
            {
                string filePath = result.ToString();
                Response.ContentType = "application/pdf";
                Response.AppendHeader("Content-Disposition", "attachment; filename=" + fileName);
                Response.TransmitFile(filePath);
                Response.End();
            }
            Response.Redirect(Request.Url.AbsoluteUri);
        }
    }
}

```

## 6. Student Query File:

- ```

protected void submit_Click(object sender, EventArgs e)
{
    string cn =
    WebConfigurationManager.ConnectionStrings["cn"].ToString();

    using (SqlConnection connection = new SqlConnection(cn))
    {
        connection.Open();

```

```

        string q = "INSERT INTO studquery
(name,email,mno,query,reply) VALUES
(@name,@email,@mno,@query,'')";

        using (SqlCommand command = new SqlCommand(q,
connection))
{
    command.Parameters.AddWithValue("@name",
((TextBox)r1.Items[0].FindControl("name")).Text);
    command.Parameters.AddWithValue("@email",
((TextBox)r1.Items[0].FindControl("email")).Text);
    command.Parameters.AddWithValue("@mno",
((TextBox)r1.Items[0].FindControl("mno")).Text);
    command.Parameters.AddWithValue("@query",
query.Text);
    command.ExecuteNonQuery();
}
Response.Write("<script>alert('Query
Sent...')</script>");
}
Response.Redirect(Request.Url.AbsoluteUri);
}

```

## 7. Leave Application File:

- ```

protected void submit_Click(object sender, EventArgs e)
{
    string cn =
WebConfigurationManager.ConnectionStrings["cn"].ToString();

    using (SqlConnection connection = new SqlConnection(cn))
{
    connection.Open();

    string query = "INSERT INTO lv_app
(name,email,mno,subject,from_dt,to_dt,lv_reason,status)
VALUES
(@name,@email,@mno,@subject,@from_dt,@to_dt,@lv_reason,
'')";

    using (SqlCommand command = new SqlCommand(query,
connection))
{
        command.Parameters.AddWithValue("@name",
((TextBox)r1.Items[0].FindControl("name")).Text);
        command.Parameters.AddWithValue("@email",
((TextBox)r1.Items[0].FindControl("email")).Text);

```

```

        command.Parameters.AddWithValue("@mno",
        ((TextBox)r1.Items[0].FindControl("mno")).Text);
        command.Parameters.AddWithValue("@subject",
        subject.Text);
        command.Parameters.AddWithValue("@from_dt",
        from_dt.Text);
        command.Parameters.AddWithValue("@to_dt", to_dt.Text);
        command.Parameters.AddWithValue("@lv_reason",
        lv_reason.Text);

        command.ExecuteNonQuery();
    }
    Response.Write("<script>alert('Applied For
Leave...')</script>");
}
Response.Redirect(Request.Url.AbsoluteUri);
}

```

## 8. Feedback File:

- ```

protected void submit_Click(object sender, EventArgs e)
{
    string cn =
WebConfigurationManager.ConnectionStrings["cn"].ToString();

    using (SqlConnection connection = new SqlConnection(cn))
    {
        connection.Open();
        string query = "INSERT INTO feedback
(name,email,mno,feedback) VALUES
(@name,@email,@mno,@feedback)";

        using (SqlCommand command = new SqlCommand(query,
connection))
        {
            command.Parameters.AddWithValue("@name",
((TextBox)r1.Items[0].FindControl("name")).Text);
            command.Parameters.AddWithValue("@email",
((TextBox)r1.Items[0].FindControl("email")).Text);
            command.Parameters.AddWithValue("@mno",
((TextBox)r1.Items[0].FindControl("mno")).Text);
            command.Parameters.AddWithValue("@feedback",
feedback.Text.Trim());

            command.ExecuteNonQuery();
        }
    }
}

```

```

        Response.Write("<script>alert('Thank you for your  

feedback...')</script>");
    }
    Response.Redirect(Request.Url.AbsoluteUri);
}

```

## 9. Profile File:

- ```

protected void upd_Click(object sender, EventArgs e)
{
    string str =
WebConfigurationManager.ConnectionStrings["cn"].ToString();
    string name =
((TextBox)profile.Items[0].FindControl("name")).Text;
    string email =
((TextBox)profile.Items[0].FindControl("email")).Text
;

    string query = "UPDATE user_login SET name=@name,
email=@email WHERE mno=@mno";

    using (SqlConnection cn = new SqlConnection(str))
    {
        using (SqlCommand cmd = new
SqlCommand(query, cn))
        {
            cmd.Parameters.AddWithValue("@name",
name);
            cmd.Parameters.AddWithValue("@email",
email);
            cmd.Parameters.AddWithValue("@mno",
Session["user"].ToString());

            cn.Open();
            cmd.ExecuteNonQuery();
            cn.Close();
        }
    }
    Response.Write("<script>alert('Profile Updated  

Successfully....')</script>");
    profile.DataBind();
}

```

## 10. Change Password File:

```
• protected void confirm_Click(object sender, EventArgs e)
{
    string str =
WebConfigurationManager.ConnectionStrings["cn"].ToString();

    string confirmpassword = this.newpassword.Text;

    using (SqlConnection cn = new SqlConnection(str))
    {
        using (SqlCommand cmd = new SqlCommand())
        {
            cmd.Connection = cn;
            cmd.CommandType = CommandType.Text;

            cn.Open();
            SqlDataReader reader = cmd.ExecuteReader();

            if (reader.HasRows)
            {
                reader.Close();

                using (SqlCommand cmd2 = new SqlCommand())
                {
                    cmd2.Connection = cn;
                    cmd2.CommandType = CommandType.Text;

                    cmd2.CommandText = "update user_login set
password = @newpassword where mno = @mno";
                    cmd2.Parameters.AddWithValue("@newpassword",
", this.newpassword.Text);

                    cmd2.ExecuteNonQuery();
                }

                Response.Write("<script>alert('Your Password
Updated Successfully...')</script>");
            }
            else
            {
                Response.Write("<script>alert('Please enter a
valid email or mobile no. or
password')</script>");
            }

            cn.Close();
        }
    }
}
```

## 11. Logout File:

