**1. INTRODUCTION**

**1. Introduction**

**1.1 Introduction to the project**

This project ‘Knowledge Bank’ focuses on the current scenario of Digitalization World. It bridges the gap between ‘Learner’ and the ‘Information Source’. Knowledge Bank provides us a web platform which enables us to search for our relevant content and download the related document files for free. Now, here comes the role of this project ‘Knowledge Bank’.

As the name suggest ‘Knowledge and Bank’ i.e. As in general term ‘Bank’ refers to a place where we keep our important things (like: money, jewelry, bon papers etc.) now coming in the context of knowledge, here we refer knowledge as ‘Information’. Information, which also have the same importance as the money and rest all. So our project main idea is to collects and share the knowledge and information of any field with any one, who is willing to gain some knowledge.

Knowledge Bank provides us a web portal where a person can upload their research or information/data and the person who is willing or searching for the knowledge can get it here at no cost.

Our project is also promoting the ‘Social Work and Awareness’ of sharing knowledge and helping those who are in need.

**1.2 Purpose of the project**

The prime objective of the Knowledge bank is to provide a platform to the users to gain and share their knowledge. There is a “Give and Get” relationship.

**1.3 Problems in the existing system**

There are many people who have great knowledge of many topics. But the problem is that they are unable to share it to the intended audience. People also face the lack of good information, nowadays.

**1.4 Solution to these problems**

Solution to this problem is to provide a web platform to the ‘Users’ to share and access the Information and knowledge to one another.

**2. Feasibility Study**

**2. Feasibility Study**

**2.1 Technical Feasibility**

**On performing technical feasibility study on this project, which is based on the inputs, processes and outputs.**

**2.2 Economic Feasibility**

On Performing Economic Feasibility on Knowledge Bank , we find that the owner of this Web Application doesn’t have to invest much cost for owning it i.e. he only have to invest one-time cost for owning it and rest on maintenance cost for one year is free and after it a minimal yearly maintenance cost will be charged by the developer.

It is an Open-Source Application which will accessible to its entire user.

**2.3 Processes and Functionalities**

1. User Registration
2. Login
3. Admin
4. Add Category
5. Add Subcategory
6. View/Edit Category
7. View/Edit Subcategory
8. View Faculty Details and his Topics Uploaded.
9. User
10. Add Topics.
11. View Topic
12. Edit Topic
13. Delete Topic
14. Visitor
15. View Topics (according to their need ;)

**2.4 Input**

1. **Faculty**

|  |
| --- |
| **Data** |
| FacultyId |
| FacultyName |
| Email |
| Mobile |
| Designation |
| RegDate |

1. **Topic**

|  |
| --- |
| **Data** |
| TopicId |
| TopicName |
| FacultyId |
| Descripton |
| UploadDate |
| File(pdf,doc,etc) |

1. **Category**

|  |
| --- |
| **Data** |
| CategoryId |
| CategoryName |

1. **Subcategory**

|  |
| --- |
| **Data** |
| SubcategoryId |
| SubcategoryName |

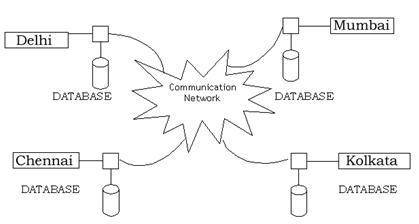
**3. Software Requirement Specification**

**3.1 Functional Requirement**

These system features include:

**Distributed Database:**

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as shown in below figure.

[](https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database/attachment/ad-distributed-database-located-at-four-different-cities)

**Client/Server System:**

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

* Some sites are client sites and others are server sites.
* All the data resides at the server sites.
* All applications execute at the client sites.

**3.2 HARDWARE**

**REQUIREMENT**

**HARDWARE REQUIRED**

**RAM**

Minimum 512 MB

**BACKUP**

CD/Pen drive

**PROCESSOR**

Intel Pentium IV

**HARD DISK**

Minimum 20GB or Above

**3.3 SOFTWARE**

**REQUIREMENT**

**SOFTWARE REQUIRED**

**FRONT END**

* Php ver. 7.2.10
* CSS
* HTML 5
* Java Script 1.8.2

**BACK END**

* MySQL 5.7.23

**WEB SERVER**

* Apache Tomcat 2.4.35

**OPERATING SYSTEM**

* Windows XP & Above

**SOFTWARE**

* Wamp 2.4

**Hypertext Preprocessor** (**PHP**)

**PHP** started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
* PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
* PHP is forgiving: PHP language tries to be as forgiving as possible.
* PHP Syntax is C-Like.

**Advantages**

1. Cross Platform
2. Easy database connection.
3. Easy to use
4. Open Source
5. PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
6. PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
7. You add, delete, and modify elements within your database through PHP.
8. Access cookies variables and set cookies.
9. Using PHP, you can restrict users to access some pages of your website.
10. It can encrypt data.

**Java Script**

**JavaScript** ,often abbreviated as **JS**, is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language), [weakly typed](https://en.wikipedia.org/wiki/Weak_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming), [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language), and [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), JavaScript is one of the three core technologies of [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) [content production](https://en.wikipedia.org/wiki/Content_engineering). It is used to make webpages interactive and provide online programs, including video games. The majority of [websites](https://en.wikipedia.org/wiki/Website) employ it, and all modern [web browsers](https://en.wikipedia.org/wiki/Web_browser) support it without the need for [plug-ins](https://en.wikipedia.org/wiki/Browser_extension) by means of a built-in [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine).

Initially only implemented [client-side](https://en.wikipedia.org/wiki/Client-side) in web browsers, JavaScript engines are now embedded in many other types of host software, including [server-side](https://en.wikipedia.org/wiki/Server-side) in web servers and databases, and in non-web programs such as word processors and [PDF](https://en.wikipedia.org/wiki/Portable_Document_Format) software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Java script is one of the most simple, versatile and effective languages used to extend functionality in websites. Uses range from on screen visual effects to processing and calculating data on web pages with ease as well as extended functionality to websites using third party scripts among several other handy features, however it also possesses some negative effects that might make you want to think twice before implementing Java script on your website. Let's look at some of its importance:

**Advantages**

* **Java script is executed on the client side.**

This means that the code is executed on the user's processor instead of the web server thus saving bandwidth and strain on the web server.

* **Java script is a relatively easy language.**

The Java script language is relatively easy to learn and comprises of syntax that is close to English. It uses the DOM model that provides plenty of prewritten functionality to the various objects on pages making it a breeze to develop a script to solve a custom purpose.

* **Java script is relatively fast to the end user**

As the code is executed on the user's computer, results and processing is completed almost instantly depending on the task (tasks in java script on web pages are usually simple so as to prevent being a memory hog) as it does not need to be processed in the site's web server and sent back to the user consuming local as well as server bandwidth.

* **Extended functionality to web pages**

Third party add-ons like Grease monkey enables Java script developers to write snippets of Java script which can execute on desired web pages to extend its functionality.

**Cascading Style Sheets** (**CSS**)

**Cascading Style Sheets** (**CSS**) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language). Although most often used to set the visual style of [web pages](https://en.wikipedia.org/wiki/Web_page) and user interfaces written in [HTML](https://en.wikipedia.org/wiki/HTML) and [XHTML](https://en.wikipedia.org/wiki/XHTML), the language can be applied to any [XML](https://en.wikipedia.org/wiki/XML) document, including [plain XML](https://en.wikipedia.org/wiki/Plain_Old_XML), [SVG](https://en.wikipedia.org/wiki/Scalable_Vector_Graphics) and [XUL](https://en.wikipedia.org/wiki/XUL), and is applicable to rendering in [speech](https://en.wikipedia.org/wiki/Speech_synthesis), or on other media. Along with HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for [web applications](https://en.wikipedia.org/wiki/Web_applications), and user interfaces for many mobile applications.

**Advantages**

* **Separation of content from presentation**

CSS facilitates publication of content in multiple presentation formats based on nominal parameters. Nominal parameters include explicit user preferences, different web browsers, the type of device being used to view the content (a desktop computer or mobile Internet device), the geographic location of the user and many other variables.

* **Site-wide consistency**

When CSS is used effectively in terms of inheritance and "cascading", a global style sheet can be used to affect and style elements site-wide. If the situation arises that the styling of the elements should be changed or adjusted, these changes can be made by editing rules in the global style sheet. Before CSS, this sort of maintenance was more difficult, expensive and time-consuming.

* **Bandwidth**

A style sheet, internal or external, specifies the style once for a range of HTML elements selected by class, type or relationship to others. This is much more efficient than repeating style information inline for each occurrence of the element. An external style sheet is usually stored in the [browser cache](https://en.wikipedia.org/wiki/Browser_cache), and can therefore be used on multiple pages without being reloaded, further reducing data transfer over a network.

* **Page reformatting**

With a simple change of one line, a different style sheet can be used for the same page. This has advantages for accessibility, as well as providing the ability to tailor a page or site to different target devices. Furthermore, devices not able to understand the styling still display the content.

* **Accessibility**

Without CSS, web designers must typically lay out their pages with techniques such as HTML tables that hinder accessibility for vision-impaired users.

**Hypertext Markup Language (HTML)**

**Hypertext Markup Language** (**HTML**) is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for creating [web pages](https://en.wikipedia.org/wiki/Web_page) and [web applications](https://en.wikipedia.org/wiki/Web_application). With [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript) it forms a triad of cornerstone technologies for the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web).[[3]](https://en.wikipedia.org/wiki/HTML#cite_note-3) [Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and render them into multimedia web pages. HTML describes the structure of a web page [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects, such as [interactive forms,](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. It provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items. HTML elements are delineated by *tags*, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets).

**Advantages**

* It is the language which can be easily understand and can be modified.
* Effective presentations can be made with the HTML with the help of its all formatting tags.
* It provides the more flexible way to deign web pages along with the text.
* Links can also be added to the web pages so it help the readers to browse the information of their interest.
* You can display HTML documents on any platforms such as Macintosh, Windows and Linux etc.
* Graphics, videos and sounds can also be added to the web pages which give an extra attractive look to your web pages.

