

# PATENT & ACQUISITION SEARCH APPLICATION

A

*Mini project Submitted*

*In the partial fulfilment of the requirements for the  
award of the degree of*

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING



BY

KUNKUMALLA SRIYA	19281A0504
VOLLALA PRANAY	19281A0540
GADE ROHITH	19281A0550
KAVVA RASHMIKA	19281A0547

**PROJECT GUIDE**

**Mrs. R. BHARATHI**

(Assistant Professor, CSE)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE**

(Approved by AICTE, New Delhi, Affiliated to JNTU, Hyderabad, T.S,  
Accredited by NBA, NAAC with 'B++' Grade) Singapur, Huzurabad,  
Karimnagar, Telangana- 505468 (2022-2023)

# PATENT & ACQUISITION SEARCH APPLICATION

---

## **CERTIFICATE**

This is to certify that **K. SRIYA (19281A0504), V. PRANAY (19281A0540), G.ROHITH (19281A0550), K. RASHMIKA (19281A0547)** students of B. Tech (C.S.E) from Kamala Institute of Technology and Science , Singapur , Huzurabad , have continuously interacted with the management in gathering the requirements for the development of the website “**PATENT & ACQUISITION SEARCH APPLICATION**”. We are satisfied with the work done by the students. The developed website had satisfied all our requirements. We wish all the success for their future.

**Place:**

**Date:**

**Signature:**



# KAMALA INSTITUTE OF TECHNOLOGY & SCIENCE

(Approved by AICTE, New Delhi and Affiliated to JNTU, Hyderabad, T.S., Accredited by NBA, NAAC with 'B++' Grade)

SINGAPUR, HUZURABAD, KARIMNAGAR, TELANGANA, INDIA – 505468

## CERTIFICATE

This is to certify that **K. SRIYA (19281A0504), V. PRANAY (19281A0540), G. ROHITH (19281A0550), K. RASHMIKA (19281A0547)** of the B. Tech (CSE) has satisfactorily completed the dissertation work for Project Work entitled **“PATENT & ACQUISITION SEARCH APPLICATION”** towards the partial fulfilment of B. Tech degree in the academic year 2022-2023.

**PROJECT GUIDE:**

R. BHARATHI

**PROJECT CO-ORDINATOR:**

**HEAD OF THE DEPARTMENT:**

DR.K. RAVINDRA BABU

**EXTERNAL EXAMINER:**

# INDEX

<b>TITLE</b>	<b>PAGE NO</b>
<b>ACKNOWLEDGEMENT</b>	<b>i</b>
<b>ABSTRACT</b>	<b>ii</b>
<b>LIST OF FIGURES</b>	<b>iii</b>
<b>LIST OF TABLES</b>	<b>iv</b>
<b>Chapter 1: Introduction</b>	<b>1</b>
1.1 About Project	1
1.2 Existing System with Drawbacks	1
1.3 Proposed System	1
<b>Chapter 2: Analysis</b>	<b>2</b>
2.1 Hardware & Software Requirements	2
2.2 Functional & non-Functional Requirements	3
2.3 Module Description	4
<b>Chapter3: Design</b>	<b>5</b>
3.1 Block Diagram	5
3.2 Data Flow Diagram	6
3.3 ER Diagram	9
3.4 UML Diagrams	11
3.4.1 Use case Diagram	12
3.4.2 Class Diagram	14
3.4.3 Sequence Diagram	15
3.4.4 Activity Diagram	16
3.5 Database Dictionary	18

<b>Chapter 4: Implementation</b>	<b>19</b>
4.1 Languages Used	19
4.2 Frontend Design	27
<b>Chapter 5: Testing</b>	<b>32</b>
5.1 Testing	32
5.2 Testing strategy	36
5.3 Black Box Testing	37
5.4 White Box Testing	38
5.5 Database testing	39
5.6 Acceptance Testing	40
5.7 Test Results	40
<b>Chapter 6: Results</b>	<b>41</b>
6.1 Screenshot for Homepage	41
6.2 Screenshot for Login successfully	42
6.3 Screenshot for Add Company	43
6.4 Screenshot for Company Events	44
<b>Chapter 7: Conclusion</b>	<b>45</b>
<b>Chapter 8: Future Scope &amp; Enhancements</b>	<b>46</b>
<b>Chapter 9: Bibliography</b>	<b>47</b>

# ACKNOWLEDGEMENT

The success of any course depends mainly on the teachers who teach us. Only good teaching can interpret the syllabus and procedure desirable changes and competent citizens. This project was a team effort and many people whose names do not appear on the cover also deserve credit. First, we thank God almighty for his manifold mercies in carrying out our project successfully.

We would like to pay our respects to our principal **Dr. K. SHANKER** and management for providing all the facilities required for completing this project work.

We sincerely thank our Head of the Department, Computer Science & Engineering, **Dr. Capt. RAVINDRA BABU**, for supporting us throughout the completion of the project work.

We sincerely thank our Guide **Mrs. R. BHARATHI**, for encouraging us in doing this project and for her guidance.

We would like to thank our project coordinator **Mrs. R. BHARATHI**, Assistant Professor in CSE for her valuable guidance and constant encouragement at each stage of this project.

We thank all teaching and non-teaching staff of Kamala Institute of Technology & Science, for supporting us in every stage of our Mini Project, entitled “**PATENT & ACQUISITION SEARCH APPLICATION.**”

We thank our Parents and Friends for their moral support throughout the project work that helped to strengthen our will.

<b>K. SRIYA</b>	<b>19281A0504</b>
<b>V. PRANAY</b>	<b>19281A0540</b>
<b>G. ROHITH</b>	<b>19281A0550</b>
<b>K. RASHMIKA</b>	<b>19281A0547</b>

# **ABSTRACT**

The sole purpose of this project is to develop a "Patent & Acquisition Search Application" which will help companies to know the type of patents filed and acquisitions made covering specific technology, application, industry areas by similar companies in IT domain in last 3 - 5 years. We provide an easy-to-use interface for any specific company in IT domain to access the application and know the patents filed and acquisitions made by other companies in the same domain.

Now-a-days every company is focusing on adoption of emerging digital technologies like AIML, AR/VR Cyber security, etc. to transform the legacy systems, products and provide them to end-users thus, helping the companies to capture the market share to a maximum extent.

One of the mechanisms to know the technology, business focus of any company is to look-out for the patents filed and acquisitions made in last 3-5 years. To provide this information, our application is very useful, and any company can access the application to extract such data which will in-turn helps to frame out the technology, business strategy to outperform the competitors in similar space.

## **LIST OF FIGURES**

<b>TITLE</b>	<b>PAGE NO</b>
3.1 Block Diagram	5
3.2 DFD Diagram	8
3.3 ER Diagram	10
3.4.1 Use Case Diagram	13
3.4.2 Class Diagram	15
3.4.3 Sequence Diagram	17
3.4.4.1 Activity Diagram Admin	18
3.4.4.2 Activity Diagram User	19



## **LIST OF TABLES**

3.5.1 Patent Table	20
3.5.2 Acquisition Table	20
3.5.3Login Table	21
3.5.4 Company Table	21
3.5.5 Contact Table	22
5.1.1 Admin Login Test Table	39

# CHAPTER 1

## INTRODUCTION

### 1.1 About the project

The purpose of this project is to develop the website of patents filed by a company. It is a process of collecting the patent filed by the company and we connect them in a backend of the website. The company will register, and admin can access an authorization. So that admin need not register again. In the frontend we must develop the webpage, template based on the requirements.

By using the website companies can search for the yearly patent on the scripting languages like Python, AIML, DBMS, MYSQL etc. filed by the company, and it can reduce the time. The acquisition search for the company patents and gather the information. So, in order to save time we created an application based on PHP as “PATENT & ACQUISITION SEARCH APPLICATION” So by using this application we can search for the patent filed by company.

### 1.2 Existing System with Drawbacks

In the present system the user needs to search for patents in different search engines. But the user can't access all patent data of top MNC companies from the browser, search engines. If the user wants to access only specific company the search will not provide the exact number patents filed by the company & the complete data will not be uploaded in browser.

#### **Drawbacks:**

- No acquisition patent data
- User need search different websites for different patents

### 1.3 Proposed System:

The main goal of our website is to list of patent data of MNC companies in the order. We newly added the acquisition data by this the user can also access to the patents owned by start-ups companies. Here the companies filed patents are shortlisted according to the user choices from the shortlisted patents the user read the patents. (with his own interest). There is an admin login to change or add the data in the database or excel sheet.

The following features of proposed system:

- The website can be proved to be very beneficial in patent data user can views the patents and acquisitions whenever they needed
- The user can also gain information and he can also download the patent
- This is the web-based applications which will save the user time.

# CHAPTER 2

## ANALYSIS

The goal of system analysis is to determine where the problem is in an attempt to fix the system. This step involves breaking down the system in different pieces to analyse the situation, analysing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

### 2.1 Hardware and Software requirements

**Hardware requirements:** The following are the hardware requirements which we have used in our project.

- Processor Needed : i3 or above.
- RAM : 1 GB or more.
- Hard disk : 40 GB or more.
- Monitor : Any Monitor.
- Keyboard : Standard Keyboard.
- Mouse : Two or Three Button Mouse.

**Software Requirements:** The following are software requirements. Technologies are specified by the client.

- Operating System : Windows XP, 7 or Higher windows OS.
- Scripting Language : PHP.
- Front – End : HTML, CSS, JavaScript, and Bootstrap.
- Back – End : MySQL.
- Web Server : XAMPP Server.

## 2.2 Functional & Non-Functional Requirements:

### Functional Requirements:

Functional requirements are associated with specific functions, tasks or Behaviours of the system. The functional requirements address the quality characteristic of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements. Because non-functional requirements tend to be stated in terms of constraints on the results of tasks which are given as functional requirements (e.g., constraints on the speed or efficiency of a given task), a task-based functional requirements statement is a useful skeleton upon which to construct a complete requirements statement.

This database describes the following:

- Maintaining and Updating report.
- Details of the customer, products, sales, invoices.
- View the details through the application connected to the database.
- Modification of data if necessary.

### Non-Functional Requirements:

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that specify specific behaviour or functions.

#### Following are the non-functional requirements:

**Consistency:** The application provides consistent user interface design to user. The designs of the screen are standardized and consistent that makes the user feel comfortable to use it.

**Convenience:** The application gives convenience to the user to store all the details through the application connected to the database. Also facilitates the admin to modify the details if required and can view the details.