- ```
protected void Page_Load(object sender, EventArgs e)
{
    Session.Clear();
    Session.Abandon();
    Session.RemoveAll();
    Response.Redirect("~/Login.aspx");
}
```

➤ Admin Side Code:

### 1. Admission List File:

- ```
<center><a href="newadmission.aspx">New Admission</a><br /><br /></center><br />

<asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False" DataKeyNames="Id"
DataSourceID="SqlDataSource1" HeaderStyle-CssClass="h"
RowStyle-CssClass="r" BorderStyle="Solid" BorderColor="Black"
AutoGenerateDeleteButton="True">
<Columns>
    <asp:BoundField DataField="Id" HeaderText="Id"
    InsertVisible="False" ReadOnly="True" SortExpression="Id"
    />
    <asp:BoundField DataField="fnm" HeaderText="First Name"
    SortExpression="fnm" />
    <asp:BoundField DataField="lnm" HeaderText="Last Name"
    SortExpression="lnm" />
    <asp:BoundField DataField="fathernm" HeaderText="Father's
    Name" SortExpression="fathernm" />
    <asp:BoundField DataField="dob" HeaderText="DOB"
    SortExpression="dob" />
    <asp:BoundField DataField="gender" HeaderText="Gender"
    SortExpression="gender" />
    <asp:BoundField DataField="adhaarno" HeaderText="Adhaar
    No." SortExpression="adhaarno" />

    <asp:TemplateField HeaderText="Photo">
        <ItemTemplate>
            <asp:Image ID="photo" runat="server" ImageUrl='<%#
                "/images/" + Eval("photo") %>' />
        </ItemTemplate>
        <ControlStyle Height="100px" Width="100px" />
    </asp:TemplateField>
```

```

<asp:BoundField DataField="parentprof"
HeaderText="Parent's Profession"
SortExpression="parentprof" />
<asp:BoundField DataField="codept" HeaderText="Co./Dept.
Name" SortExpression="codept" />
<asp:BoundField DataField="designation"
HeaderText="Designation" SortExpression="designation" />
<asp:BoundField DataField="address" HeaderText="Address"
SortExpression="address" />
<asp:BoundField DataField="whatsappno"
HeaderText="Whatsapp No." SortExpression="whatsappno" />
<asp:BoundField DataField="email" HeaderText="Email"
SortExpression="email" />
<asp:BoundField DataField="studmno" HeaderText="Student's
Mobile No." SortExpression="studmno" />
<asp:BoundField DataField="pmno" HeaderText="Parent's
Mobile No." SortExpression="pmno" />
<asp:BoundField DataField="lastexam" HeaderText="Last
Appeared/Passed Exam" SortExpression="lastexam" />
<asp:BoundField DataField="yrofpassing" HeaderText="Year
Of Passing" SortExpression="yrofpassing" />
<asp:BoundField DataField="instnm" HeaderText="Institute
Name" SortExpression="instnm" />
<asp:BoundField DataField="pr" HeaderText="Percentage"
SortExpression="pr" />
<asp:BoundField DataField="std" HeaderText="Standard"
SortExpression="std" />
<asp:BoundField DataField="schoolnm" HeaderText="School
Name" SortExpression="schoolnm" />
<asp:BoundField DataField="course" HeaderText="Course"
SortExpression="course" />
<asp:BoundField DataField="sem" HeaderText="Semester"
SortExpression="sem" />
<asp:BoundField DataField="collegenm" HeaderText="College
Name" SortExpression="collegenm" />
<asp:BoundField DataField="othercourse" HeaderText="Other
Course" SortExpression="othercourse" />
<asp:BoundField DataField="status" HeaderText="Status"
SortExpression="status" />
<asp:HyperLinkField HeaderText="Confirm" Text="Confirm"
DataNavigateUrlFields="id"
DataNavigateUrlFormatString="confirmad.aspx?id={0}" />
</Columns>

<HeaderStyle CssClass="h"></HeaderStyle>

<RowStyle CssClass="r"></RowStyle>
</asp:GridView>

```

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:cn %>" 
SelectCommand="SELECT * FROM [admission]" 
DeleteCommand="DELETE FROM [admission] WHERE [id] =
@id"></asp:SqlDataSource>
```

## 2. Confirm Admission File:

- ```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        int id = Convert.ToInt32(Request.QueryString["id"]);

        string connectionString =
WebConfigurationManager.ConnectionStrings["cn"].ConnectionString;
        using (SqlConnection cn = new
SqlConnection(connectionString))
        {
            cn.Open();
            SqlCommand cmd = new SqlCommand("update admission
set status='confirmed' where id=@id", cn);
            cmd.Parameters.AddWithValue("@id", id);
            cmd.ExecuteNonQuery();
        }

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

        Response.Write("<script>window.location='admissions.as
px'</script>");
    }
}
```

### 3. New Admission File:

- ```

protected void Page_Load(object sender, EventArgs e)
{
    othercourse.DataSource = SqlDataSource1;
    othercourse.DataBind();
}
protected void submit_Click(object sender, EventArgs e)
{
    string str =
    WebConfigurationManager.ConnectionStrings["cn"].ToString()
;

    string check = "SELECT adhaarno FROM admission WHERE
adhaarno=@adhaarno";
    string query = "INSERT INTO
admission(fnm,lnm,fathernm,dob,gender,adhaarno,photo,paren
tprof,codept,designation,address,whatsappno,email,studmno,
pmno,lastexam,yrofpassing,instdnm,pr,std,schoolnm,course,se
m,collegenm,othercourse,status)
VALUES(@fnm,@lnm,@fathernm,@dob,@gender,@adhaarno,@photo,@
parentprof,@codept,@designation,@address,@whatsappno,@emai
l,@studmno,@pmno,@lastexam,@yrofpassing,@instdnm,@pr,@std,@
schoolnm,@course,@sem,@collegenm,@othercourse,'pending')";

    using (SqlConnection cn = new SqlConnection(str))
    {
        using (SqlCommand sqlcmd = new SqlCommand(check, cn))
        {
            sqlcmd.Parameters.AddWithValue("@adhaarno",
adhaarno.Text);

            cn.Open();

            using (SqlDataReader dr = sqlcmd.ExecuteReader())
            {
                if (dr.HasRows)
                {
                    Response.Write("<script>alert('The adhaarn
no. already exists')</script>");
                }
                else
                {
                    dr.Close();

                    using (SqlCommand cmd = new
SqlCommand(query, cn))
                    {