**Apache Tomcat**

**Apache Tomcat**, often referred to as **Tomcat Server**, is an open-source [Java Servlet Container](https://en.wikipedia.org/wiki/Servlet_container) developed by the [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) (ASF). Tomcat implements several [Java EE](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) specifications including [Java Servlet](https://en.wikipedia.org/wiki/Java_Servlet), [Java Server Pages](https://en.wikipedia.org/wiki/JavaServer_Pages) (JSP), [Java EL](https://en.wikipedia.org/wiki/Unified_Expression_Language), and [Web Socket](https://en.wikipedia.org/wiki/WebSocket), and provides a "pure [Java](https://en.wikipedia.org/wiki/Java_(programming_language))" [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [web server](https://en.wikipedia.org/wiki/Web_server) environment in which [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) code can run.

Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License) 2.0 license, and is [open-source software](https://en.wikipedia.org/wiki/Open-source_software).

## Features

Tomcat 7.x implements the Servlet 3.0 and JSP 2.2 specifications. It requires Java version 1.6, although previous versions have run on Java 1.1 through 1.5. Versions 5 through 6 saw improvements in [garbage collection](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)), JSP parsing, performance and scalability. Native wrappers, known as "Tomcat Native", are available for [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and UNIX for platform integration.

Tomcat 8.x implements the Servlet 3.1 and JSP 2.4 Specifications. Apache Tomcat 8.5.x is intended to replace 8.0.x and includes new features pulled forward from Tomcat 9.0.x. The minimum Java version and implemented specification versions remain unchanged.

**MySQL** 

**MySQL** is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (**SQL**).**SQL** is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. **MySQL** is an essential part of almost every open source **PHP** application. Good examples for PHP & MySQL-based scripts are [WordPress](https://www.siteground.com/tutorials/wordpress/), [Joomla](https://www.siteground.com/tutorials/joomla/), [Magneto](https://www.siteground.com/tutorials/magento/) and [Drupal](https://www.siteground.com/tutorials/drupal/). One of the most important things about using **MySQL** is to have a **MySQL** specialized host. Here are some of the things Site Ground can offer:

* We have long experience in providing technical support for **MySQL**-based web sites. Thanks to it our servers are perfectly optimized to offer the best overall performance for most **MySQL** applications.
* We offer a lot of free **MySQL** tools including CMS systems, forums, galleries, blogs, shopping carts and more.
* We support MySQL 5 and we provide unlimited **MySQL** databases on all our hosting plans.

**Advantages**

## ****1.   Data Security****

MySQL is globally renowned for being the most secure and reliable database management system used in popular web applications like WordPress, Drupal, Joomla, Facebook and Twitter.

## ****2.    On-Demand Scalability****

MySQL offers unmatched scalability to facilitate the management of deeply embedded apps using a smaller footprint even in massive warehouses that stack terabytes of data.

## ****3.    High Performance****

MySQL features a distinct storage-engine framework that facilitates system administrators to configure the MySQL database server for a flawless performance.

## ****4.    Round-the-clock Uptime****

MySQL comes with the assurance of 24X7 uptime and offers a wide range of high availability solutions like specialized cluster servers and master/slave replication configurations.

## ****5.    Comprehensive Transactional Support****

MySQL tops the list of robust transactional database engines available on the market. With features like complete atomic, consistent.

## ****6.    Complete Workflow Control****

With the average download and installation time being less than 30 minutes, MySQL means usability from day one. Whether your platform is Linux, Microsoft, Macintosh or UNIX.

## ****7.    Reduced Total Cost of Ownership****

By migrating current database apps to MySQL, enterprises are enjoying significant cost savings on new projects.

## ****8.    The Flexibility of Open Source****

All the fears and worries that arise in an open source solution can be brought to an end with My SQL’s round-the-clock support and enterprise indemnification.

* 1. **SYSTEM DESIGN**

**4.1 Introduction**

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements business or organization.   
  
A systemic approach is required for a coherent and well-running system. Bottom-Up or Top-Down approach is required to take into account all related variables of the system. A designer uses the modelling languages to express the information and knowledge in a structure of system that is defined by a consistent set of rules and definitions. The designs can be defined in graphical or textual modelling languages. 

**4.2 Entity Relationship**

**Diagram (ERD)**

**E-R Diagram**

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

The ER model becomes an abstract [data model](https://en.wikipedia.org/wiki/Data_modeling) that defines a data or information structure that can be implemented in a [database](https://en.wikipedia.org/wiki/Database), typically a [relational database](https://en.wikipedia.org/wiki/Relational_database).

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. An ER model can also be expressed in a verbal form, for example: one building may be divided into zero or more apartments, but one apartment can only be located in one building.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a [database](https://en.wikipedia.org/wiki/Database). In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a [relational database](https://en.wikipedia.org/wiki/Relational_database) a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the [three schema approach](https://en.wikipedia.org/wiki/Three_schema_approach) to [software engineering](https://en.wikipedia.org/wiki/Software_engineering).

**ERD entity symbols**

Entities are objects or concepts that represent important data. Entities are typically nouns such as product, customer, location, or promotion. There are three types of entities commonly used in entity relationship diagrams.

|  |  |  |
| --- | --- | --- |
| Strong Entity Symbol | Strong entity | These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity. |
| Weak Entity Symbol | Weak entity | Weak entities depend on some other entity type. They don't have primary keys, and have no meaning in the diagram without their parent entity. |
| Associative entity symbol | Associative entity | Associative entities relate the instances of several entity types. They also contain attributes specific to the relationship between those entity instances. |

**ERD relationship symbols**

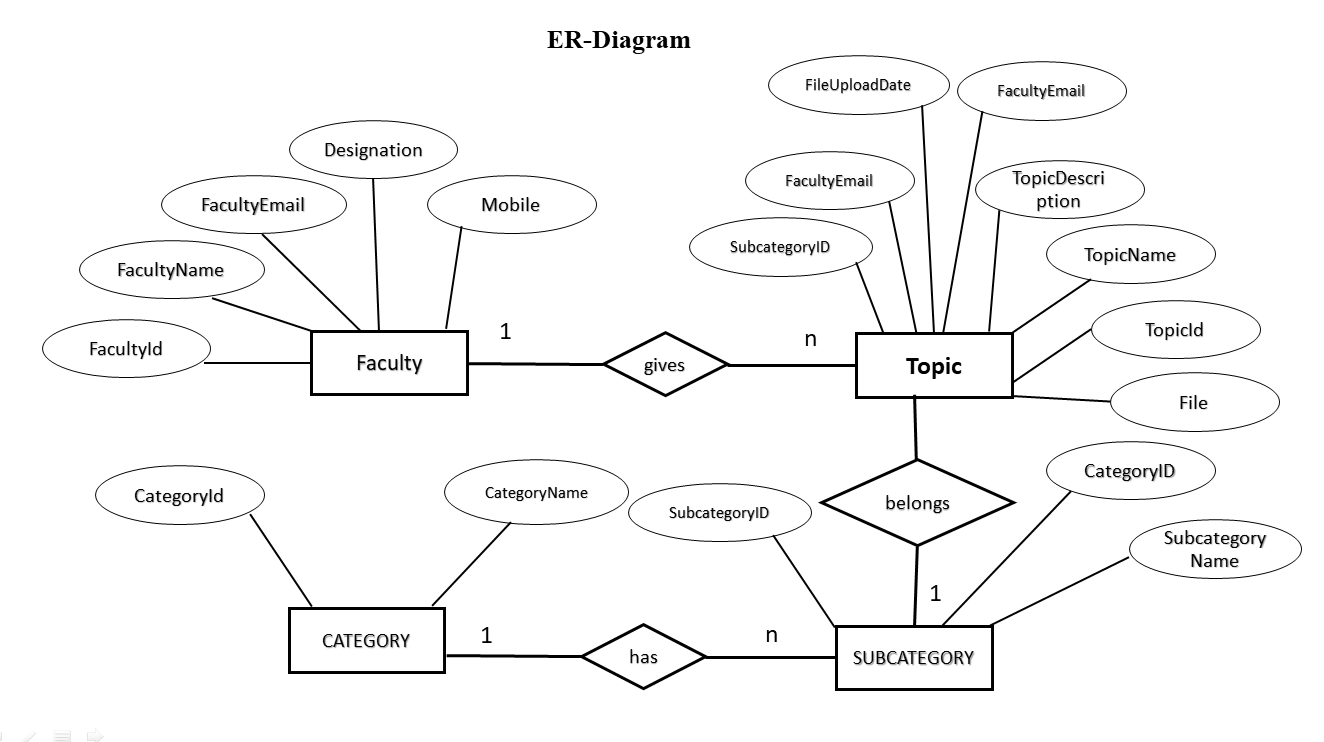
|  |  |  |
| --- | --- | --- |
| Relationship Symbol | Relationship | Relationships are associations between or among entities. |
| Weak Relationship Symbol | Weak relationship | Weak Relationships are connections between a weak entity and its owner. |

**ERD attribute symbols**

ERD attributes are characteristics of the entity that help users to better understand the database. Attributes are included to include details of the various entities that are highlighted in a conceptual ER diagram.

|  |  |  |
| --- | --- | --- |
| Attribute Symbol | Attribute | Attributes are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship. |
| Multivalued Attribute Symbol | Multivalued attribute | Multivalued attributes are those that are can take on more than one value. |
| Derived Attribute Symbol | Derived attribute | Derived attributes are attributes whose value can be calculated from related attribute values. |

**4.2. Entity-Relationship Diagram**

****

**4.3 Data Flow**

**Diagram (DFD)**

**Data Flow Diagram**

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modeling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the [visualization](https://en.wikipedia.org/wiki/Data_visualization) of [data processing](https://en.wikipedia.org/wiki/Data_processing).

A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured [flowchart](https://en.wikipedia.org/wiki/Flowchart) which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model.