**Availability:** The content must be available to authorized user. This website provides admin to login and view the details of the product.

**Security:** Administrators can only perform administrative tasks on pages they are privileged to access. User will not be allowed to access the administrator pages. This site provides password access control to avoid unauthorized admin to login.

**Reliability:** The application provides an effective method to maintain the back-end to store all the details securely. All details are managed by this application effectively.

**Size:** The performance of the project depends on the size of the project. We put lots of effort in reducing lines of code. In this project the storage space is utilized efficiently.

**Scalability:** Scalable software can remain stable while adapting to changes, Upgrades, overhauls and resource reduction. Scalability is an attribute of a tool or a system to increase its capacity and functionalities based on its user's demand.

## 2.3 Module Description

The modules used in this system are:

- Admin
- General user

1. **Admin:** This class is used for getting users and product information from the database and it is also used to update the database with the information about new-user registration, product checkout, and user details.

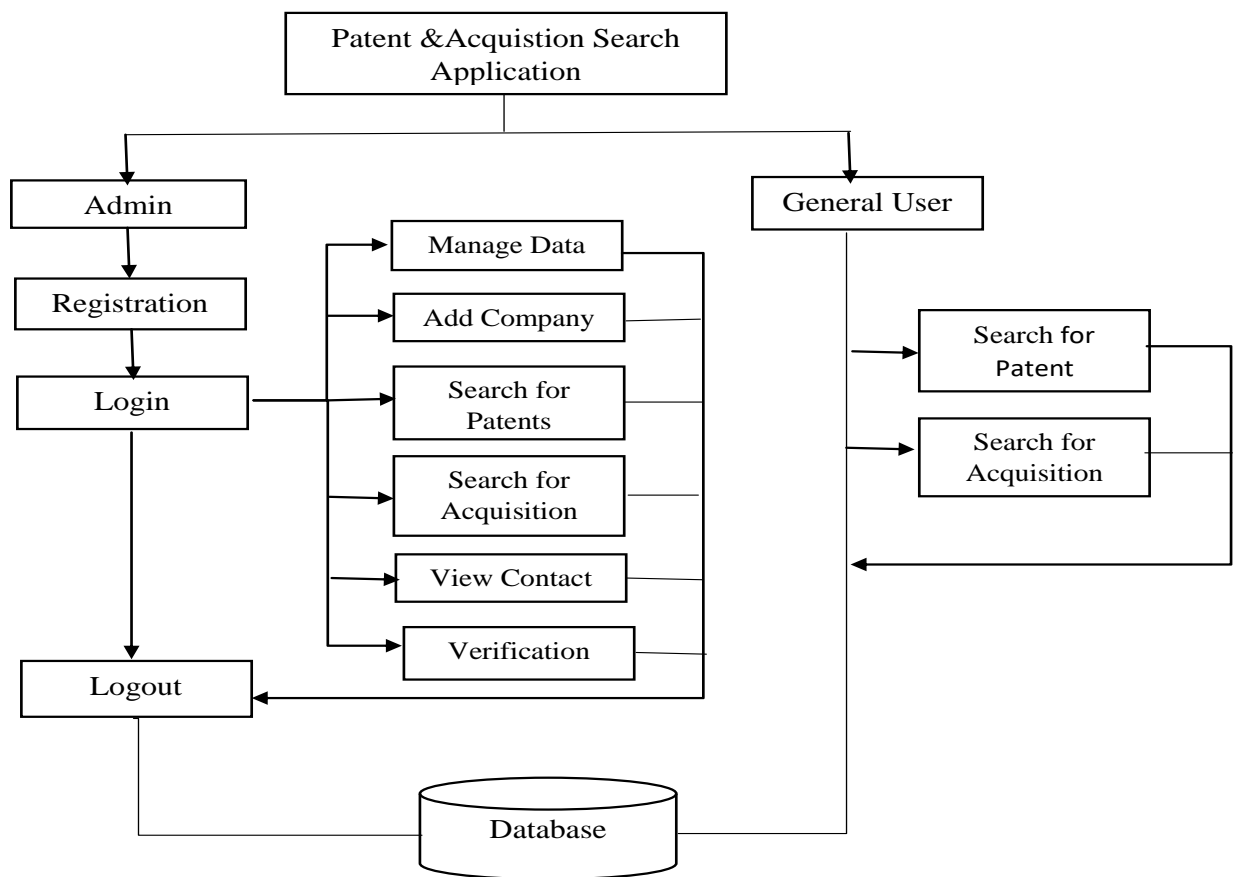
2. **General user:** A user who has both registered an individual Profile in our Portal and established an association with an Organisation Profile.

# CHAPTER 3

## DESIGN

### 3.1 Block Diagram:

The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of Implementation [5]. Main operations are to add, view, update and delete the details of the patent, acquisition. The following figure (3.1) represents the block diagram.



**Figure 3.1: Block Diagram**

### 3.2 Data flow diagrams:

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process. A DFD [1] provides no information about the timing of processes, or about whether processes will operate in sequence or in parallel.

It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored. It is common practice to draw a context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modelled purely in terms of data flows across the system boundary. The context diagram shows the entire system as a single process and gives no clues as to its internal organization. This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modelled [2]. The Level 1 DFD shows how the system is divided into subsystems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.

It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system. Data-flow diagrams were invented by Larry Constantine, the original developer of structured design, based on Martin and Estrin's "data-flow graph" model of computation. Data-flow diagrams (DFDs) are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data-flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented.

The old system's data flow diagrams can be drawn up, compared with the new system's Data-flow diagrams to draw comparisons to implement a more efficient system. Data-flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately influences the structure of the whole system from order to dispatch to report.

How any system is developed can be determined through a data-flow diagram. In the Course of developing a set of levelled dataflow diagrams the analyst/designer is forced to address how the system may be decomposed into component subsystems, and to identify the transaction data in the data model.

There are different notations to draw data-flow diagrams, defining different visual representations for processes, data stores, data flow, and external entities. Data flow diagrams ("bubble charts") are directed graphs [3] in which the nodes specify processing activities and the arcs specify data items transmitted between processing nodes. Below Figure (3.2) represents the DFD for managing product.

### **DFD Symbols**

In the DFD, there are four symbols

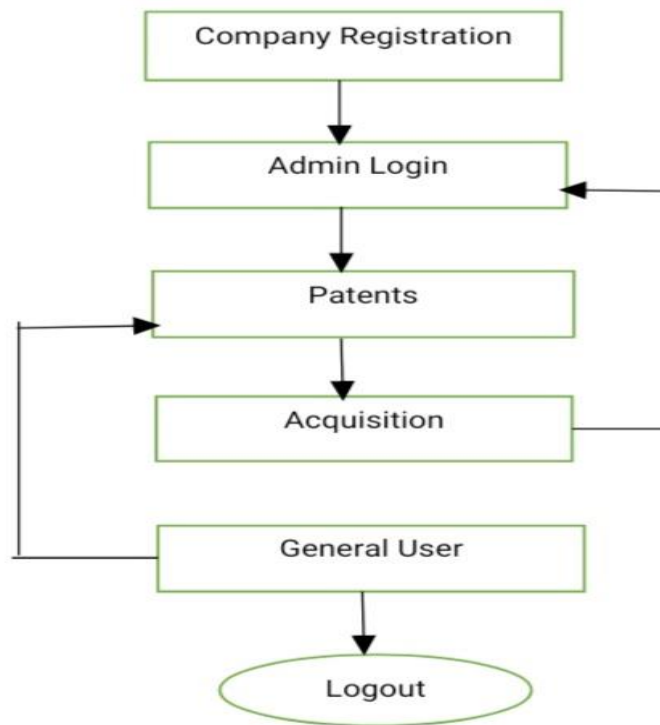
- A square defines a source (originator) or destination of system data.
- An arrow identifies data flow. It is the pipeline through which the information flows.
- A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
- An open rectangle is a data store, data at rest or a temporary repository of data.

### **Constructing a DFD:**

Several rules of thumb are used in drawing DFD'S:

- Processes should be named and numbered for an easy interface. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flows from source to the destination although they may flow back to the source. One way to indicate this is to draw a long flow line back to a source.
- An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.





**Figure 3.2: DFD Diagram**

### 3.3 ER Diagram:

In software engineering, an entity-relationship model (ER model) is a data model for describing data or information aspects of a business domain or its process requirements. In an abstract way the main components of ER model are entities and the relationships that can exist among them.

#### Elements in ER diagram

There are three basic elements in an ER Diagram.

- Entity
- Attribute
- Relationship

There are more elements which are based on the main elements [4]. They are weak entities, multi valued attributes, derived attributes, weak relationships and recursive relationships. Cardinality is one of the notations used in ER diagrams.

**Entity:** An entity can be a person, place, event or object that is relevant to a given system. They are represented by a rectangle and named using nouns.

**Weak Entity:** A weak entity is an entity that depends on the existence of another entity. It can be defined as an entity that cannot be identified by its own attributes.