```

```
cmd.Parameters.AddWithValue("@fnm",
fnm.Text);
cmd.Parameters.AddWithValue("@lnm",
lnm.Text);
cmd.Parameters.AddWithValue("@fathernm",
, fathernm.Text);
cmd.Parameters.AddWithValue("@dob",
dob.Text);
cmd.Parameters.AddWithValue("@gender",
gender.SelectedValue);
cmd.Parameters.AddWithValue("@adhaarno",
, adhaarno.Text);
cmd.Parameters.AddWithValue("@photo",
Path.GetFileName(photo.PostedFile.FileName));
cmd.Parameters.AddWithValue("@parentprof",
parentprof.Text);
cmd.Parameters.AddWithValue("@codept",
codept.Text);
cmd.Parameters.AddWithValue("@designation",
designation.Text);
cmd.Parameters.AddWithValue("@address",
address.Text);
cmd.Parameters.AddWithValue("@whatsappno",
whatsappno.Text);
cmd.Parameters.AddWithValue("@email",
email.Text);
cmd.Parameters.AddWithValue("@studmno",
, studmno.Text);
cmd.Parameters.AddWithValue("@pmno",
pmno.Text);
cmd.Parameters.AddWithValue("@lastexam",
, lastexam.Text);
cmd.Parameters.AddWithValue("@yrofpassing",
yrofpassing.Text);
cmd.Parameters.AddWithValue("@instnm",
instnm.Text);
cmd.Parameters.AddWithValue("@pr",
pr.Text);
cmd.Parameters.AddWithValue("@std",
std.Text);
cmd.Parameters.AddWithValue("@schoolnm",
, schoolnm.Text);
cmd.Parameters.AddWithValue("@course",
course.SelectedValue);
cmd.Parameters.AddWithValue("@sem",
sem.SelectedValue);
cmd.Parameters.AddWithValue("@collegenm
```

```
        ", collegenm.Text);
cmd.Parameters.AddWithValue("@othercourse", othercourse.SelectedValue);
cmd.ExecuteNonQuery();

        string filePath =
Server.MapPath("/images/") +
Path.GetFileName(photo.PostedFile.FileName);
photo.SaveAs(filePath);

Response.Write("<script>alert('Applied
for admission
successfully...')</script>");

Response.Write("<script>window.location
='messages.aspx'</script>");
    }
}
cn.Close();
}
}
}
```

## 4. Courses File:

- ```
• protected void add_Click(object sender, EventArgs e)
{
    if (category.Text.Trim() != "" && course.Text.Trim() !=
    "" && fees.Text.Trim() != "" && duration.Text.Trim() != "")
    {
        string connectionString =
WebConfigurationManager.ConnectionStrings["cn"].ToString();
        using (SqlConnection connection = new
SqlConnection(connectionString))
        {
            connection.Open();
            string query = "INSERT INTO courses (category,
course, fees, duration) VALUES (@category,
@course, @fees, @duration)";
            SqlCommand cmd = new SqlCommand(query, connection);
            cmd.Parameters.AddWithValue("@category", category.Text);
            cmd.Parameters.AddWithValue("@course", course.Text);
            cmd.Parameters.AddWithValue("@fees", fees.Text);
            cmd.Parameters.AddWithValue("@duration", duration.Text);
            cmd.ExecuteNonQuery();
        }
    }
}
```

```

        using (SqlCommand command = new SqlCommand(query,
connection))
{
    command.Parameters.AddWithValue("@category",
category.Text.Trim());
    command.Parameters.AddWithValue("@course",
course.Text.Trim());
    command.Parameters.AddWithValue("@fees",
fees.Text.Trim());
    command.Parameters.AddWithValue("@duration",
duration.Text.Trim());

    command.ExecuteNonQuery();
}
Response.Write("<script>alert('Course Added
Successfully...')</script>");
}
Response.Redirect(Request.Url.AbsoluteUri);
}
else
{
    Response.Write("<script>alert('Please fill in all
fields')</script>");
}
}

```

## 5. Users File:

- `<asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False" DataSourceID="SqlDataSource1"
CellPadding="10" Width="100%" AutoGenerateDeleteButton="True"
DataKeyNames="mno" HeaderStyle-CssClass="h" RowStyle-
CssClass="r">
<Columns>
 <asp:BoundField DataField="id" HeaderText="Id"
SortExpression="id" />
 <asp:BoundField DataField="name" HeaderText="Name"
SortExpression="name" />
 <asp:BoundField DataField="email"
HeaderText="Email" SortExpression="email" />
 <asp:BoundField DataField="mno" HeaderText="Mobile
No." SortExpression="mobile" />
 <asp:BoundField DataField="usertype"
HeaderText="Usertype" SortExpression="usertype" />
</Columns>

<HeaderStyle CssClass="h"></HeaderStyle>`

```

<RowStyle CssClass="r"></RowStyle>
</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:cn %>" 
SelectCommand="SELECT [id], [name], [email], [mno],
[usertype] FROM [user_login]" DeleteCommand="DELETE
FROM [user_login] WHERE [mno] =
@mno"></asp:SqlDataSource>

```

## 6. Messages File:

```

• protected void send_Click(object sender, EventArgs e)
{
    if (subject.Text.Trim() != "" && message.Text.Trim() != ""
        && date.Text.Trim() != "")
    {
        string connectionString =
WebConfigurationManager.ConnectionStrings["cn"].ToString();

        using (SqlConnection connection = new
SqlConnection(connectionString))
        {

            connection.Open();

            string query = "INSERT INTO messages(subject,
message, date) VALUES (@subject, @message,
@date)";

            using (SqlCommand command = new SqlCommand(query,
connection))
            {

                command.Parameters.AddWithValue("@subject",
subject.Text.Trim());
                command.Parameters.AddWithValue("@message",
message.Text.Trim());
                command.Parameters.AddWithValue("@date",
date.Text.Trim());

                command.ExecuteNonQuery();

                Response.Write("<script>alert('Message Sent
Successfully...')</script>");
            }
        }
    }
}