Using any convention’s DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

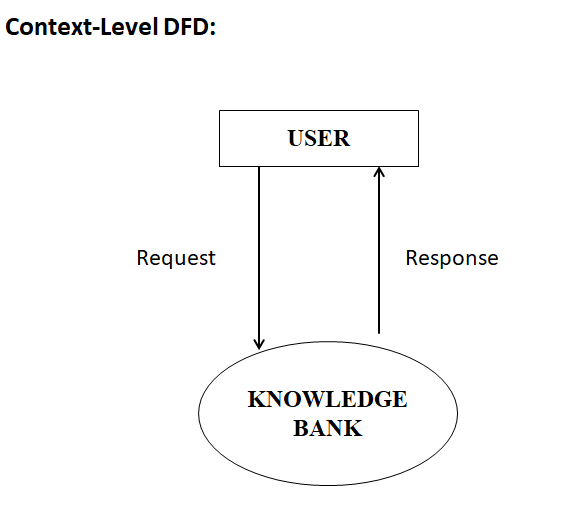
**a. External entity**: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

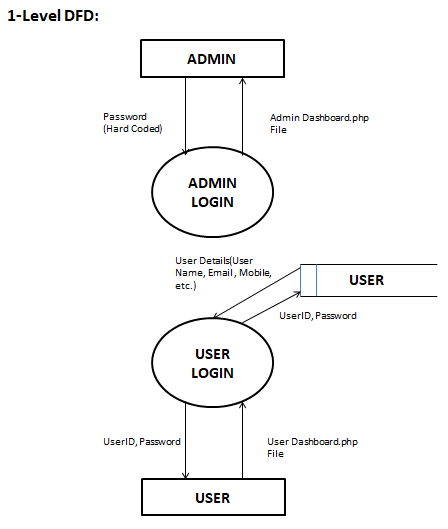
**b. Process**: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as “Submit payment.”

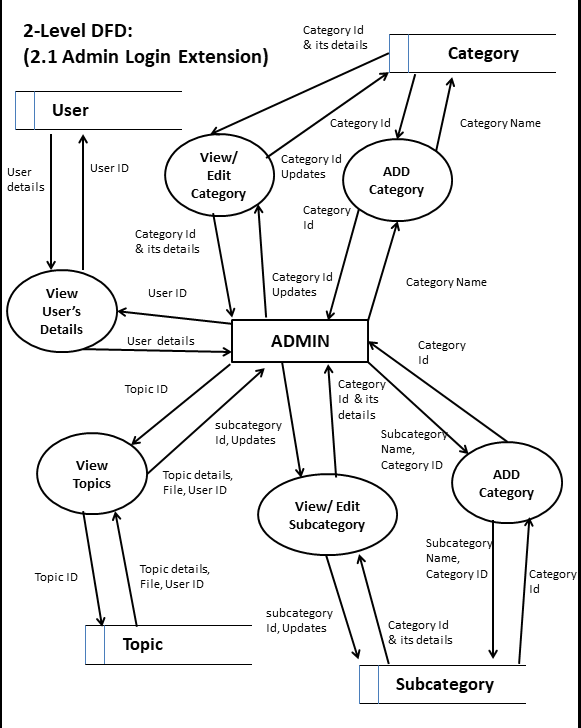
**c. Data store**: files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as “Orders.”

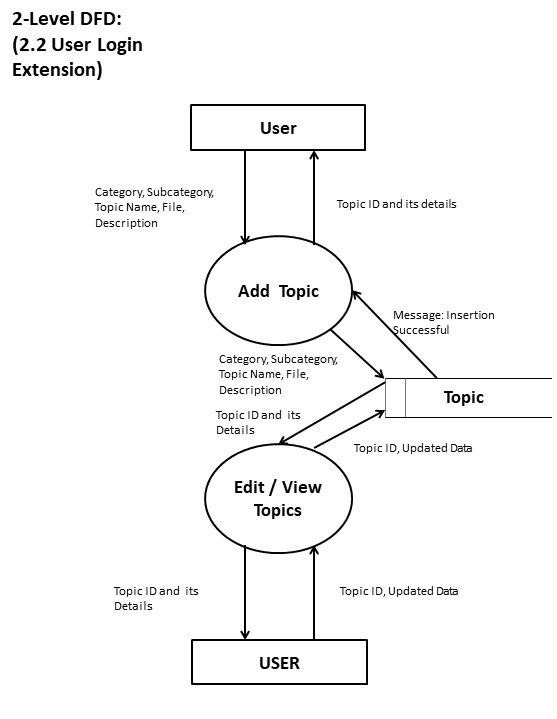
**d. Data flow**: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.”