**Attribute:** An attribute is a property, or characteristic of an entity, relationship, or another attribute.

**Multi valued Attribute:** If an attribute can have more than one value it is called a multi valued Attribute.

**Derived Attribute:** An attribute derived from another attribute.

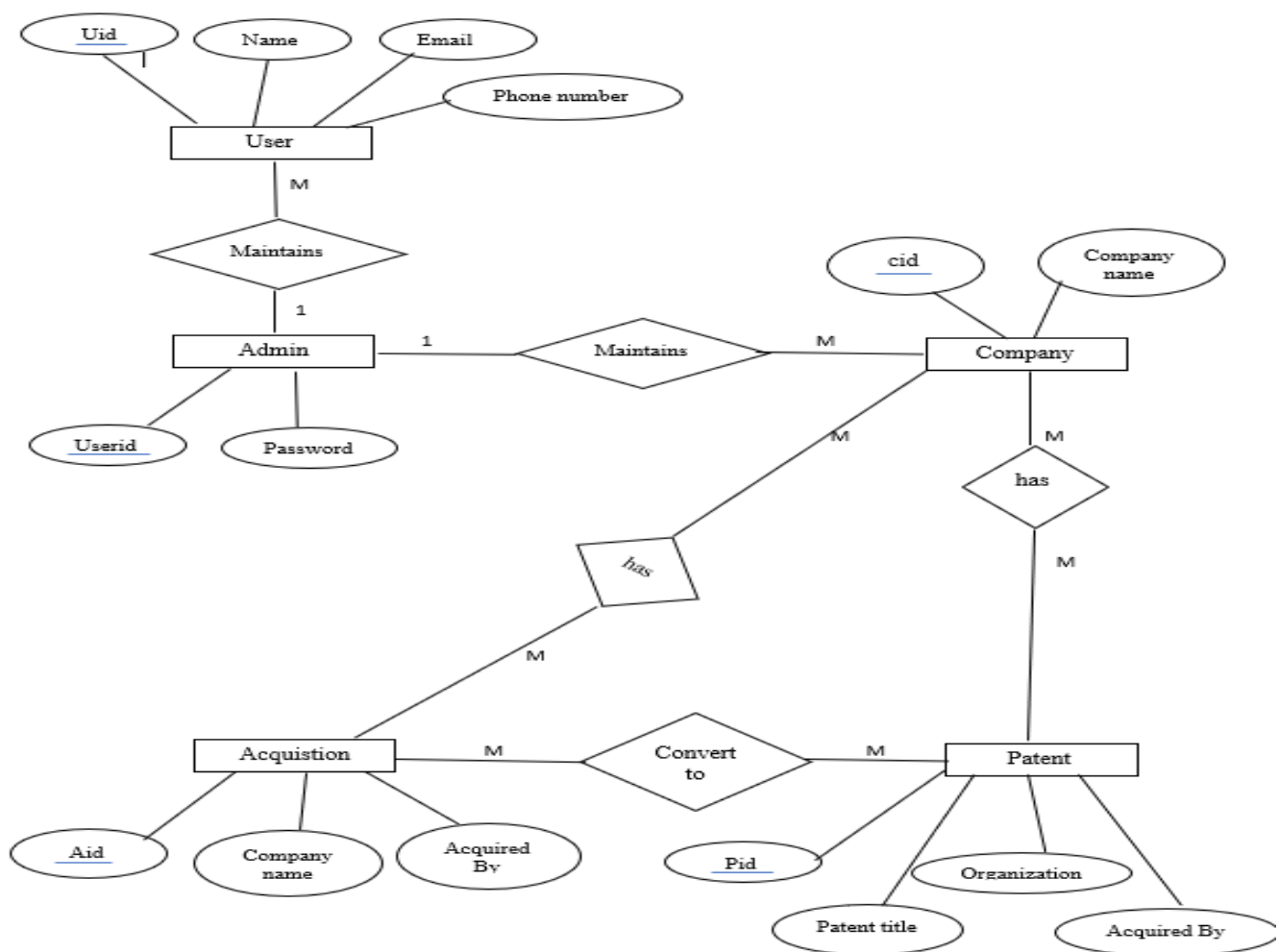
**Relationship:** A relationship describes how entities interact.

**Cardinality:** Cardinality specifies how many instances of an entity relate to one instance of another entity. Cardinality specifies the maximum number of relationships.

## Entities and their attributes

The following figure 3.3 shows ER Diagram

User	User id (Primary key), Name, Email, Phone.
Admin	User id (Primary key), Password.
Company	Company id (Primary key), Company name.
Patent	Patent id (Primary key), Patent, Organization, Acquired by.
Acquisition	Acquisition id (Primary key), Company name, Acquired by.



**Figure 3.3: ER Diagram**

### **3.4 UML Diagrams:**

The Unified Modelling Language (UML) is a Standard language for specifying, Visualizing, constructing, and documenting the software system and its components [6]. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. Structural models represent the framework for the system and this framework is the place where all other components exist. So, the class diagram, component diagram and deployment diagrams are the part of structural modelling.

They all represent the elements and the mechanism to assemble them. But the structural model never describes the dynamic behaviour of the system. Behavioural model describes the interaction in the system. It represents the interaction among the structural diagrams [7]. Behavioural modelling shows the dynamic nature of the system. Architectural model represents the overall framework of the system. It contains both structural and behavioural elements of the system. Architectural model can be defined as the blueprint of the entire system. Package diagram comes under architectural modelling.

The Unified Modelling Language encompasses a number of models

- Use Case Diagram
- Class Diagram
- Sequence Diagram
- Activity Diagram

#### **3.4.1 Use Case Diagram:**

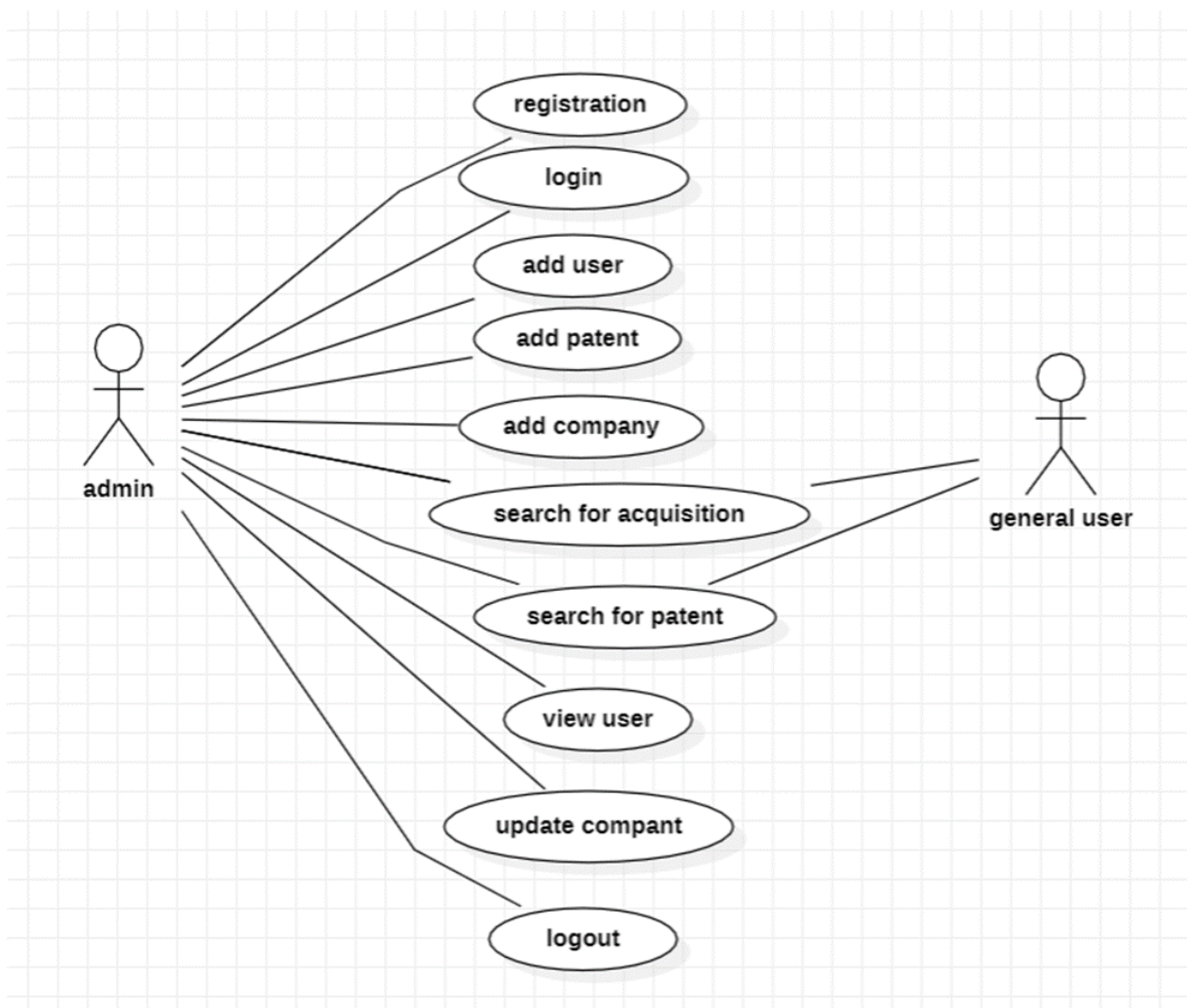
Use case diagrams are one of the five diagrams in the UML for modelling the dynamic Aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for modelling the dynamic aspects of systems). Use case diagrams are central to modelling the behaviour of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relations.

The key points are:

- The main purpose is to show the interaction between the use cases and the actor.

- To represent the system requirement from the user's perspective.
- Use cases are the functions that are to be performed in the module.

The following figure 3.4.1 shows the use case diagram for Admin. The admin uploads a metadata file and a raw file to the back-end system via POST request. The system takes the metadata and uses it to unpack the raw files which contain the patent data. The system then takes the patent data and saves it into tabular data, which are shown as patent and acquisitions listed to the user.