```

```

        Response.Redirect(Request.Url.AbsoluteUri);
    }
    else
    {
        Response.Write("<script>alert('Please fill in all
fields')</script>");
    }
}

```

## 7. Student Query File:

- ```

<asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False" DataKeyNames="Id"
DataSourceID="SqlDataSource1" CellPadding="10" Width="100%"
HeaderStyle-CssClass="h" RowStyle-CssClass="r">
    <Columns>
        <asp:BoundField DataField="Id" HeaderText="Id"
InsertVisible="False" ReadOnly="True"
SortExpression="Id" />
        <asp:BoundField DataField="name" HeaderText="Name"
SortExpression="name" />
        <asp:BoundField DataField="email"
HeaderText="Email" SortExpression="email" />
        <asp:BoundField DataField="mno" HeaderText="Mno"
SortExpression="mno" />
        <asp:BoundField DataField="query"
HeaderText="Query" SortExpression="query" />
        <asp:BoundField DataField="reply"
HeaderText="Reply" SortExpression="reply" />
        <asp:HyperLinkField HeaderText="Reply Query"
Text="Reply" DataNavigateUrlFields="id"
DataNavigateUrlFormatString="replyquery.aspx?id={0
}" />
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server"
ConnectionString=<%$ ConnectionStrings:cn %>"'
SelectCommand="SELECT * FROM [studquery]">
</asp:SqlDataSource>

```

## 8. Reply Student Query File:

- ```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        if (Request.QueryString["id"] != null)
        {
            int id =
                Convert.ToInt32(Request.QueryString["id"]);
            LoadData(id);
            this.id.Value = id.ToString();
        }
    }
}

protected void LoadData(int id)
{
    string connectionString =
        WebConfigurationManager.ConnectionStrings["cn"].ConnectionString;
    string query = "select * from studquery where id=@id";

    using (SqlConnection connection = new
        SqlConnection(connectionString))
    {
        SqlCommand command = new SqlCommand(query,
            connection);
        command.Parameters.AddWithValue("@id", id);

        connection.Open();

        SqlDataReader reader = command.ExecuteReader();

        if (reader.Read())
        {
            Query.Text = reader["query"].ToString();
            Reply.Text = reader["reply"].ToString();
        }
        reader.Close();
    }
}

protected void send_Click(object sender, EventArgs e)
{
    int id = Convert.ToInt32(this.id.Value);
    string reply = Reply.Text.Trim();
}

```

```

string connectionString =
WebConfigurationManager.ConnectionStrings["cn"].ConnectionString;
string query = "update studquery set reply=@reply where
id=@id";

using (SqlConnection connection = new
SqlConnection(connectionString))
{
    SqlCommand command = new SqlCommand(query,
connection);
    command.Parameters.AddWithValue("@reply", reply);
    command.Parameters.AddWithValue("@id", id);

    connection.Open();

    command.ExecuteNonQuery();
}
Response.Write("<script> alert('Reply Sent
Successfully')</script>");

Response.Write("<script>window.location='studquery.aspx
'</script>");
}

```

## 9. Leave Application File:

- <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False" DataKeyNames="Id"
DataSourceID="SqlDataSource1" CellPadding="10" Width="100%"
HeaderStyle-CssClass="h" RowStyle-CssClass="r">
<Columns>
 <asp:BoundField DataField="Id" HeaderText="Id"
 InsertVisible="False" ReadOnly="True"
 SortExpression="Id" />
 <asp:BoundField DataField="name" HeaderText="Name"
 SortExpression="name" />
 <asp:BoundField DataField="email"
 HeaderText="Email" SortExpression="email" />
 <asp:BoundField DataField="mno" HeaderText="Mobile
 No." SortExpression="mno" />
 <asp:BoundField DataField="subject"
 HeaderText="Subject" SortExpression="subject" />
 <asp:BoundField DataField="from\_dt"
 HeaderText="From Date" SortExpression="from\_dt" />
 <asp:BoundField DataField="to\_dt" HeaderText="To
 Date" SortExpression="to\_dt" />

```

        <asp:BoundField DataField="lv_reason"
HeaderText="Leave Reason"
SortExpression="lv_reason" />
<asp:BoundField DataField="status"
HeaderText="Status" SortExpression="status" />
<asp:HyperLinkField HeaderText="Grant"
Text="Grant" DataNavigateUrlFields="id"
DataNavigateUrlFormatString="grantlv.aspx?id={0}"
/>
</Columns>

</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource1"
runat="server"
ConnectionString="<%$ ConnectionStrings:cn %>"
SelectCommand="SELECT * FROM
[lv_app]"></asp:SqlDataSource>
```

## 10. Grant Leave File:

- ```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        int id = Convert.ToInt32(Request.QueryString["id"]);

        string connectionString =
WebConfigurationManager.ConnectionStrings["cn"].ConnectionString;
        using (SqlConnection cn = new
SqlConnection(connectionString))
        {
            cn.Open();
            SqlCommand cmd = new SqlCommand("update lv_app set
status='granted' where id=@id", cn);
            cmd.Parameters.AddWithValue("@id", id);
            cmd.ExecuteNonQuery();
        }

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

        Response.Write("<script>window.location='lv_app.aspx'<
/script>");
    }
}
```

## 11. PDF File:

- ```

protected void btnUpload_Click(object sender, EventArgs e)
{
    if (fileUpload.HasFile)
    {
        string fileName =
            Path.GetFileName(fileUpload.PostedFile.FileName);
        string fileExtension = Path.GetExtension(fileName);
        if (fileExtension == ".pdf")
        {
            string savePath = Server.MapPath("~/pdf/") +
                fileName;
            if (File.Exists(savePath))
            {
                Response.Write("<script>alert('The pdf
already exists')</script>");
            }
            else
            {
                fileUpload.PostedFile.SaveAs(savePath);
                string connectionString =
                    ConfigurationManager.ConnectionStrings["cn"]
                    .ConnectionString;
                string query = "INSERT INTO pdf (course,
sem, subject, pdf, path) VALUES (@category,
@sem, @subject, @pdf, @path)";
                using (SqlConnection conn = new
                    SqlConnection(connectionString))
                {
                    using (SqlCommand cmd = new
                        SqlCommand(query, conn))
                    {
                        cmd.Parameters.AddWithValue("@category",
                            course.Text);
                        cmd.Parameters.AddWithValue("@sem",
                            sem.Text);

                        cmd.Parameters.AddWithValue("@subject",
                            subject.Text);
                        cmd.Parameters.AddWithValue("@pdf",
                            fileName);
                        cmd.Parameters.AddWithValue("@path",
                            savePath);
                        conn.Open();
                        cmd.ExecuteNonQuery();
                        conn.Close();
                    }
                }
            }
        }
    }
}