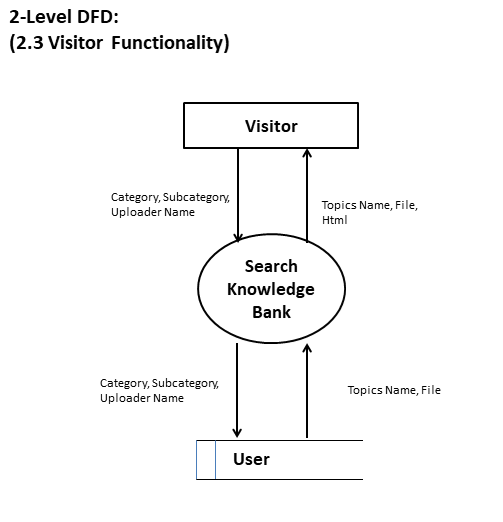
**Data Flow Diagram**

****

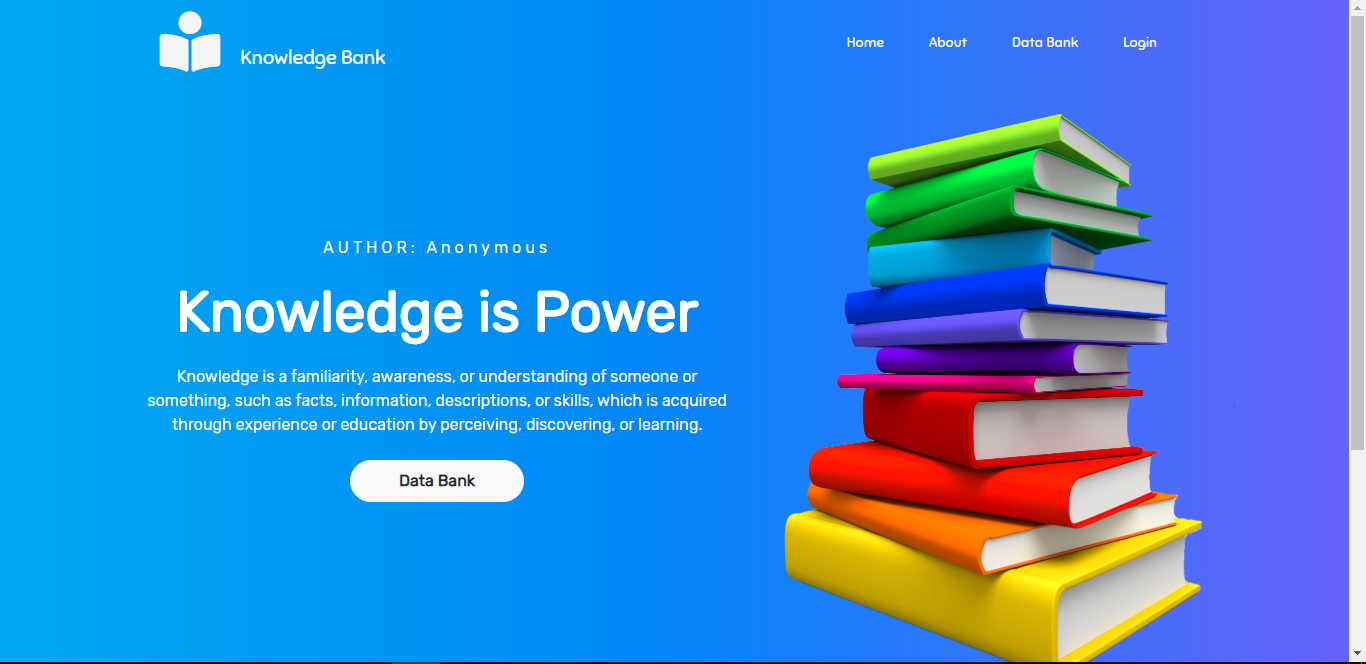
****

****

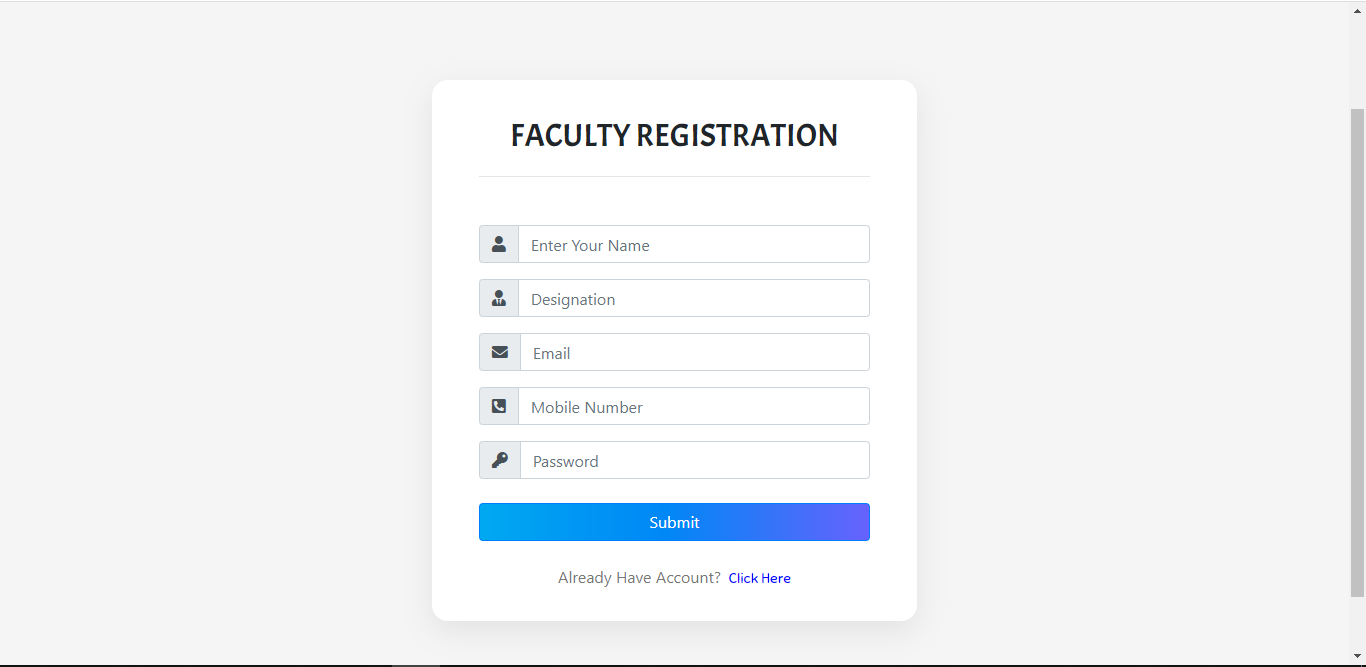
****

****

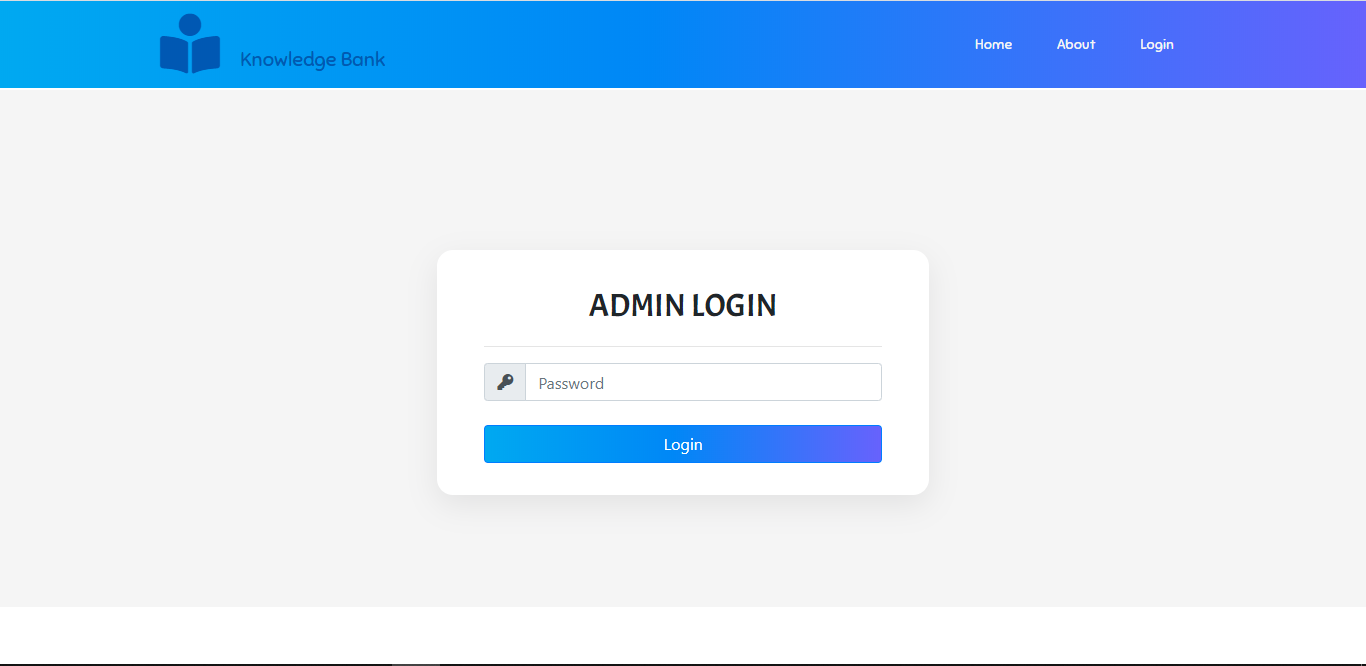
* 1. **Output Screen**
  2. **Home Page**