**Figure 3.4.1: Use Case Diagram**

### 3.4.2 Class Diagram

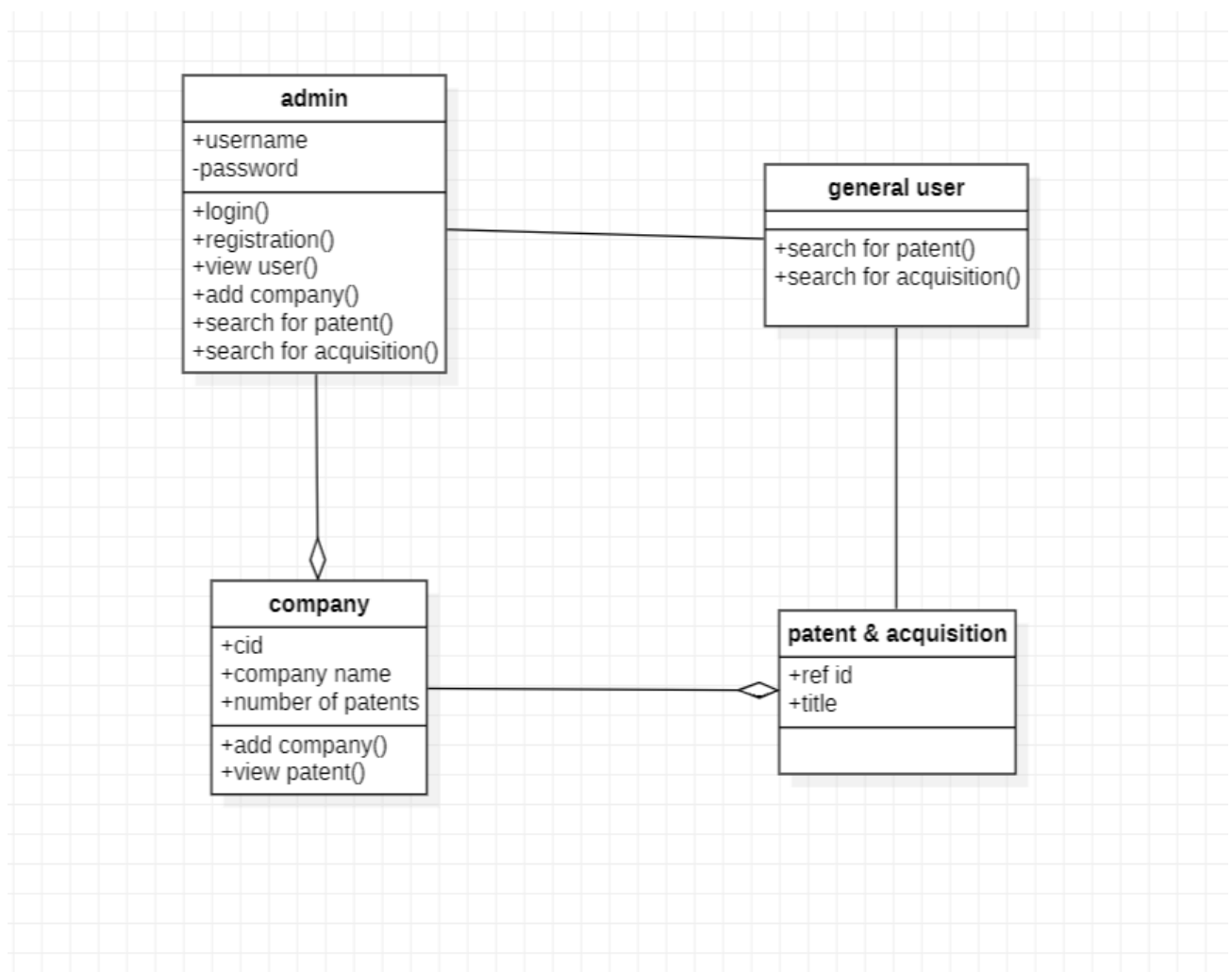
A “Class Diagram” shows a set of classes, interfaces and collaborations and their Relationships. These diagrams are the most common diagrams in modelling object-oriented systems. The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the Constraints imposed on the system.[6] The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams which can be mapped directly with object-oriented languages.

The below Figure 3.4.2 shows the methods that are used in this class diagram and the Description of each class is listed below.

**Admin:** This class is used for getting users and product information from the database and it is also used to update the database with the information about new-user registration, product checkout, and user details.

3.     **Company:** A company is a legal entity formed by a group of individuals to engage in and operate a business commercial or industrial enterprise.
4.     **Patent:** A patent is an exclusive right granted for an invention, which is a product or a process that provides, in general, new way of doing something, or offers a new technical solution to a problem.
5.     **Acquisition:** An act by one company to acquire control of another company, either through an agreed-upon deal or a takeover.
6.     **General user:** A user who has both registered an individual Profile in our Portal and established an association with an Organisation Profile.



**Figure 3.4.2: Class Diagram**

### 3.4.3 Sequence Diagram

Sequence diagram is an interaction diagram which focuses on the time ordering of messages. It shows a set of objects and messages exchanged between these objects. This diagram illustrates the dynamic view of a system.

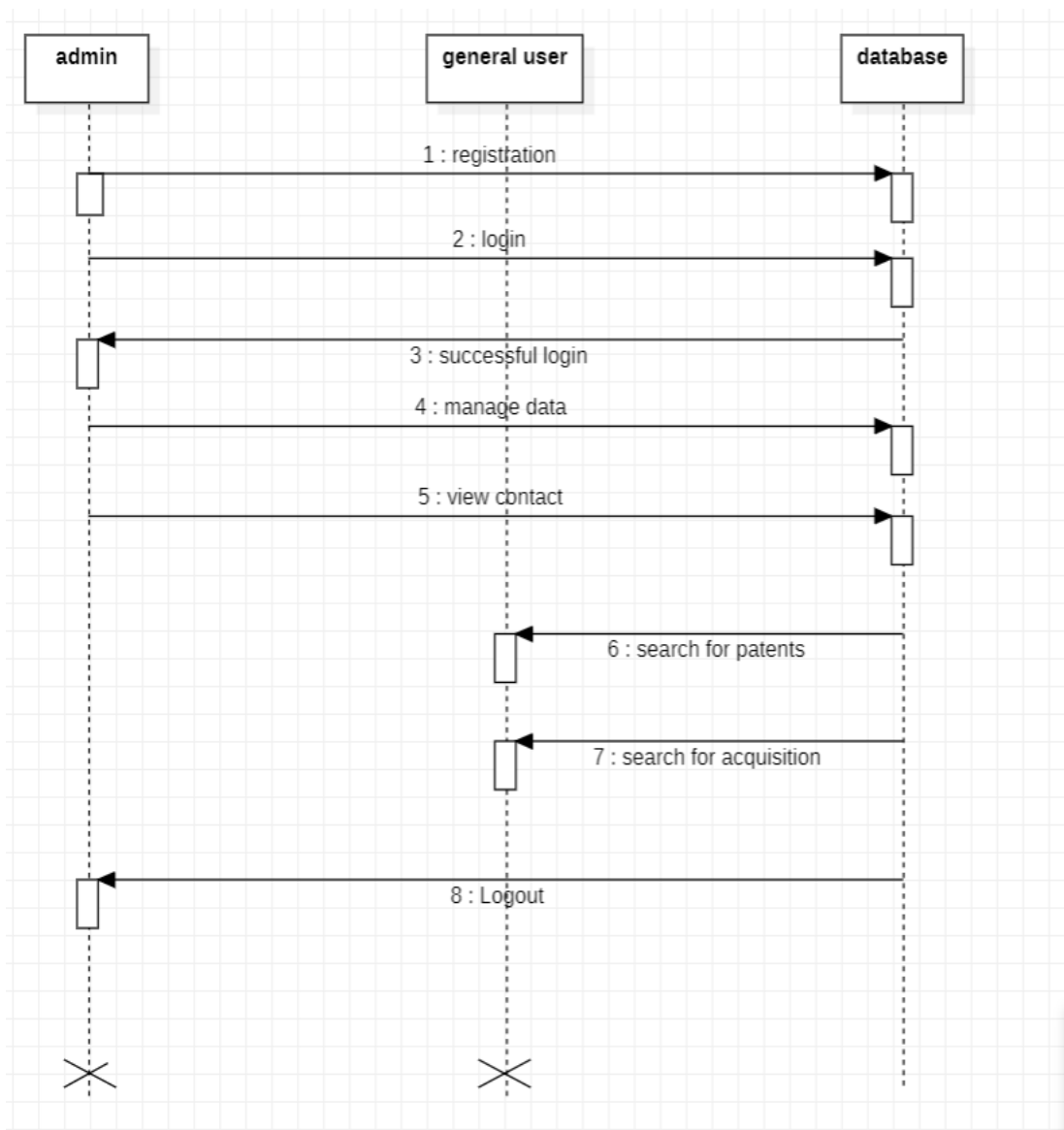
**The key points are:**

1. The main purpose is to represent the logical flow of data with respect to a process

2. A sequence diagram displays the objects and not the classes.

### Sequence Diagram

Below figure 3.4.3 shows the admin operations. The admin first enters login credentials then the back end takes actions. If the login credentials are matched with the database then the admin is successfully logged into the site. Here admin can perform operations like manage database and user view records.