```

```
        Response.Redirect(Request.Url.AbsoluteUri);
    }
}
```

## 12. Feedback File:

- ```
<asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False" DataKeyNames="Id"
DataSourceID="SqlDataSource1" AutoGenerateDeleteButton="True"
CellPadding="10" Width="100%" HeaderStyle-CssClass="h"
RowStyle-CssClass="r">
    <Columns>
        <asp:BoundField DataField="Id" HeaderText="Id"
            InsertVisible="False" ReadOnly="True"
            SortExpression="Id" />
        <asp:BoundField DataField="name" HeaderText="Name"
            SortExpression="name" />
        <asp:BoundField DataField="email" HeaderText="Email"
            SortExpression="email" />
        <asp:BoundField DataField="mno" HeaderText="Mobile No."
            SortExpression="mno" />
        <asp:BoundField DataField="feedback"
            HeaderText="Feedback" SortExpression="feedback" />
    </Columns>
</asp:GridView>

<asp:SqlDataSource ID="SqlDataSource1"
runat="server"
ConnectionString="<%$ ConnectionStrings:cn %>"
SelectCommand="SELECT * FROM [feedback]"
DeleteCommand="DELETE FROM [feedback] WHERE [id] =
@id" ></asp:SqlDataSource>
```

### 13. Profile File:

- ```
• protected void upd_Click(object sender, EventArgs e)
{
    string str =
        WebConfigurationManager.ConnectionStrings["cn"].ToString();
    string name =
```

```

((TextBox)profile.Items[0].FindControl("name")).Text;
string email =
((TextBox)profile.Items[0].FindControl("email")).Text;

string query = "UPDATE user_login SET name=@name,
email=@email WHERE mno=@mno";

using (SqlConnection cn = new SqlConnection(str))
{
    using (SqlCommand cmd = new SqlCommand(query, cn))
    {
        cmd.Parameters.AddWithValue("@name", name);
        cmd.Parameters.AddWithValue("@email", email);
        cmd.Parameters.AddWithValue("@mno",
Session["admin"].ToString());

        cn.Open();
        cmd.ExecuteNonQuery();
        cn.Close();
    }
}
Response.Write("<script>alert('Profile Updated
Successfully....')</script>");
profile.DataBind();
}

```

#### 14. Change Password File:

- ```

protected void confirm_Click(object sender, EventArgs e)
{
    string str =
WebConfigurationManager.ConnectionStrings["cn"].ToString();

    string confirmpassword = this.newpassword.Text;

    using (SqlConnection cn = new SqlConnection(str))
    {
        using (SqlCommand cmd = new SqlCommand())
        {
            cmd.Connection = cn;
            cmd.CommandType = CommandType.Text;

            cn.Open();
            SqlDataReader reader = cmd.ExecuteReader();

            if (reader.HasRows)
            {

```

```
        reader.Close();
        using (SqlCommand cmd2 = new SqlCommand())
        {
            cmd2.Connection = cn;
            cmd2.CommandType = CommandType.Text;

            cmd2.CommandText = "update user_login set
password = @newpassword where mno = @mno";

            cmd2.Parameters.AddWithValue("@newpassword",
", this.newpassword.Text);

            cmd2.ExecuteNonQuery();
        }

        Response.Write("<script>alert('Your Password
Updated Successfully...')</script>");
    }
    else
    {
        Response.Write("<script>alert('Please enter
a valid email or mobile no. or
password')</script>");
    }
    cn.Close();
}
}
```

### 15. Logout File:

- ```
• protected void Page_Load(object sender, EventArgs e)
{
    Session.Clear();
    Session.Abandon();
    Session.RemoveAll();
    Response.Redirect("~/Login.aspx");
}
```

## CHAPTER:5

### Testing

5.1 Techniques and Strategies

5.2 Cost Estimation Model

5.3 Future Scope and Further Enhancement of the Project

5.4 Bibliography

5.5 Appendices

5.6 Glossary

#### 5.1 Techniques and Strategies

Testing is a process to show the corrections of the program. Testing is needed to show completeness, to improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary reduce the testing costs and operation time.

Testing software extends throughout the coding phase and it represents the ultimate review of configurations, design and coding. Based on the way the software reacts to these testing.

We can decide whether the configuration that has been built is study or not. All components of an application are tested, as the failure to do so many results in a series of bugs after the software is put to use.

## 1. White Box Testing

White Box (or glass box) testing is the process of giving input to the system and checking how the system processes input to generate output.

It refers to the testing a system with full knowledge and access to all source code and other architecture documents. This testing enables to reveal bugs and vulnerabilities quickly in comparison with trial-and-error method. More complete testing coverage is ensured by exactly knowing what to test.

White box testing involves thorough testing of the application. It requires knowledge of code and the test cases chosen verifies if the system is implemented as expected. It typically includes checking with the data flow, exceptions, and errors, how they are handled, comparing if the code produces the expected results.

## 2. Black Box Testing

Black Box testing is the process of giving input to the system and checking the output of the system without bothering how the output is generated. It refers to testing a system without knowledge of specification to the internal workings of the system, access to the source code, and knowledge of the architecture.

Essentially this approach mimics in a close approach, how an attacker typically follows approach to the application. However, the uncovering of issues or vulnerabilities could be further longer, because of lacking internal application knowledge.

Black box testing is done at an outer level of the system. Test cases merely check if the output is correct for the given input. User is not expected to the internal flow or design of the system.

### 3. Grey Box Testing

Grey Box testing is a combination of White Box and Glass Box Testing. In this, the tester has little knowledge about the internal working of the software. So, he tests the output as well as process carried out to generate the output.