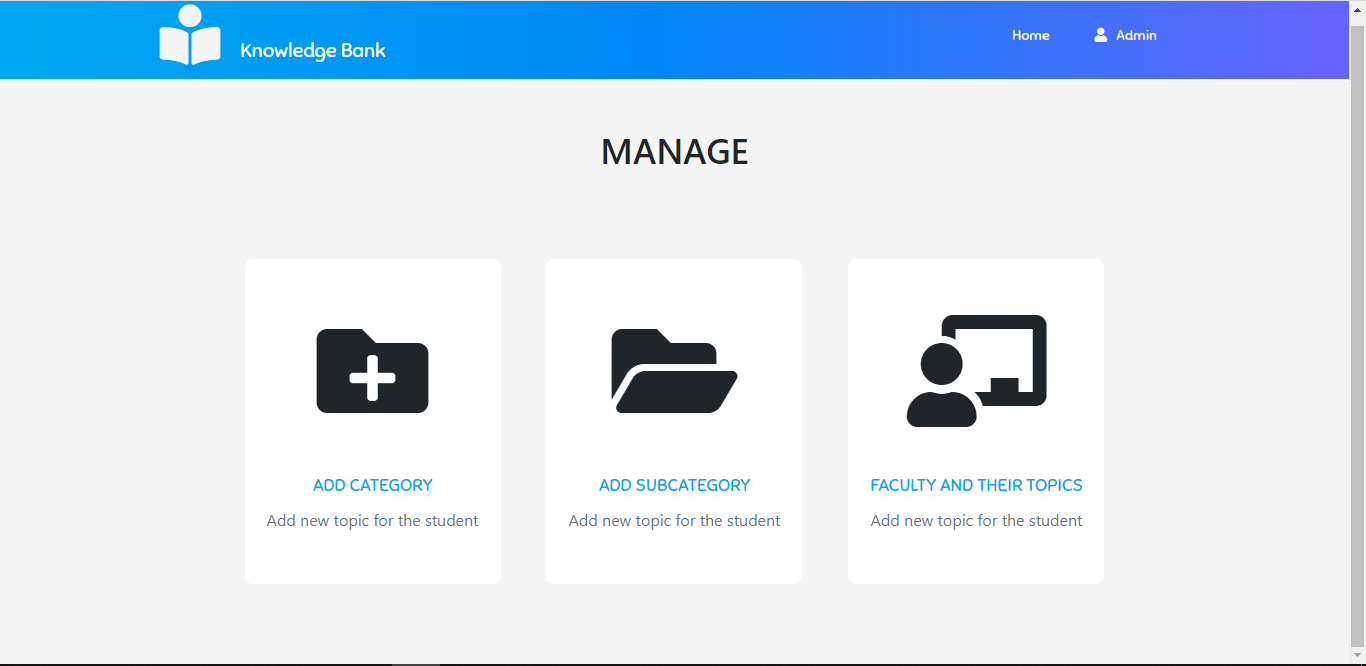
* 1. **Faculty Registration Page**



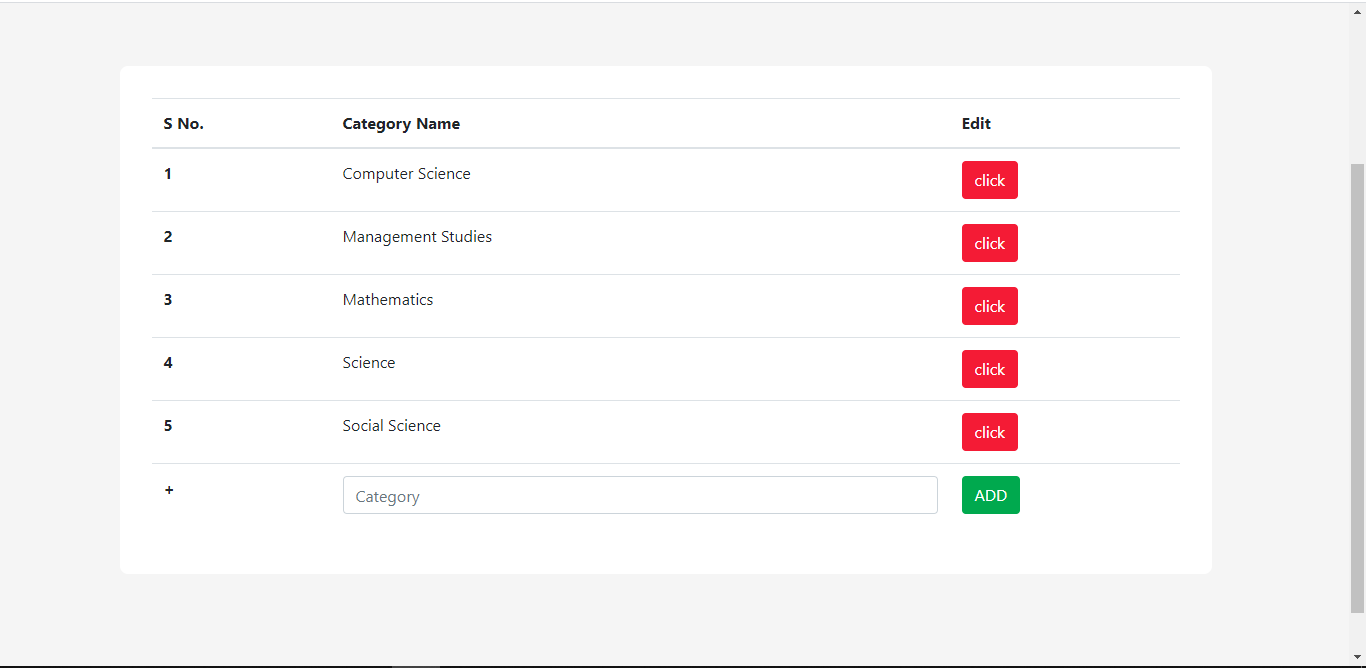
* 1. **Admin Login Page**



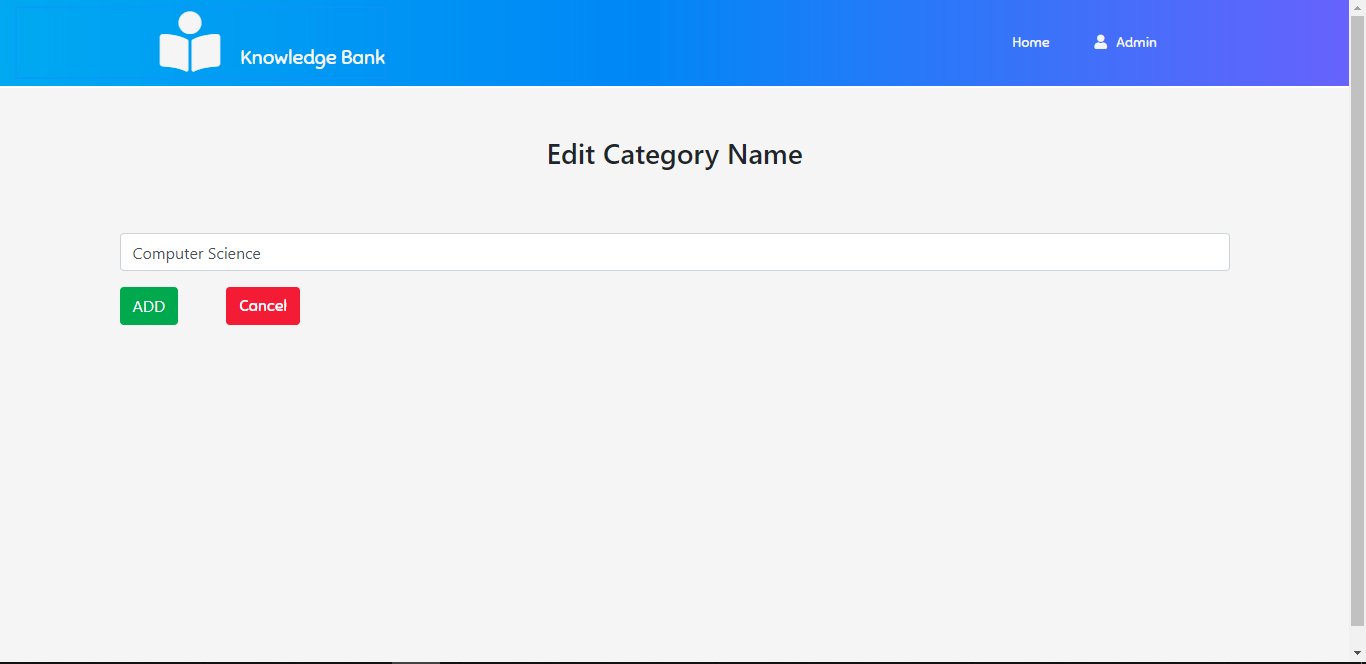
* 1. **Admin Dashboard**



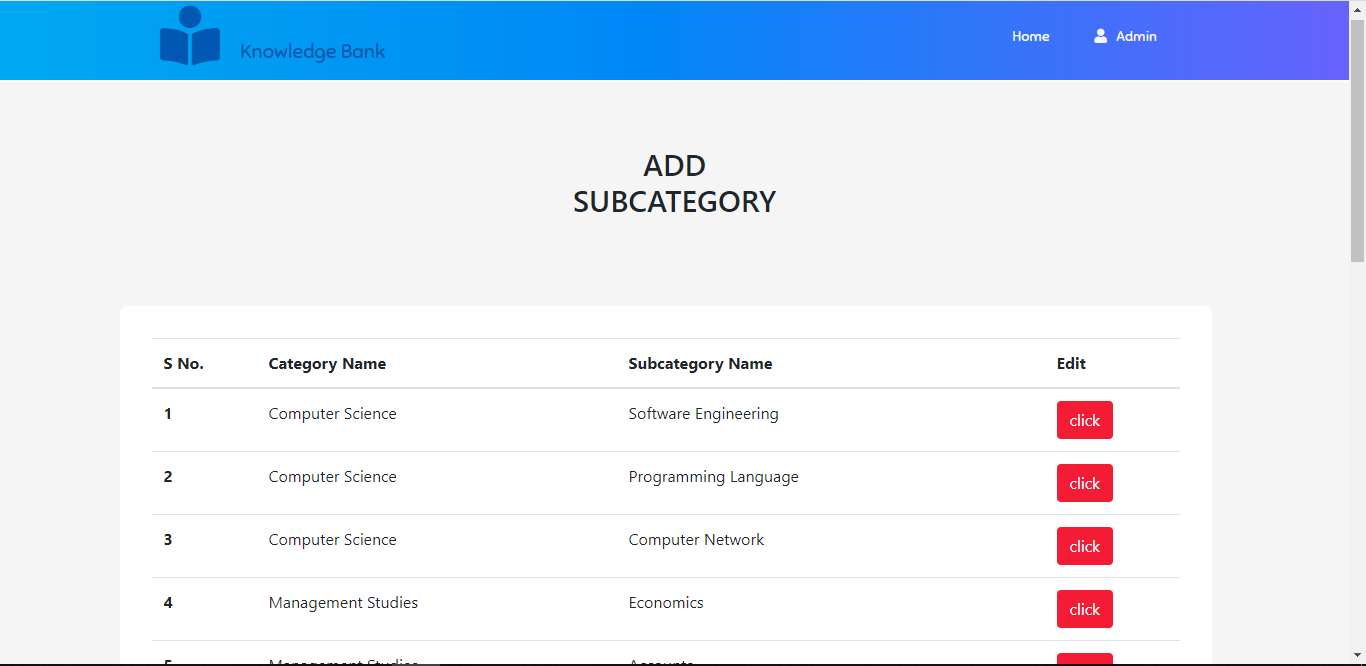
* 1. **Admin Add Category**



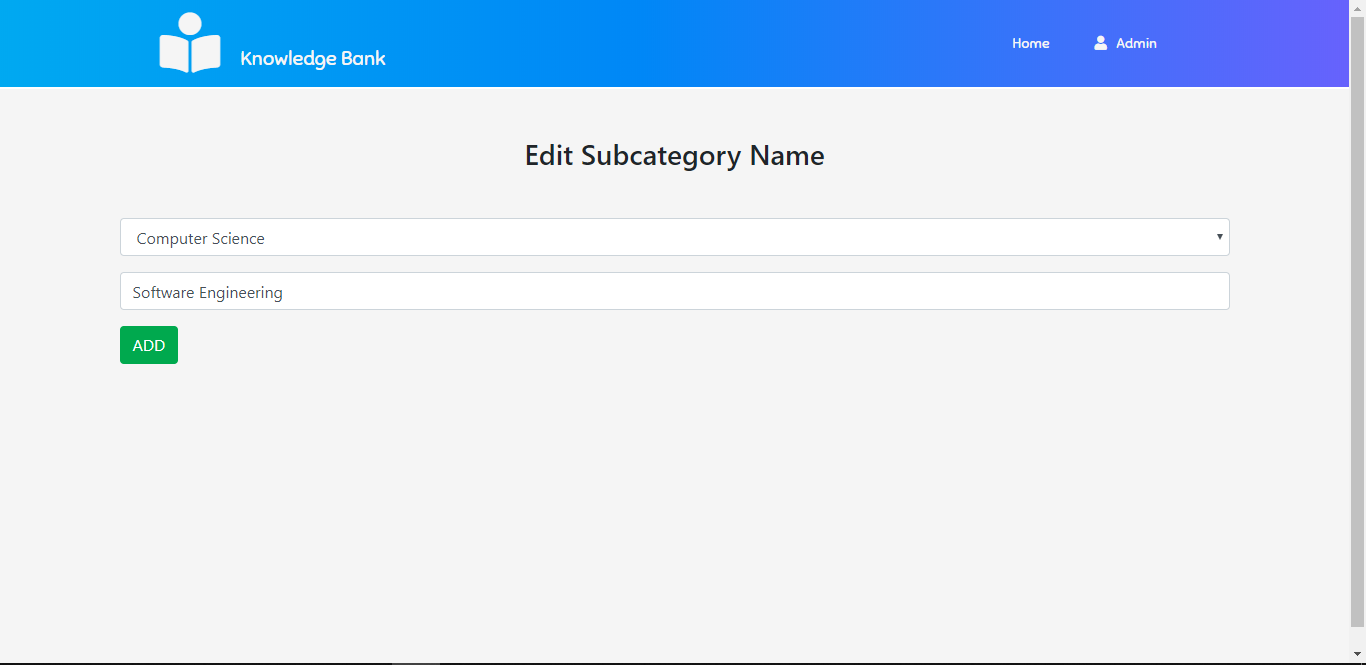
* 1. **Edit Category Name**



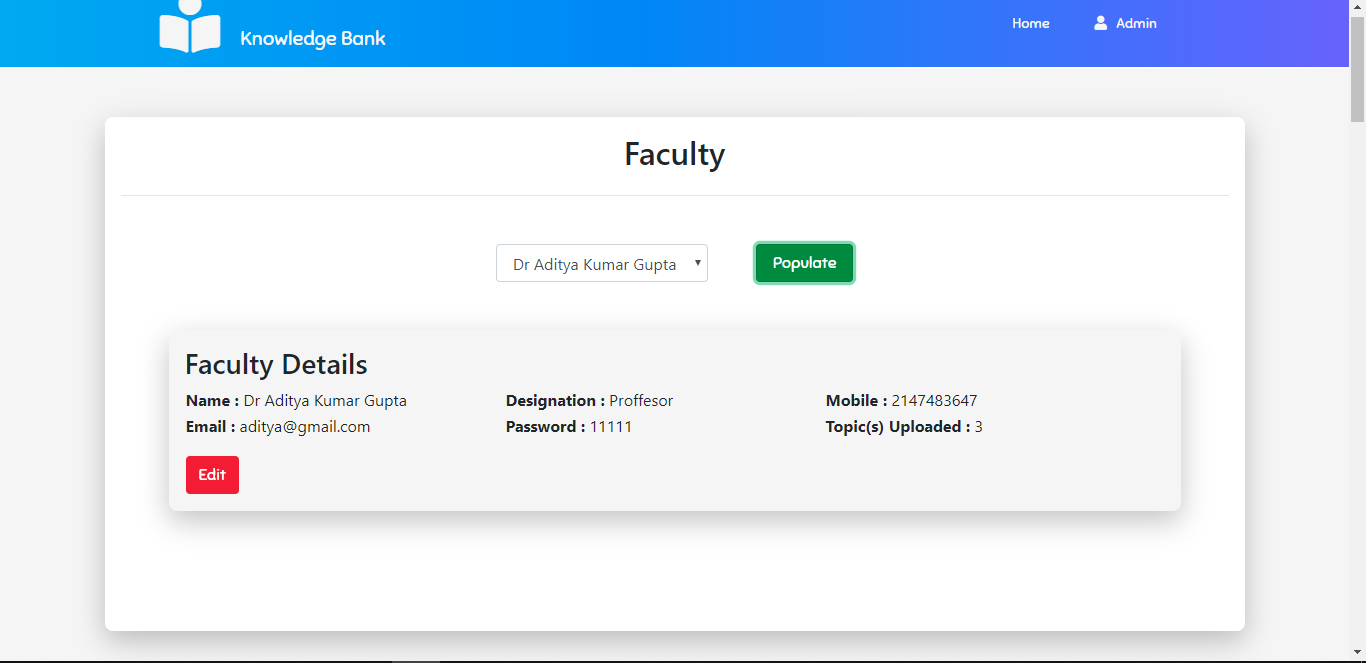
* 1. **Admin Add Sub-Category**

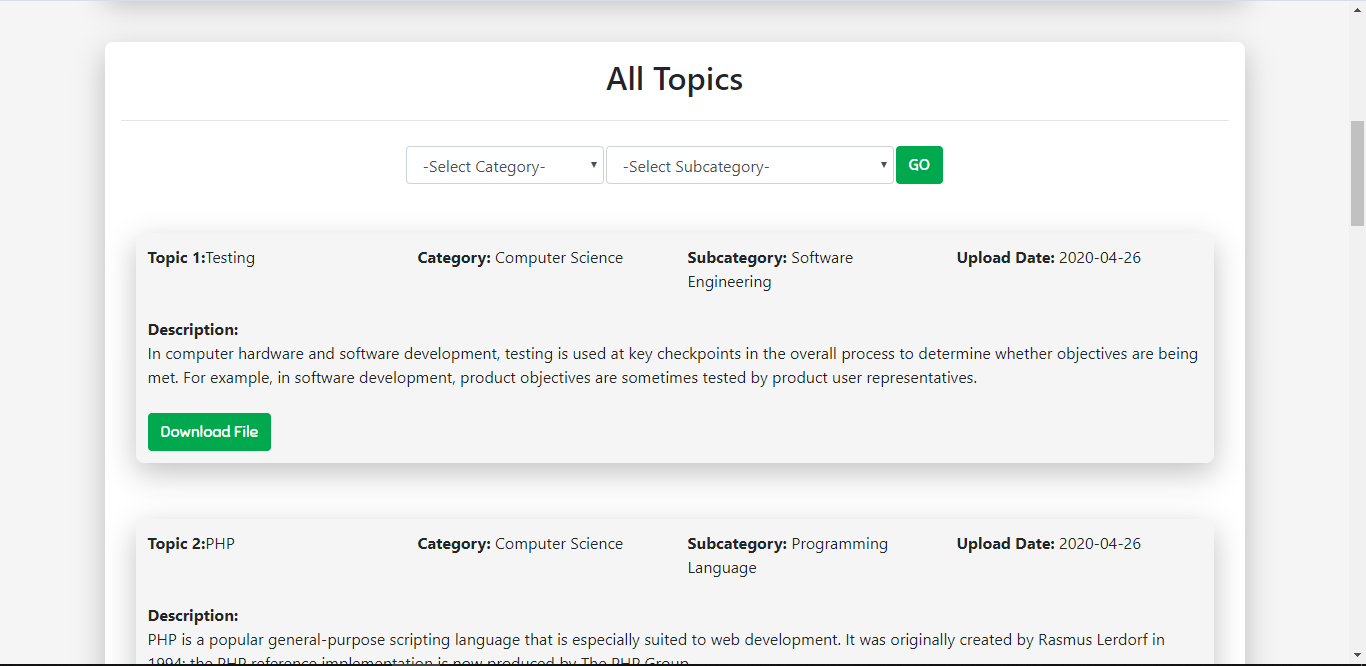


* 1. **Edit Subcategory Name**

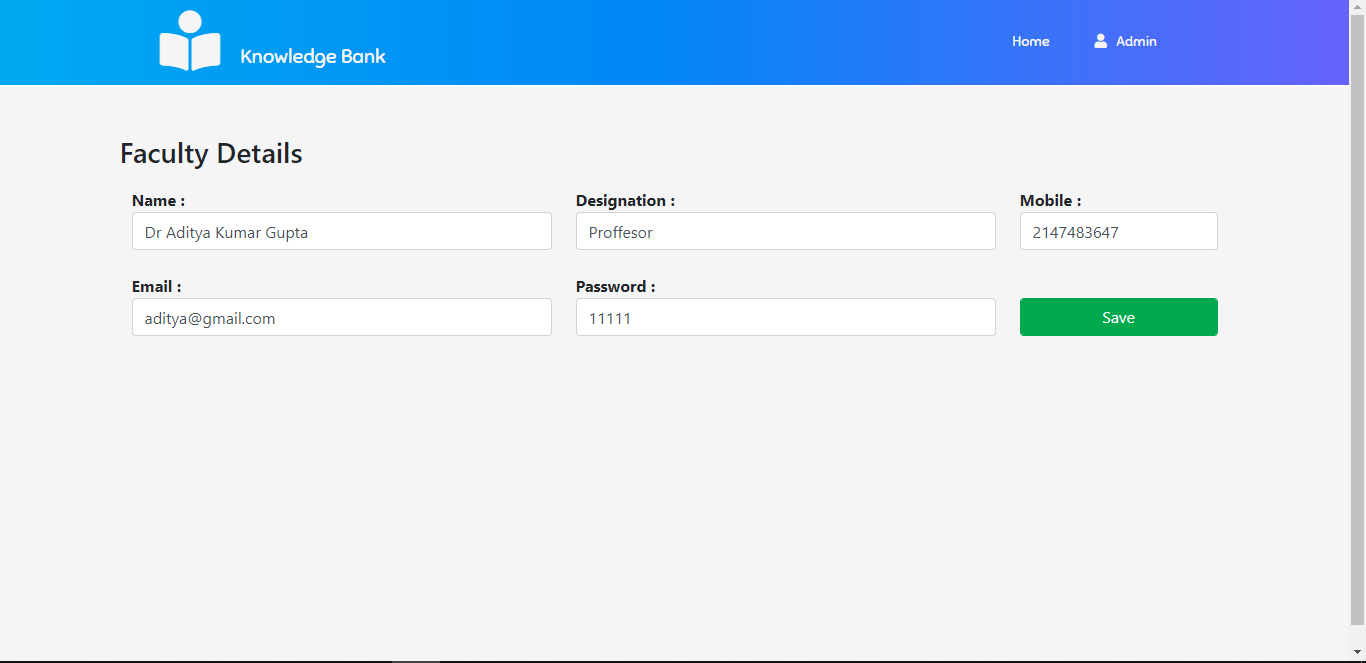


* 1. **Admin View Faculty and their Topics**

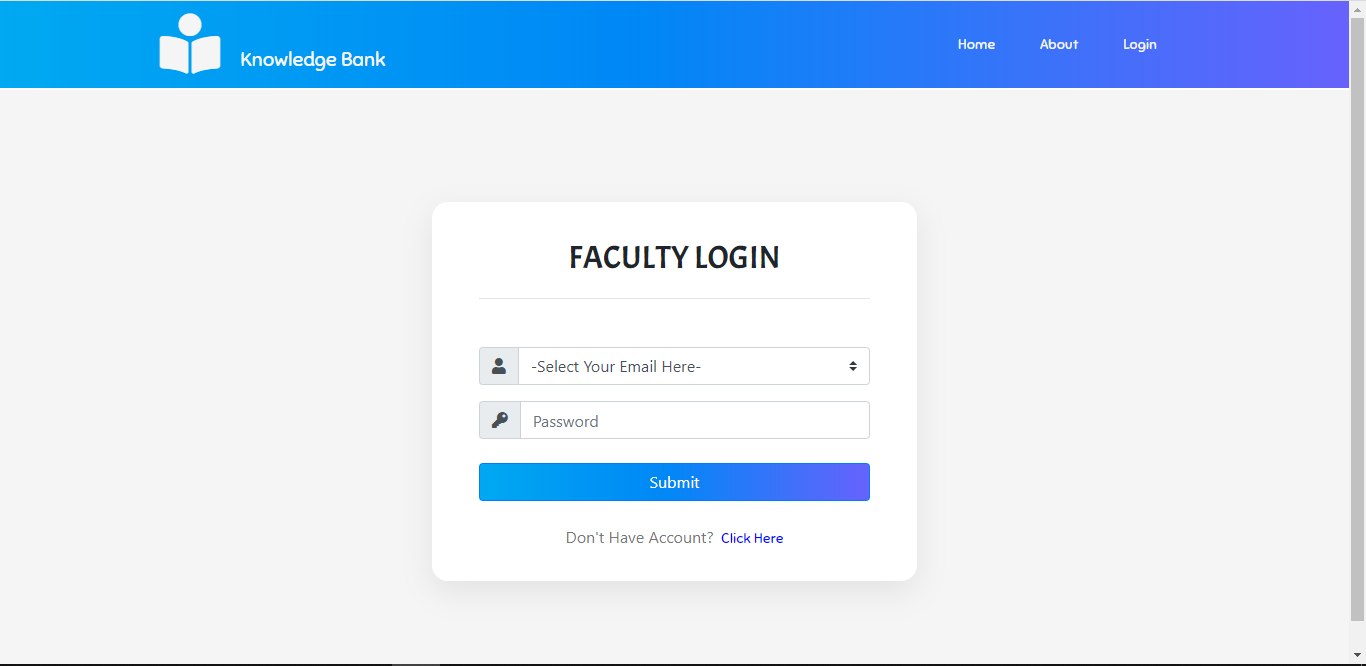




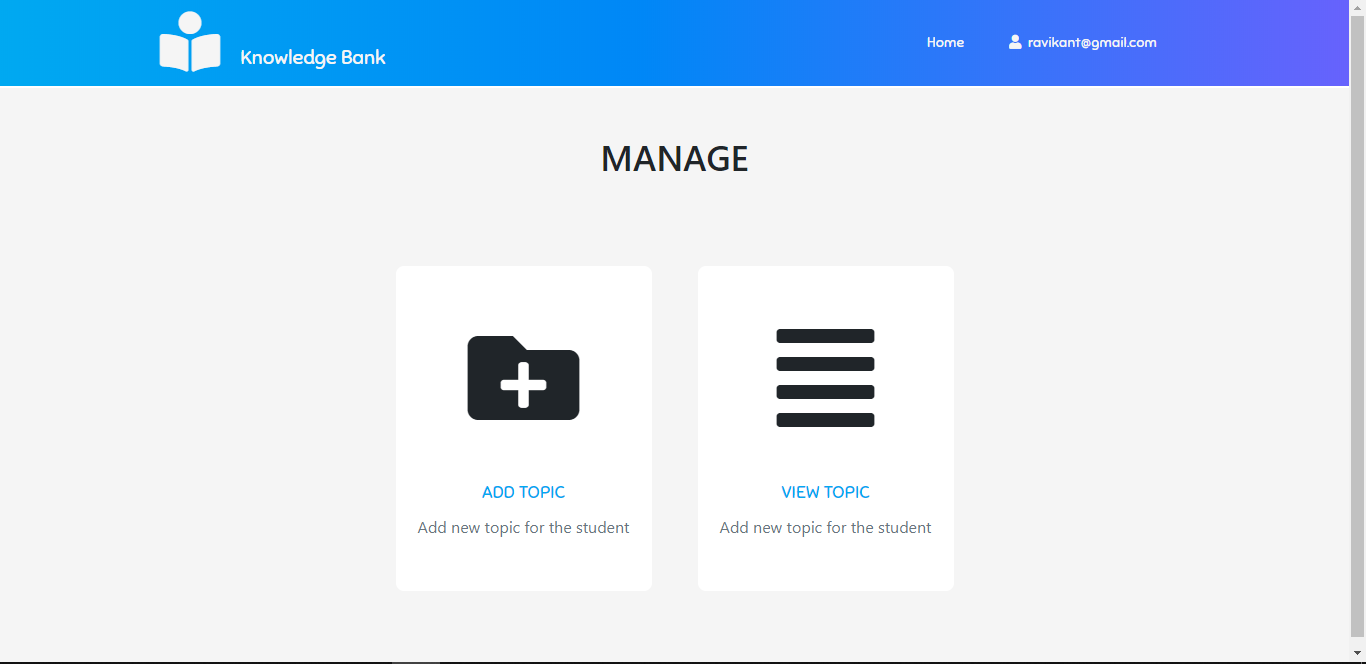
* 1. **Edit Faculty Details**



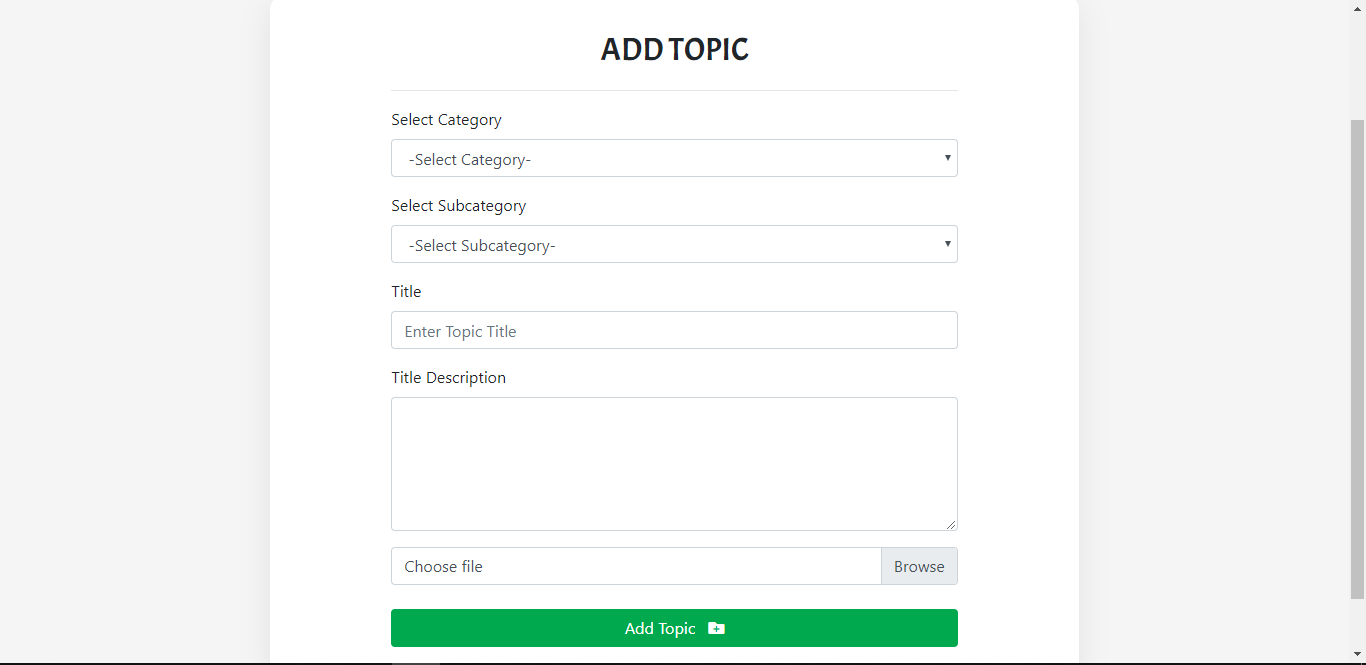
* 1. **Faculty Login Page**



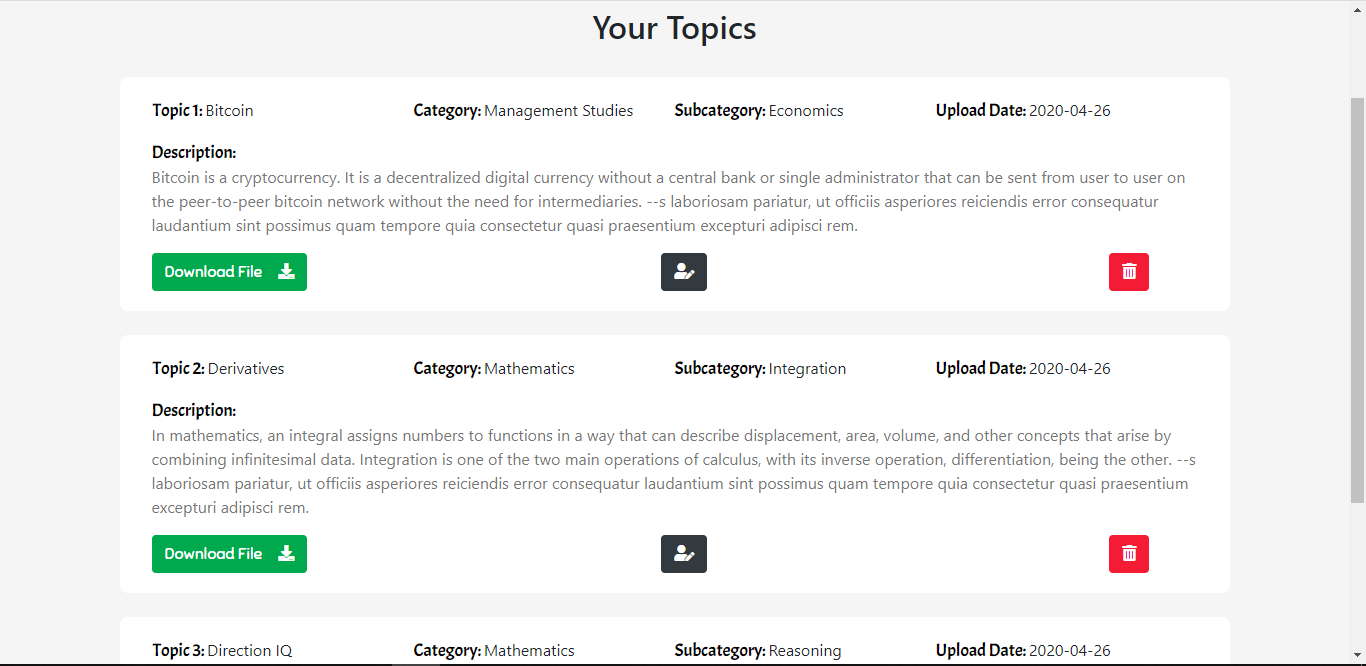
* 1. **Faculty Dashboard**



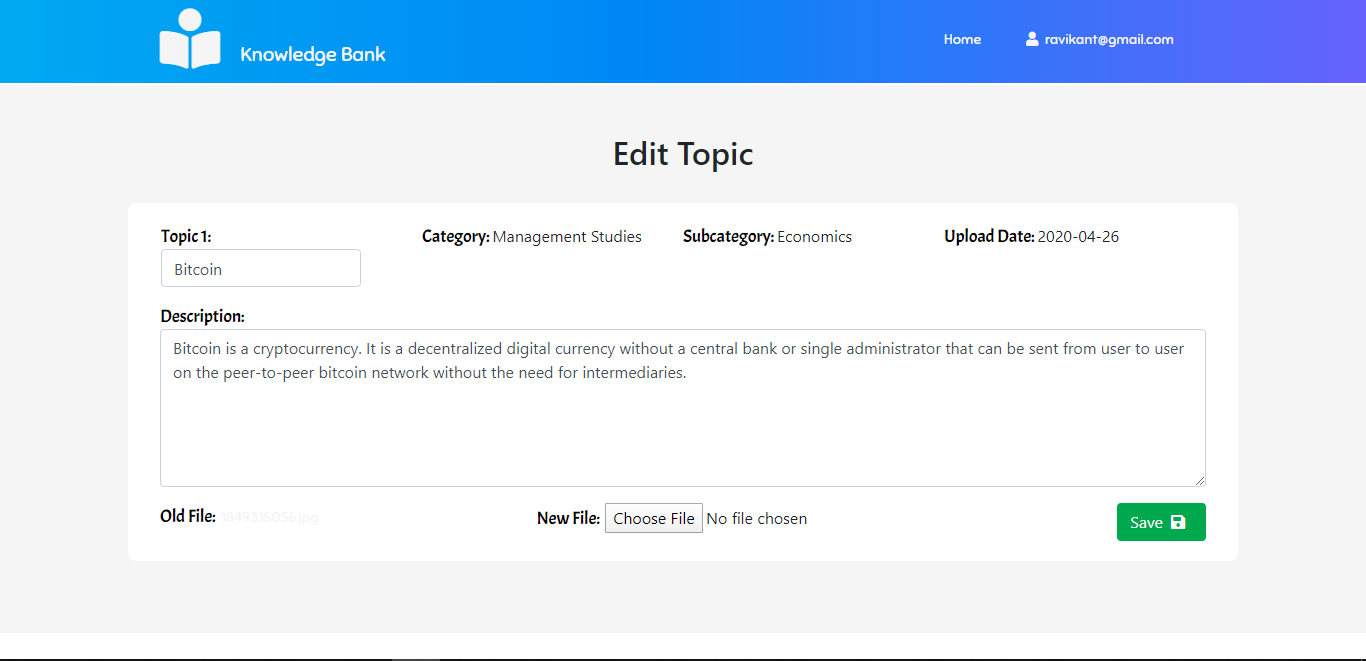
* 1. **Faculty Add Topic**



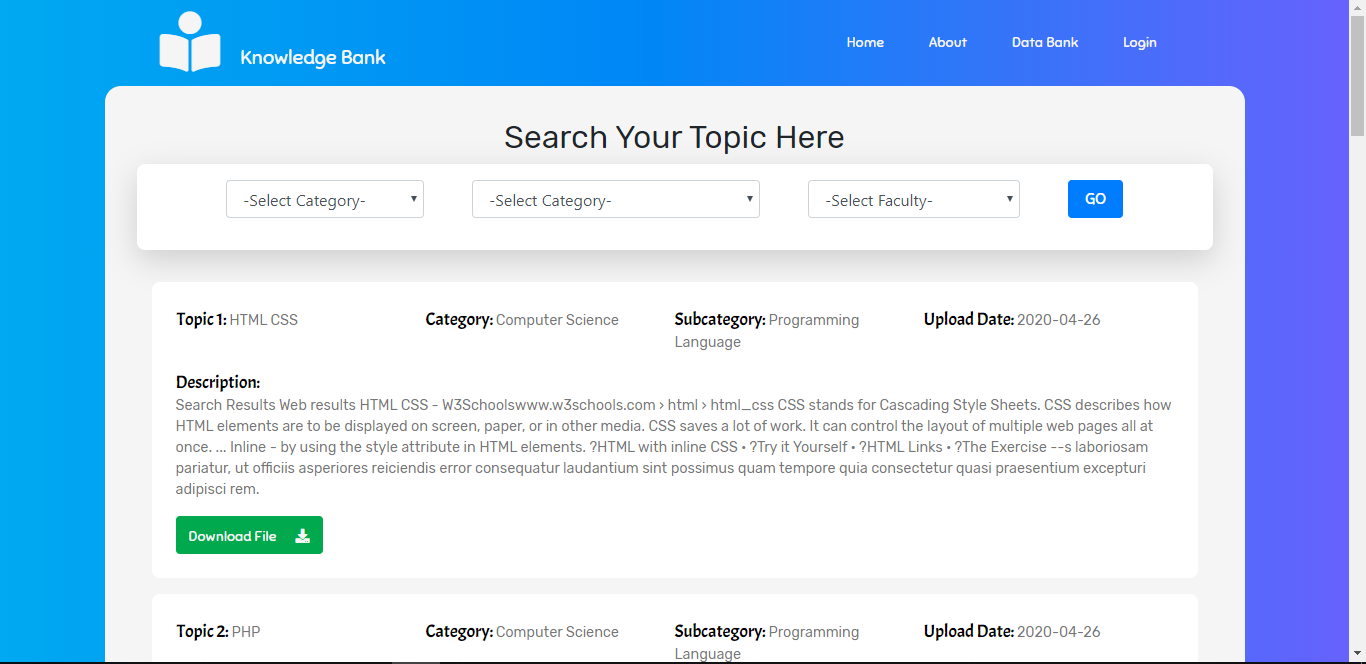
* 1. **Faculty View Topics**

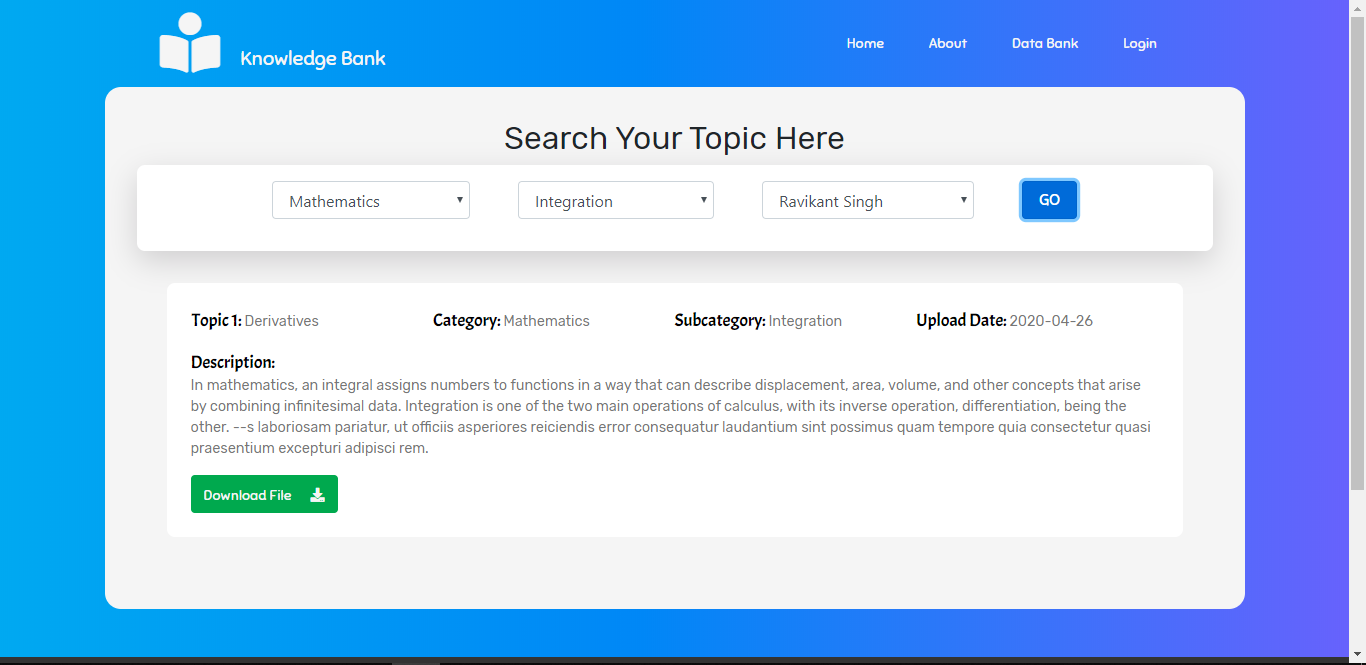


* 1. **Faculty Edit Topic**



* 1. **Data Bank**





**6. System Testing and Implementation**

**6.1** **Introduction of Testing**

System testing and implementation is the last step in software development. So this last chapter discusses the system testing and implementation. System