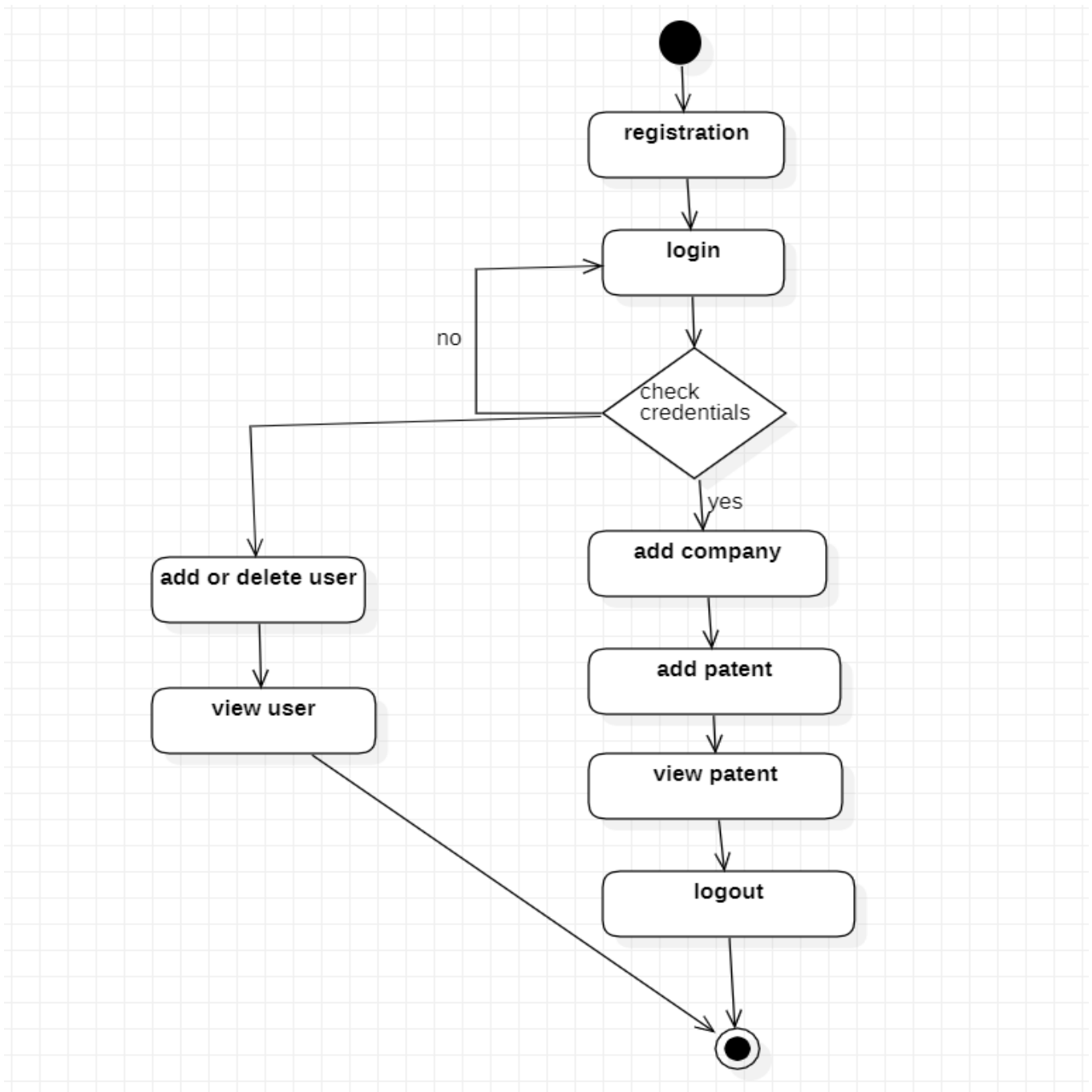


**Figure 3.4.3: Sequence Diagram**

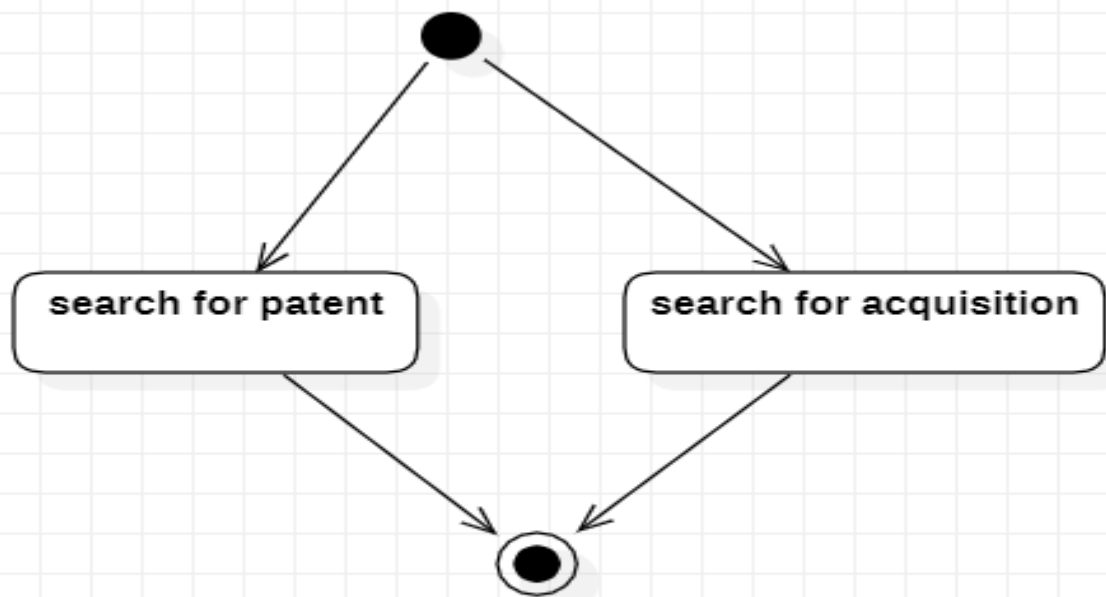


### 3.4.4 Activity Diagram:

Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. An activity is shown as a rounded box containing the name of operation. This activity diagram describes the behaviour of the system.



**Figure 3.4.4.1: Activity Diagram for Admin**



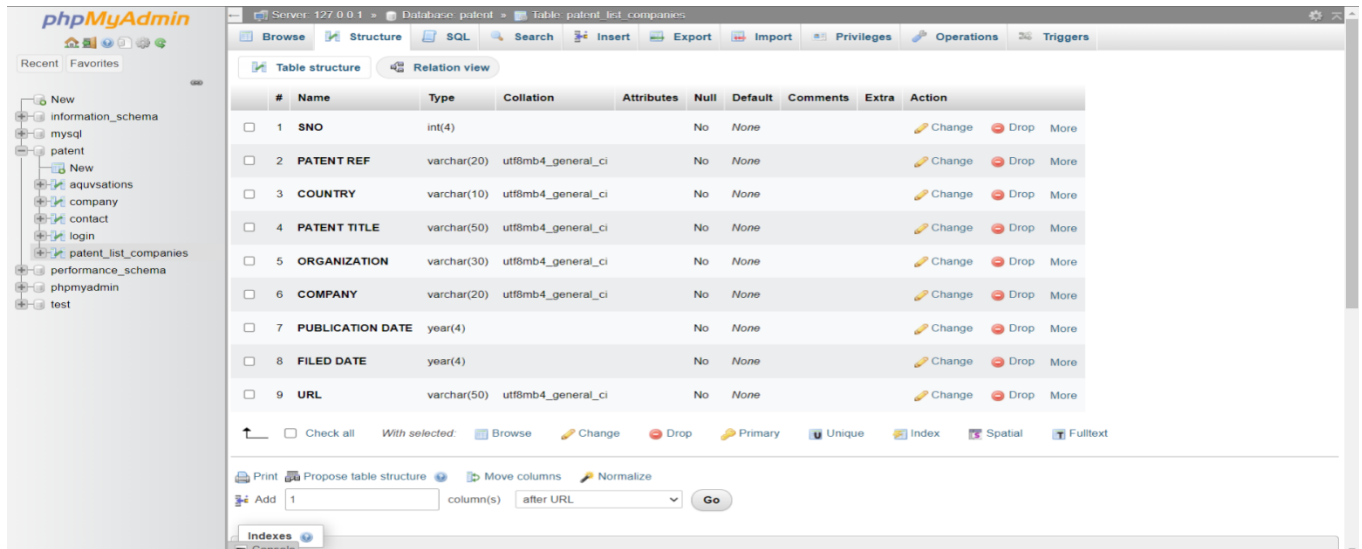
**Figure 3.4.4.2: Activity Diagram for User**

## 3.5 Data Dictionary:

### Patent Details:

This table 3.5.1 stores the information about the patent.

**Table 3.5.1: Patent Details**



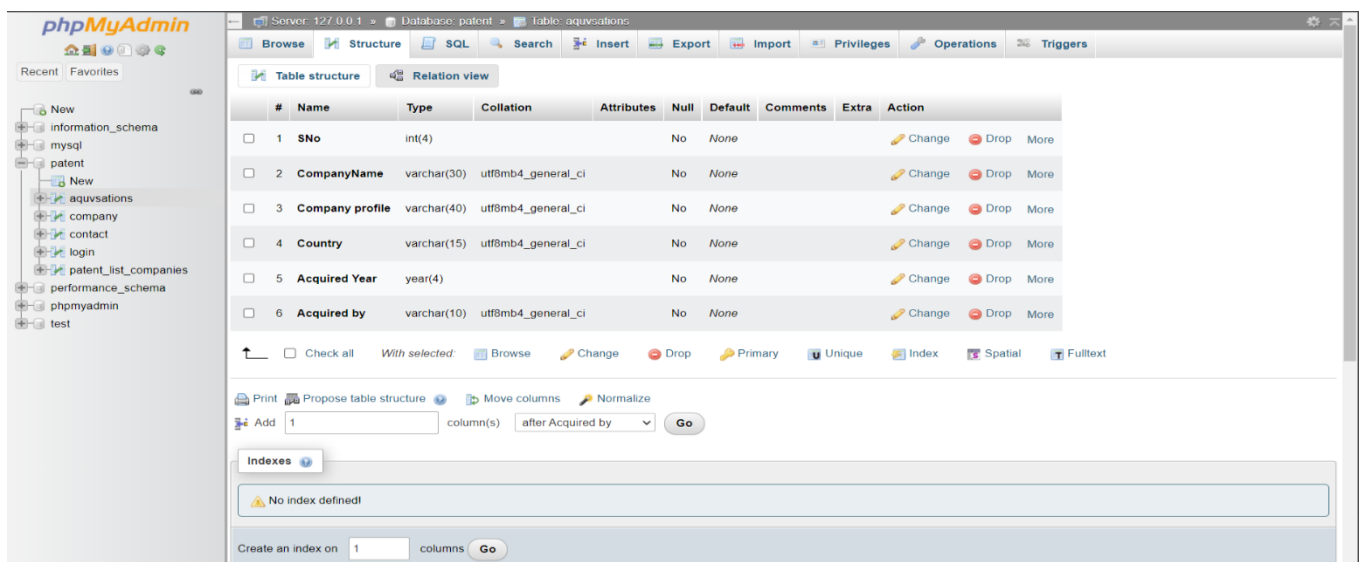
The screenshot shows the phpMyAdmin interface for the 'patent' database. The 'Table structure' tab is selected, displaying the following table structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	SNO	int(4)			No	None			Change Drop More
2	PATENT REF	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
3	COUNTRY	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More
4	PATENT TITLE	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
5	ORGANIZATION	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
6	COMPANY	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
7	PUBLICATION DATE	year(4)			No	None			Change Drop More
8	FILED DATE	year(4)			No	None			Change Drop More
9	URL	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

### Acquisitions Details:

This table 3.5.2 stores the information of customers like Name, Address, Contact number, and Email address.

**Table 3.5.2: Acquisitions Details**



The screenshot shows the phpMyAdmin interface for the 'acquisitions' database. The 'Table structure' tab is selected, displaying the following table structure:

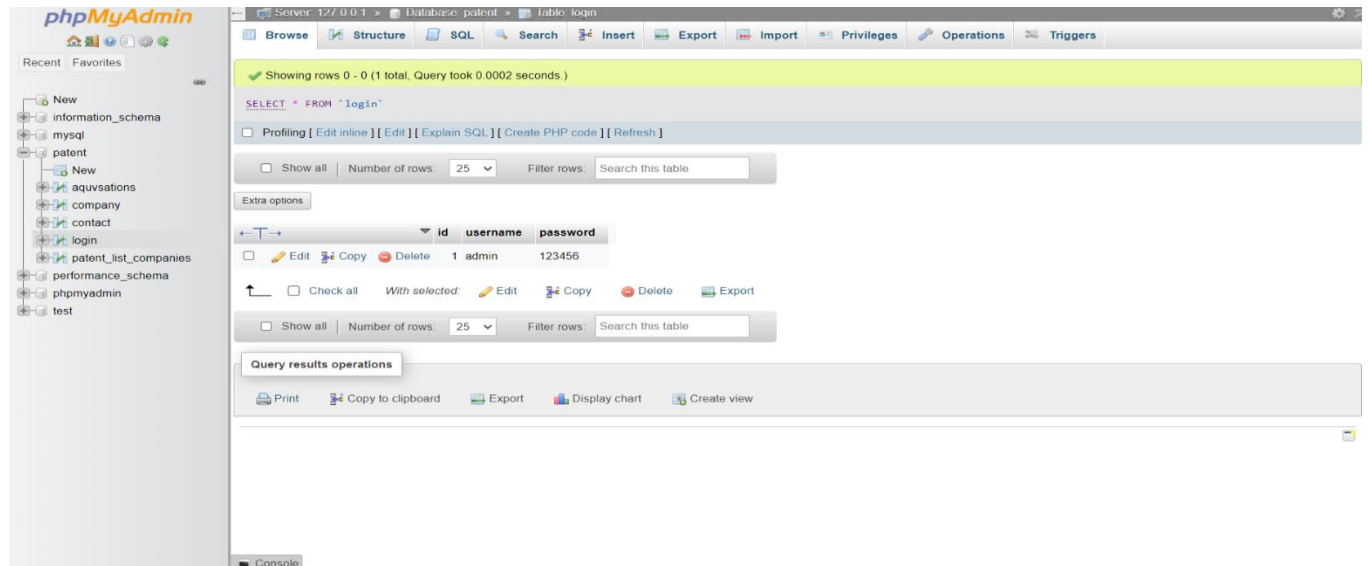
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	SNo	int(4)			No	None			Change Drop More
2	CompanyName	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
3	Company profile	varchar(40)	utf8mb4_general_ci		No	None			Change Drop More
4	Country	varchar(15)	utf8mb4_general_ci		No	None			Change Drop More
5	Acquired Year	year(4)			No	None			Change Drop More
6	Acquired by	varchar(10)	utf8mb4_general_ci		No	None			Change Drop More

Below the table structure, there is a section for indexes. It shows a warning: "No index defined!". At the bottom, there is a form to create an index on column(s) with a "Go" button.

## Login Details:

This table 3.5.3 stores the information about the login of each admin.

**Table 3.5.3: Login Details**



The screenshot shows the phpMyAdmin interface with the 'login' table selected. The table structure is displayed as follows:

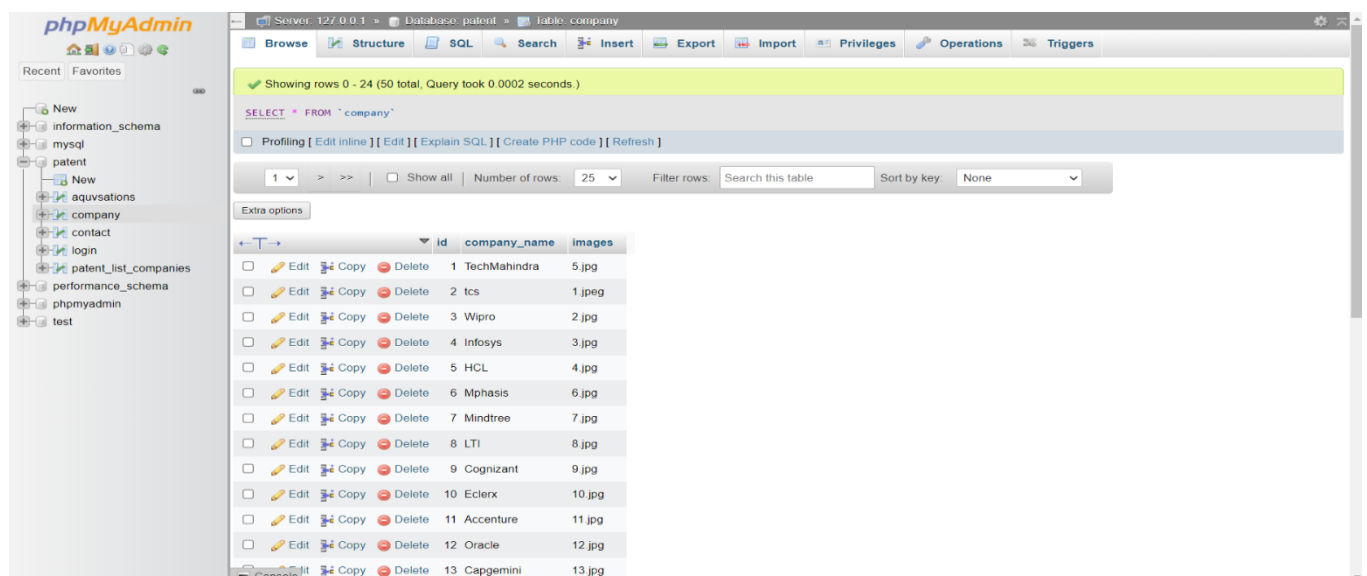
id	username	password
1	admin	123456

The interface also shows the query results operations menu with options like Print, Copy to clipboard, Export, Display chart, and Create view.

## Company Details

This table 3.5.4 stores the information about the companies.

**Table 3.5.4: company Details**



The screenshot shows the phpMyAdmin interface with the 'company' table selected. The table structure is displayed as follows:

id	company_name	images
1	TechMahindra	5.jpg
2	tcs	1.jpeg
3	Wipro	2.jpg
4	Infosys	3.jpg
5	HCL	4.jpg
6	Mphasis	6.jpg
7	Mindtree	7.jpg
8	LTI	8.jpg
9	Cognizant	9.jpg
10	Eclerx	10.jpg
11	Accenture	11.jpg
12	Oracle	12.jpg
13	Capgemini	13.jpg

The interface also shows the query results operations menu with options like Print, Copy to clipboard, Export, Display chart, and Create view.

## Contact Details

This table 3.5.5 stores the information about the contact details of user.

**Table 3.5.5: contact Details**

Server: 127.0.0.1 » Database: patent » Table: contact

Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

`SELECT * FROM `contact``

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	name	email	phonenumber	message
<input type="checkbox"/>	9	SRIYA KUNKUMALLA	sriyakunkumalla@gmail.com	2147483647	hello
<input type="checkbox"/>	10	rohith	gaderohith2002@gmail.com	2147483647	hi
<input type="checkbox"/>	11	pranay	pranayvollala@gmail.com	9346304128	hi

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

Console

# CHAPTER 4

## IMPLEMENTATION

### 4.1 Languages Used

#### 4.1.1 PHP:

PHP is one of the Web's most popular programming languages. According to Net craft, PHP was running on more than 20 million Web servers in July 2007 at the time of writing, it's the fourth most popular programming language in the world according to TIOBE, beaten only by Java, C, and C++. With the introduction of version 5.3, there's never been a better time to learn PHP. PHP is a programming language for building dynamic, interactive Web sites. As a general rule, PHP programs run on a Web server and serve Web pages to visitors on request. One of the key features of PHP is that you can embed PHP code within HTML Web pages, making it very easy for you to create dynamic content quickly. It is simple for a professional programmer to learn & they can use it effectively. If we already know structure-oriented programming, then learning PHP is very easy.

#### Features of PHP:

- Simple
- Secure
- Portable

#### Simple:

Executable file or program rather than, virus programs we have malicious programs that can gather private information, such as credit card number, bank account balances & passwords by searching the contents of your computers local file system.

#### Secure:

As we know many people are affected by a viral infection when they download what exactly does the phrase "dynamic, interactive Websites" mean? A dynamic Web page is a page whose contents can automatically change each time the page is viewed. Contrast this with a static Web page, such as 32 a simple HTML file, which looks the same each time it is displayed (at least until the page is next edited). Meanwhile, an interactive Web site is a site that responds to input from its visitors. A Web

forum is a good example users can post new messages to the forum, which are then displayed on the site for all to see. Another simple example is a "contact us" form, where visitors interact with the page by filling out and sending a form, which is then emailed to the Webmaster.

### **Portable:**

As already we have discussed the compatibility of the operating system, computers, and chips. On Internet the programs have to be dynamically downloaded to all the various types of platforms like windows for Wamp, Linux for lamp and support all platforms using XAMPP.

### **4.1.2 Database Management System (DBMS):**

A Database is an integrated collection of user-related data stored with minimum redundancy, serves many users/applications quickly and efficiently. A database system is a computerized recordkeeping system, i.e., it is a computerized system whose overall purpose is to maintain information and make that information available on demand.

Database Management System is divided into 4 main components

- Database.
- Hardware.
- Software.
- User.

**Database:** It consists of the collection of persistent data that is used by the application system.

**Hardware:** The processor(s) and associated main memory that is used to support the execution of database system software.

**Software:** The layer between the physical database and the users that handles all requests from the user for access to the database.

**User:** There are three types of users

- Application Programmers
- End-User<sup>33</sup>
- Database Administrator (DBA)

## **Types of DBMS:**

There are four major categories of DBMS data models.

- Hierarchical
- Network
- Inverted
- Relational

## **Relational Database Management Systems:**

Database Management System has evolved from hierarchical to network to relational models. Today, the most widely accepted database model is the relational model. The relational database management system uses only its relational capabilities to manage the information stored in the database.

The relational model has three different aspects.

- Structures.
- Operation.
- Integrity rules.

### **Structures:**

They are well-defined objects that store the data of a database structure and the data Contained within them can be manipulated by operations.

### **Operations:**

They have clearly defined actions that allow users to manipulate the data and structures of a database. The operations on a database must adhere to a predefined set of integrity rules.

## **Server Programming Language:**

PHP is always there. I've never found a hosting company that didn't offer it. Java is sometimes an extra-cost option if it's available at all, and Python and Ruby are often unavailable. PERL is as 34



common as PHP, but it's an even worse language. It's fast. It's so widely used that there's lots of optimization for it, especially when used with Apache.

So, the answer to why PHP, since I can and have used almost every language that ever existed, is that it's pleasant enough to use, always available, extremely well supported, and nearly always has a function to There are three other languages you'll be using, as web application developers always use at least four languages.

The others are:

- HTML
- CSS
- JavaScript
- MYSQL
- Bootstrap

HTML and JavaScript run in the browser; never on the server. SQL is passed to the Database from your PHP program, or sometimes used directly on the database, so it's a server language. Design is the first step in moving from the problem domain to the solution domain. Design is essentially the bridge between requirements specification and the final solution.

The goal of the design process is to produce a model or representation of a system, which Can be used later to build that system. The produced model is called the "Design of the System". It is a plan for a solution for the system.

#### **4.1.3 HTML:**

HTML is a hypertext mark-up language that is in reality a backbone of any website. Every website can't be structured without the knowledge of HTML. If we make our web page only with the help of HTML, then we can't add many of the effective features in a web page, for making a web page more effective we use various platforms such as CSS. So here we are using this language to make our web pages more effective as well as efficient. And to make our web pages dynamic we are using JavaScript.