It refers to a testing system by knowing limited information about the internals of the system. The knowledge is always limited for detailed design documents and architecture diagrams. In concise, it is a good blend of black and white box testing, which leverages the strengths of each of the testing.

Grey box testing is a combination of both black box and white box testing. This is because it involves access to the system; however, at an outer level. A little knowledge of the system is expected in grey box testing.

### 4. Non-Functional Testing

Non-functional testing is the testing of a software application or system for its non-functional Requirements: the way a system operates, rather than specific behaviors of that system.

This is contrast to functional testing, which tests against functional requirements that describe the functions of a system and its components.

The names of many non-functional tests are often used interchangeably because of the overlap in scope between various non-functional requirements. For example, software performance is a broad term that includes many specific requirements like reliability and scalability.

## ➤ Software Testing Strategies

Testing involves:

### 1. Unit Testing

The unit testing is purpose of unit testing is to ensure that each program is fully tested.

### 2. Integration Testing

The integration testing is individual program units or programs are integrated and tested as a complete system to ensure that the software requirements.

### 3. Acceptance Testing

This testing involves planning and the execution of various types of tests in order to demonstrate that the implemented software system satisfied the requirements.

Finally, our project meets the requirements after going through all the levels of testing.

## 5.2 Cost Estimation Model

### ➤ Costing Objectives

- To ensure viability
  - ✓ Feasibility study
  - ✓ Resource planning
  - ✓ Cost/benefit analysis
- Provide input for pricing (including bidding negotiations etc.)
- To serve as a management tool
  - ✓ Cost Control and Management
  - ✓ Risk management
  - ✓ Budget planning
- Criteria for good project coding
  - ✓ Accurate
  - ✓ Realistic (good procurement and engineering practice)
  - ✓ Consistent
  - ✓ Transparent
  - ✓ Cost-effective
  - ✓ Good Documentation

## Project Costing

Many looks upon project testing as a cost. While it is true that software testing does cost money, in many cases significant amounts of money, it is also an activity that an organization to avoid costly failures further on in the development process.

Most understand this relationship project testing is spending money to save money. What many do not also realize is that software testing also produces valuable assets for the organization. This article will discuss those assets of software testing.

### ➤ Cost Estimation

- Working time estimation is given below:

1.19 month	= 50 days
5 hrs / day	= 250 hours

- Now, the expenses & cost estimation are given below:

Computer rent	= 5,000 /-
+ Light Bill Rs. 7.50 / unit (worth 400 units)	= 3,000 /-
+ Database design & creation	= 1,500 /-
+ Coding & Validation	= 2,500 /-