testing is an essential step for the development of a reliable and error-free system. Once source code has been generated, software must be tested to uncover and correct as many errors as possible before delivery to your customer. Your goal is to design a series of test cases that have a high likelihood finding errors but how, there are different methods that provides a systematic guidance for designing tests that, Exercise the internal logic of software components, and Exercise the input and output domains of the program to uncover errors in the program function, behavior, and performance. The work product is a set of test cases designed to exercise both internal logic and external requirements is designed and documented, expected results are defined, and actual results are recorded. The primary objectives of testing software are to execute a program with the intent of finding an error; a good test case will find an as-yet-undercover error, and a successful that uncover an as-yet-undercover error.

**6.2 Testing Strategies**

The basic strategies that were used for testing were following

* Specification Testing
* Black Box Testing
* White Box Testing
* Regression Testing
* Acceptance Testing
* Assertion Testing
* Unit Testing
* System Testing

**Specification Testing**

Even if the code testing is performed exclusively, it doesn’t ensure against program failure. Code testing doesn’t answer whether the code meets the agreed specifications document.

It doesn’t also determine whether allaspects of the design are implemented.

Range of values expected including both valid and invalid data. It helps in finding discrepancies between the system and its original objective.

**Unit testing:**

Breaks down the software into components and verifies the functionalities of each

Individually within the programming environment.

**Integration Testing:**

This was done after successfully testing each module of software. Testing is done by creating interfaces between components and making sure that they communicate efficiently and that necessary input and output is transferred for the overall efficient functioning of the system.

**Acceptance Testing**:

Acceptance testing was done after the implementation of the system. The

Acceptance testing will check if the system works correctly in the user environment and if all

User specified functionalities are present.

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

**7. LIMITATIONS**

**Limitations**

1. **Upgraded Device/System.**

One should have upgraded System for using our techno logy, i.e. it will not work only on CUI based platforms.