#### **4.1.4 CSS:**

CSS Stands for "Cascading Style Sheet." Cascading style sheets are used to format the

layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that 35 previously could only be defined in a page's HTML. The basic purpose of CSS is to separate the content of a web document (written in any mark-up language) from its presentation (that is written using Cascading Style Sheets CSS gives the option of selecting various style schemes and rules according to the requirements and it also allows the same HTML document to be presented in more than one varying style.

#### **4.1.5 JavaScript:**

JavaScript is considered to be one of the most famous scripting languages of all time. JavaScript, by definition, is a Scripting Language of the World Wide Web. The main usage of JavaScript is to add various Web functionalities, Web form validations, browser detections, creation of cookies, and so on. JavaScript is one of the most popular scripting languages and that is why it is supported by almost all web browsers available today like Firefox.

We used the browser Opera or Internet Explorer. JavaScript is considered to be one of the most powerful scripting languages in use today. It is often used for the development of client-side web development. JavaScript is used to make web pages more interactive and dynamic. JavaScript is a lightweight programming language, and it is embedded directly into the HTML code. JavaScript, as the name suggests, was influenced by many languages, especially Java.

The advantages of JavaScript: -

- An interpreted language.
- Embedded within HTML.
- The minimal syntax is easy to learn.
- Performance sign for simple, small programs.

#### **4.1.6 MYSQL:**

MYSQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MYSQL is developed, marketed, and supported by MYSQL, which is a Swedish company. MYSQL is becoming so popular because of many good reasons.

- MYSQL is released under an open-source license. So, you have nothing to pay to use it. MYSQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.

- MYSQL uses a standard form of the well-known SQL data language.
- MYSQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MYSQL works very quickly and works well even with large data sets.
- MYSQL is very friendly to PHP, the most appreciated language for web development.
- MYSQL is customizable. The open-source GPL license allows programmers to modify the MYSQL software to fit their specific environments.

#### **4.1.7 Bootstrap:**

Bootstrap is a free open-source development framework for the creation of websites and web apps. The Bootstrap framework is built on HTML, CSS, and JavaScript (JS) to facilitate the development of responsive, mobile-first sites and apps. Responsive design makes it possible for a web page or app to detect the visitor's screen size and orientation and automatically adapt the display; accordingly, the mobile-first approach assumes that smart phones, tablets, and task-specific mobile apps are employees' primary tools for getting work done and addresses the requirements of those technologies in design.

Bootstrap includes user interface components, layouts, and JS tools along with the framework for implementation. The Software is available precompiled or as source code. Mark Otto and Jacob Thornton developed Bootstrap at Twitter as a means of improving the consistency of tools used on the site and reducing maintenance. The software was formerly known as Twitter Blueprint and is sometimes referred to as Twitter Bootstrap.

In computers, the word bootstrap means to boot: to load a program into a computer using a much smaller initial program to load in the desired program (which is usually an operating system). In the physical world, bootstrap is a small strap or loop at the back of a leather boot that enables you to pull the entire boot on, and in general, usage, bootstrapping is the leveraging of a small initial effort into something larger and more significant. There is also a common expression, "pulling yourself up by your bootstraps," meaning to leverage yourself to success from a small beginning.

## 4.2 Frontend Design:

### Homepage:

The homepage of the application (figure 4.2.1) is common to all the system users. This interface is available through the web application at the time of login. The feature of the homepage is to allow the admin to login in their computers. This page shows the navigating categories. There are five categories:

1. Home
2. About us
3. Companies
4. Areas
5. Contact us

Each category links to an individual page containing the items related to the category to which it is assigned.

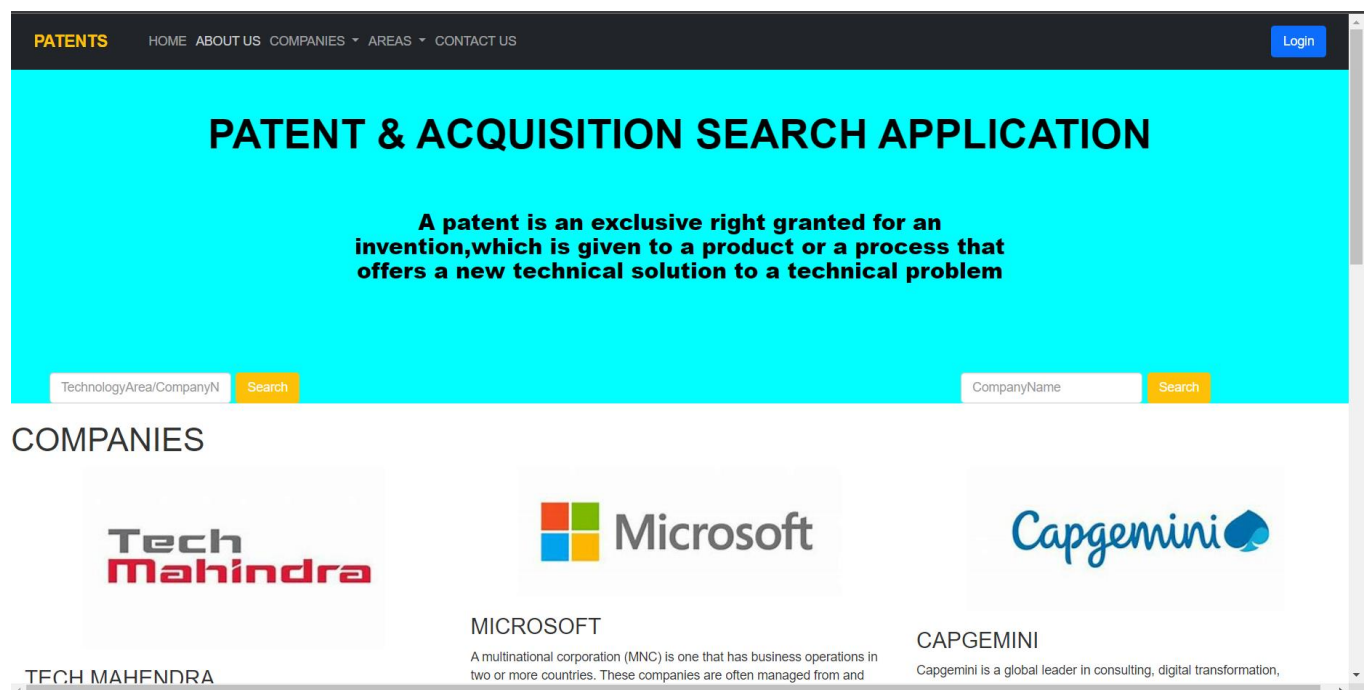
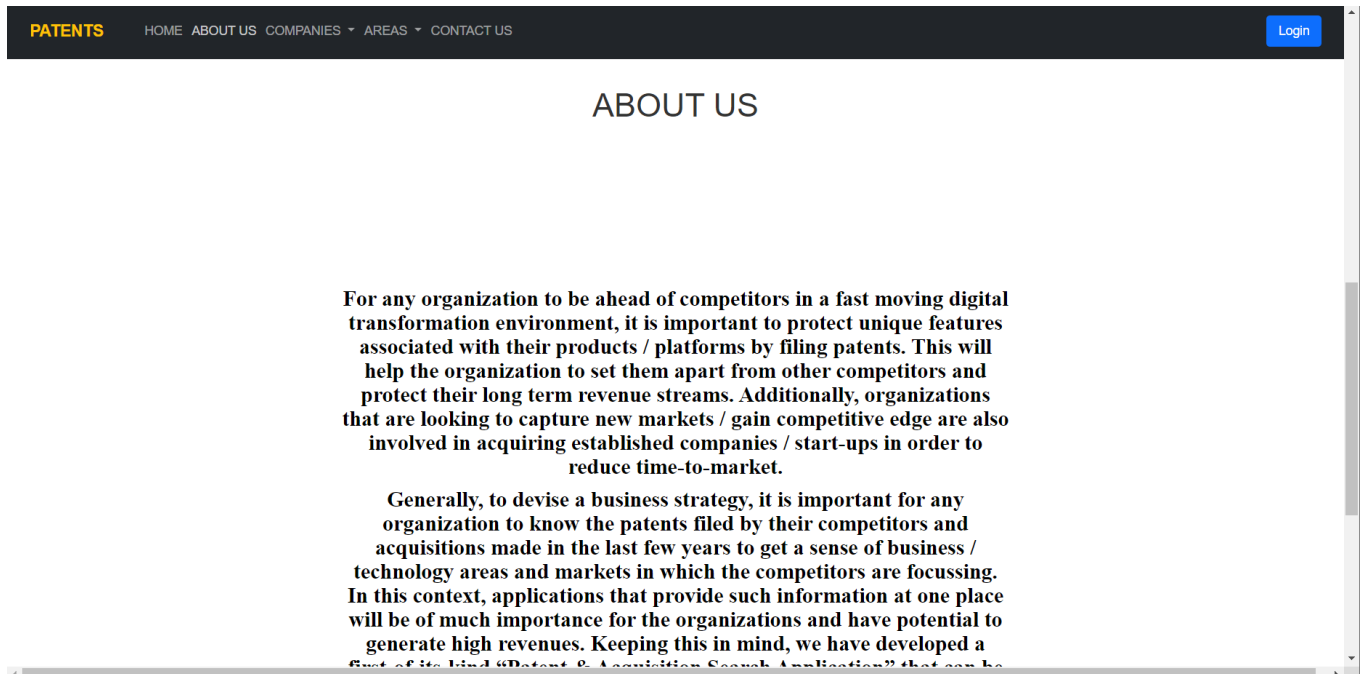


Figure 4.2.1: Screenshot for Home page

## About us:

Below figure 4.2.2 represents about the Patent & Acquisition Search Application . This page describes the establishment of the company.



**Figure 4.2.2: Screenshot for about page**

## Services:

Below figure 4.2.3 & 4.2.4 represents the services provided by the Patent & Acquisition Search Application.