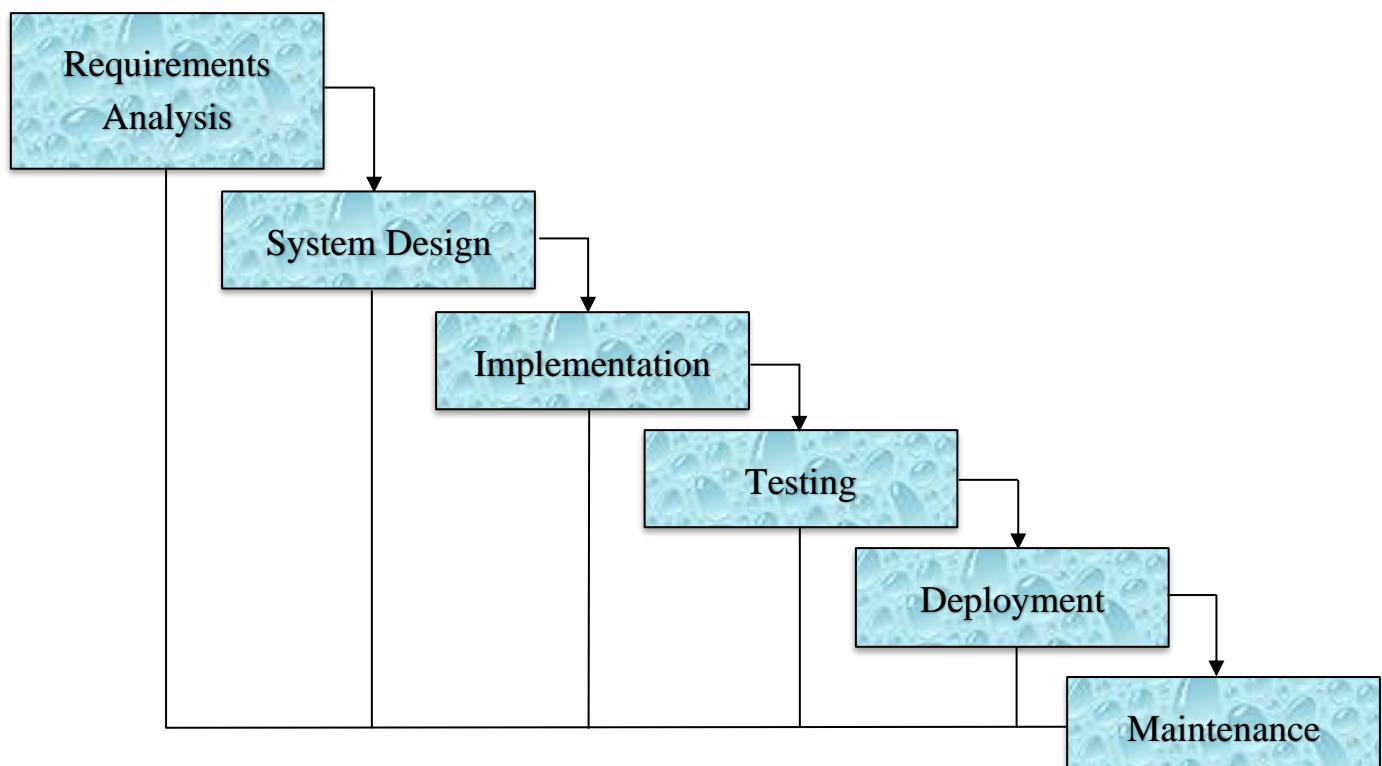
---

Total = 12,000 /-

## SDLC Model

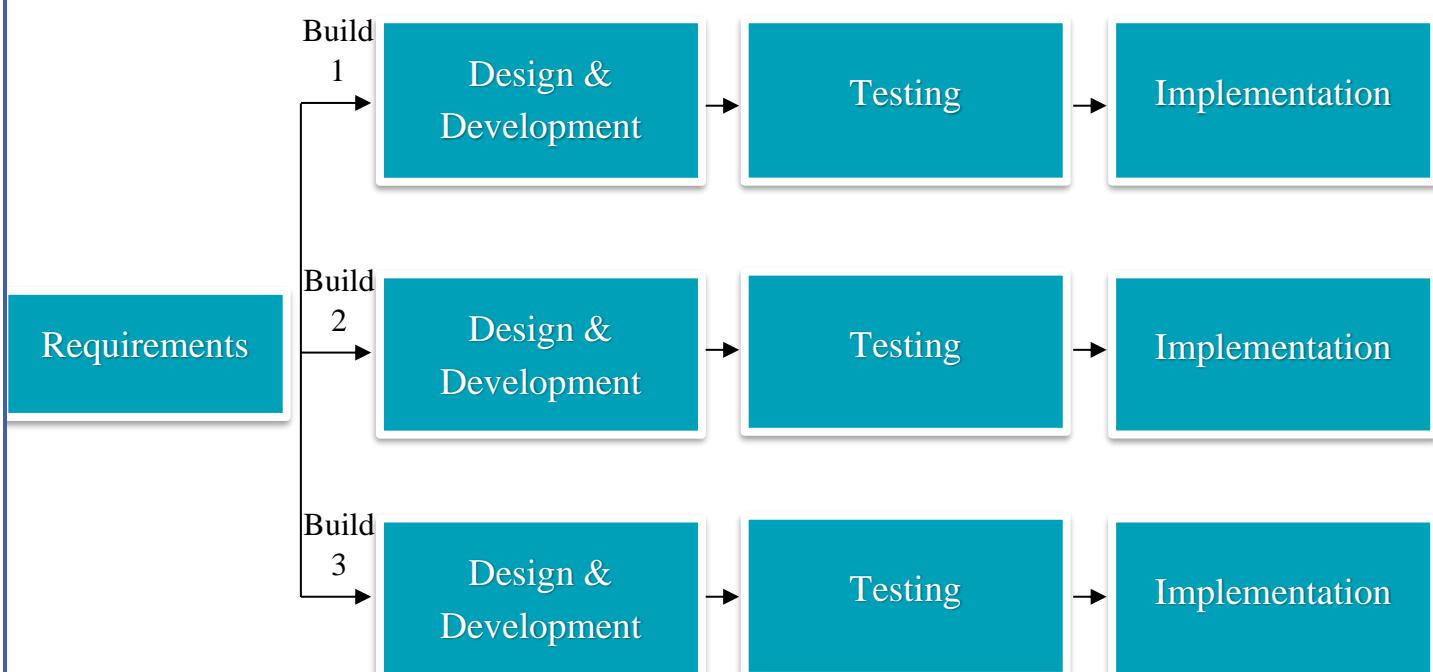
### ➤ Waterfall Model

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.