2. **There is no provision for more than one Admin.**

**8. System Security**

**Security in Proposed System**

‘Knowledge Bank’ has provided the individual login to Admin and User, so that no one can breach into others confidential boundary. SMS Gateway is embedded so that all the users must authenticated via mobile verification.

**9. CONCLUSION**

**Conclusion**

The prime objective of the Knowledge bank is to provide a platform to the users to gain and share their knowledge. There is a “Give and Get” relationship.

**10. FUTURE**

**ENHANCEMENT**

**Future Enhancement**

1. It may include the blogging facility for uses.
2. SMS gateway can be included in this project for extra security and validation..
3. Android app can be developed for more handiness and comfortability.
4. There can be a module for admin to get register and more than admin can handle the Knowledge Bank.

**11. BIBILIOGRAPHY**

**Bibliography**

1. <http://www.jscripters.com/javascript-advantages-and-disadvantages/>
2. https://en.wikipedia.org/wiki/JavaScript
3. <http://www.jscripters.com/javascript-advantages-and-disadvantages/>
4. https://en.wikipedia.org/wiki/JavaScript
5. http://developmentcenter.blogspot.in/2011/03/characteristics-of-html-language.html
6. https://en.wikipedia.org/wiki/HTML
7. https://en.wikipedia.org/wiki/Apache\_Tomcat