## ➤ Iterative Model

In Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

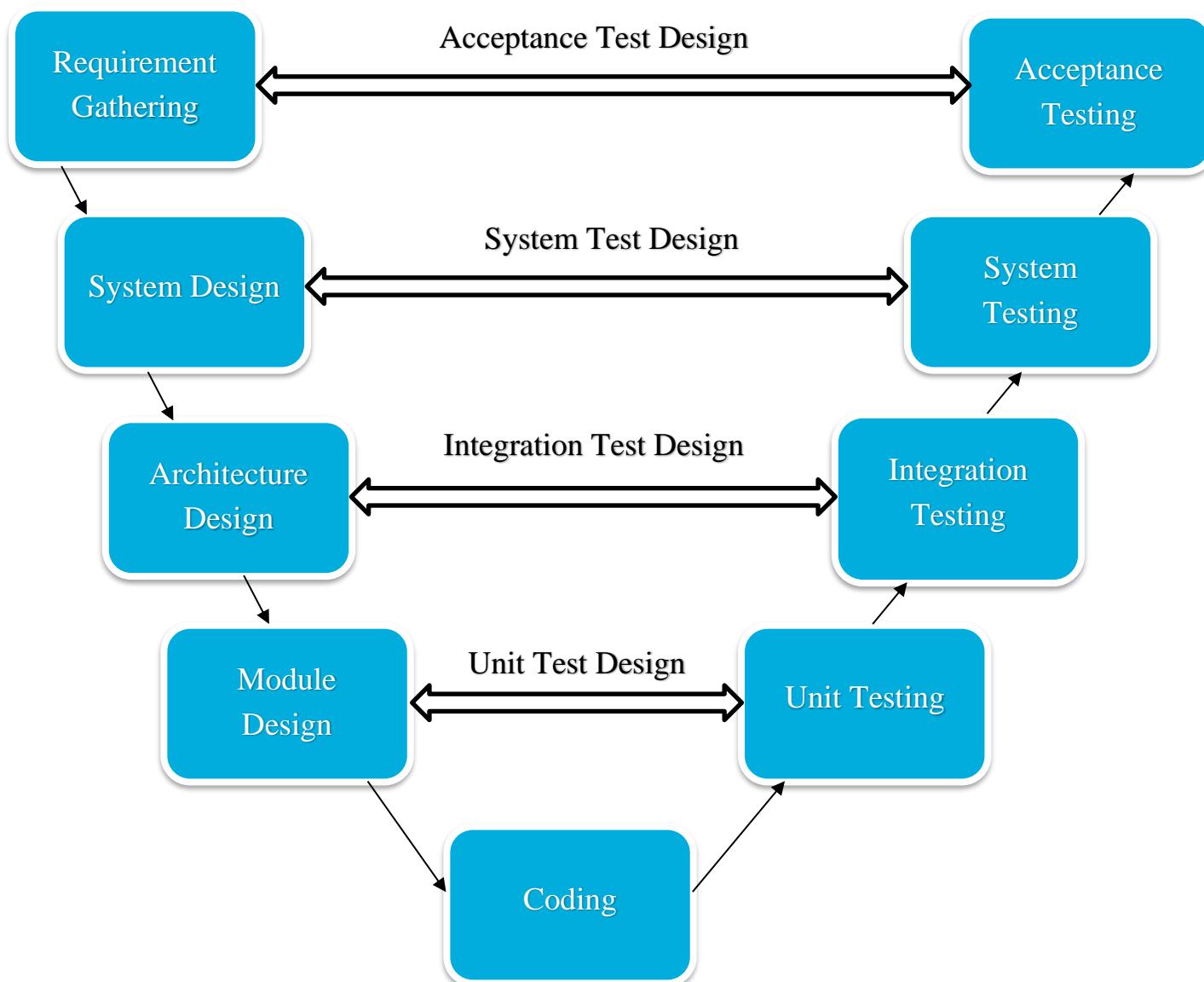


## ➤ V-Model

The V-Model is SDLC model where execution of processes Verification happens in a sequential manner in V-shape. It is also known as Verification and Validation model. V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase.

## ➤ V-Model Design

Under V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the V-Model and Validation phases on the other side. Coding phase joins the two sides of the V-Model. The below figure illustrates the different phases in V-Model of SDLC.

VerificationValidation

## ➤ Spiral Model

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. Spiral model is a combination of iterative development process model and sequential linear development model i.e., waterfall model with very high emphasis on risk analysis. It allows for incremental releases of the product, or incremental refinement through each iteration around the spiral.

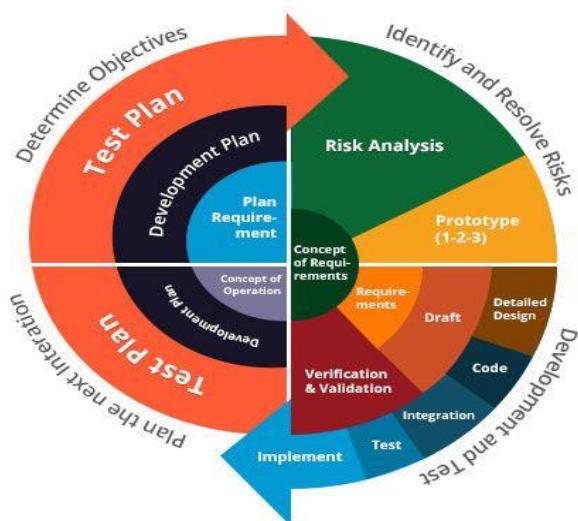
## ➤ Spiral Model Design

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

- Identification:

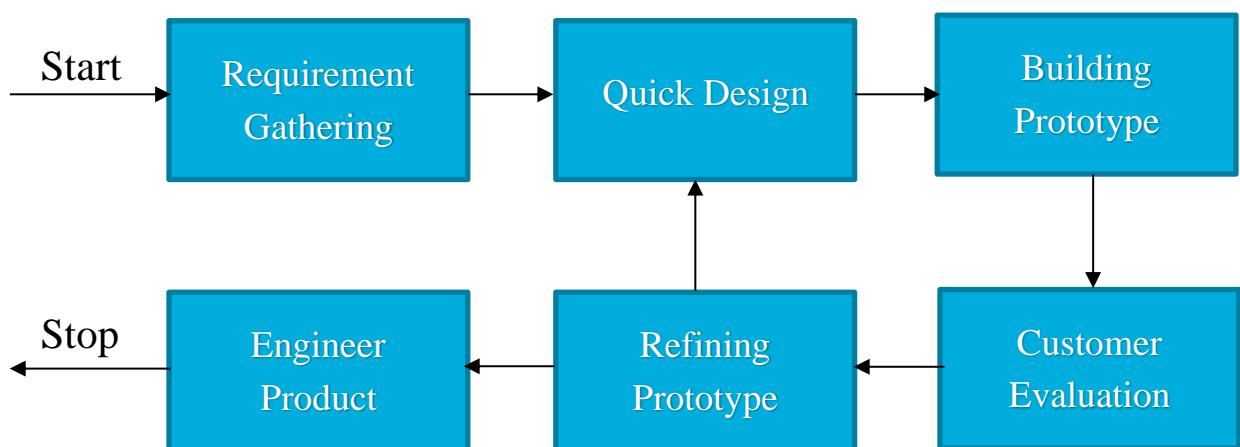
This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in.

This phase This also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral the product is deployed in the identified market.



## ➤ Prototype Model

The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. The prototype is usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.



### 5.3 Future Enhancements

A good project is one which never stops developing according to the changing situations and technologies there is a lot of scope of future enhancements.

- Online Payment.
- Online Teaching.
- Student Attendance.
- Online Mock Tests.
- New User To-Do List.

### 5.4 Bibliography

For the successful working of my project, I have referred many sources for the code snippets, logic and tips and tricks from the various books as well as web sites. Most I searched for the required possessions on the google.com search engine.

- Web Link:
  1. google.com
  2. w3schools.com

## 5.5 Appendices

The project Aavishkar Institute is the customize working on Admission of students and provides online material to them.

The software takes care of all requirement and helps to the students for taking admission easily.

We have successfully designed, coded and implemented our project with a lot of hard work.

Finally, I would like to thank our project guide Dr. Neha K. Shah for giving helpful guidelines for our project and also for giving suggestions when difficulties raised in our project.

## 5.6 Glossary

ASP	:	Active Server Pages
CSS	:	Cascading Style Sheets
DFD	:	Data Flow Diagram
ERD	:	Entity Relationship Diagram
GUI	:	Graphical User Interface
HTML	:	Hyper Text Markup Language
IDE	:	Integrated Development Environment
ISO	:	International Organization for Standardization
PERT	:	Program Evaluation Review Technique
RDBMS	:	Relational Database Management System
SOAP	:	Simple Object Access Protocol
SDLC	:	Software Development Life Cycle
SRS	:	Software Requirements Specification
SVG	:	Scalable Vector Graphics
SQL	:	Structured Query Language
T-SQL	:	Transact SQL
W3C	:	World Wide Web Consortium
XHTML	:	Extensible Hyper Text Markup Language
XML	:	Extensible Markup Language
XUL	:	Extensible User Interface Language

# Thank You