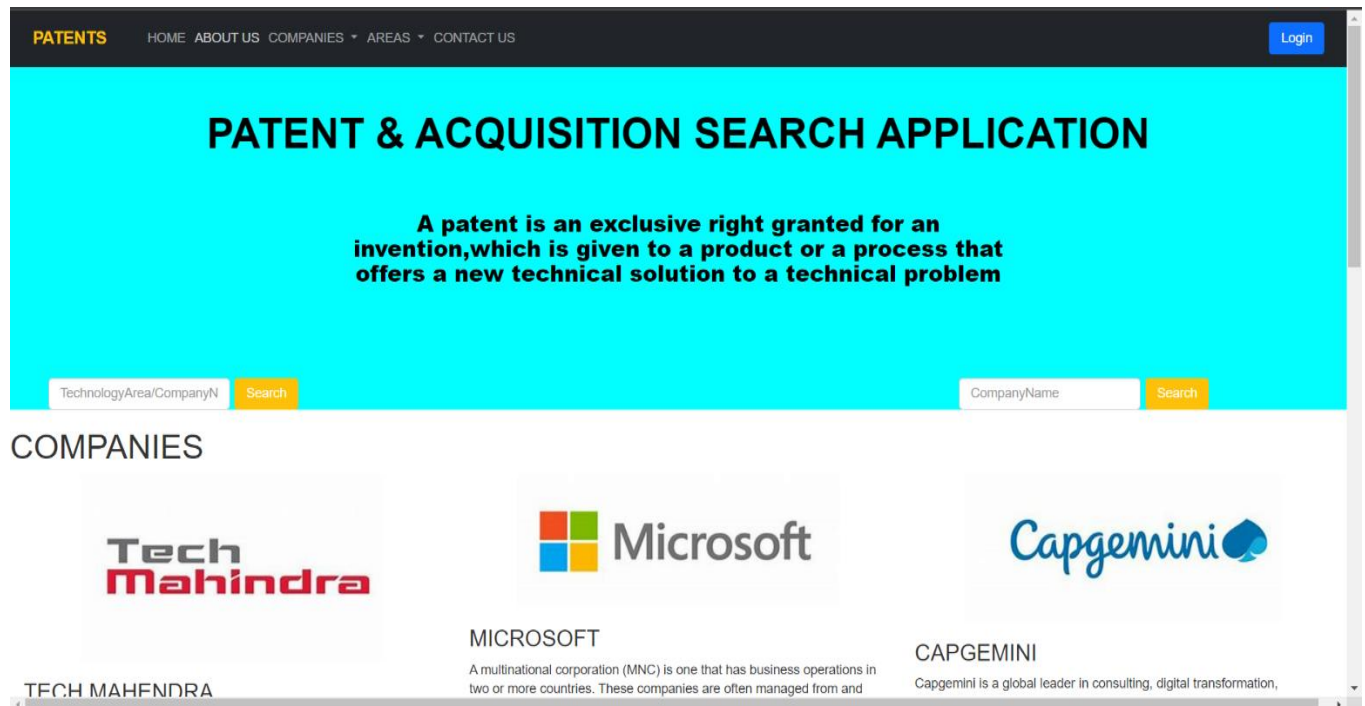


Figure 4.2.3: Screenshot for Services provided by patents

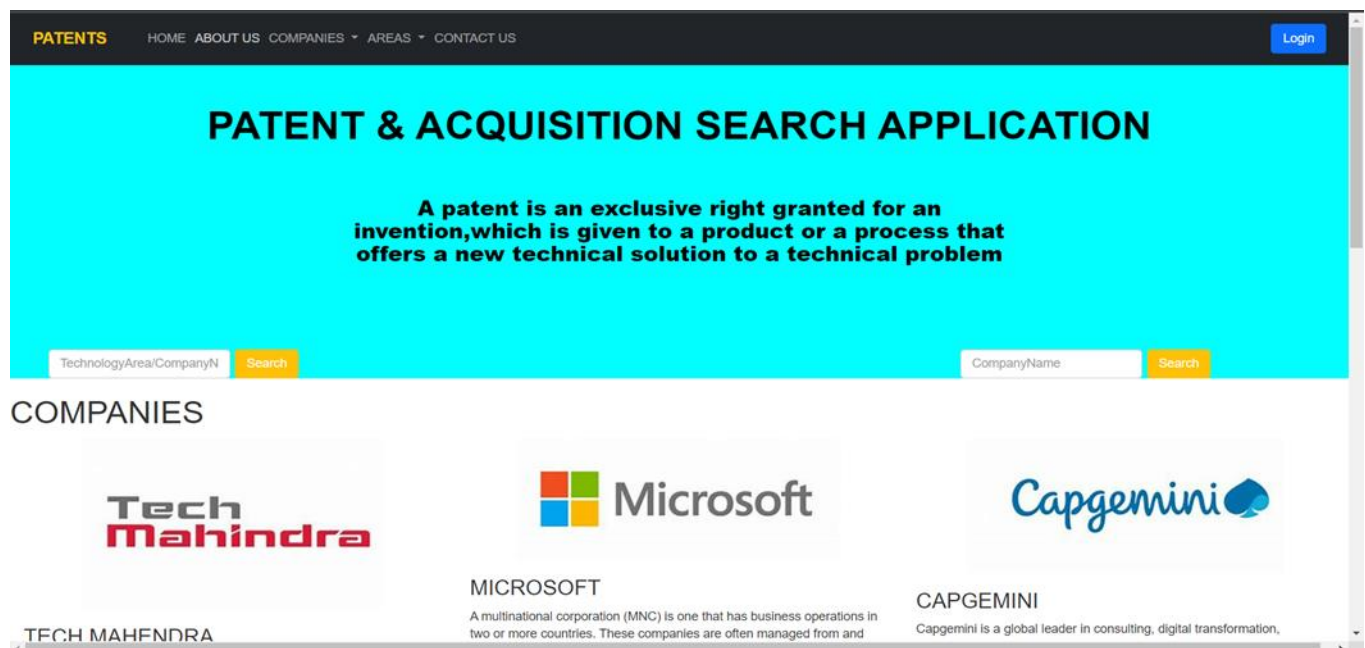


Figure 4.2.4: Screenshot for Services provided by acquisition

## Companies:

Below figure 4.2.5 represents all the company names with their filed patents year wise which are done by the Patent & Acquisition Search Application.

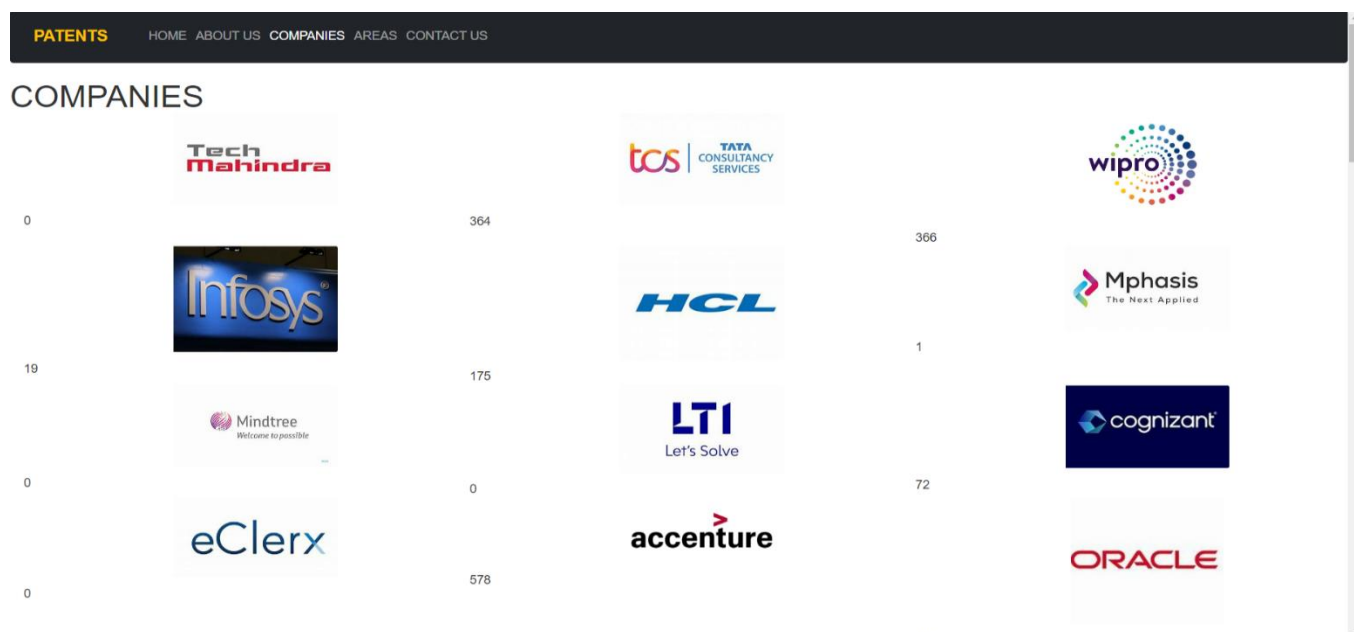


Figure 4.2.5: Screenshot for Companies

## Contact:

Below figure 4.2.6 represents the process of contacting the patent organizers by the customers who needs to contact them.

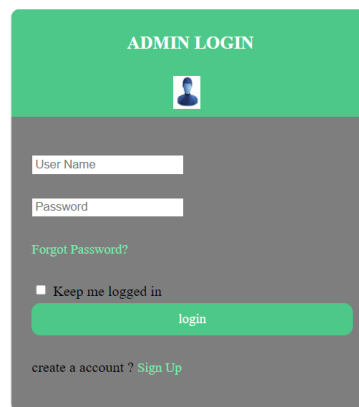
The screenshot shows a contact form titled 'CONTACT'. The form includes a red header bar with the title 'CONTACT'. Below the header, there is a red bar with an envelope icon. The form contains four input fields: NAME, EMAIL, MOBILE, and a large text area for a message. A red 'submit' button is located at the bottom of the form.

Figure 4.2.6: Screenshot for Contact

### **Admin login:**

This figure 4.2.7 represents the admin login page. Here the admin enters his/her login credentials.

On clicking the login button admin will enter into the admin portal.



The screenshot shows a web form titled "ADMIN LOGIN" in a green header bar. Below the header is a small profile icon. The form contains two input fields: "User Name" and "Password". Below these fields is a link "Forgot Password?". There is a checkbox labeled "Keep me logged in". A green "login" button is positioned below the checkbox. At the bottom of the form, there is a link "create a account ? Sign Up".

**Figure 4.2.7: Screenshot for Admin login page**



# CHAPTER 5

## TESTING

It is the process of testing the functionality and it is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers an undiscovered error. Software testing is usually performed for one of two reasons:

- Defect Detection
- Reliability estimation

### 5.1 Testing

#### Admin Login Page:

1. Figure 5.1.1 shows attempting login by leaving both username and password fields empty.

The expected output is to show “please fill out this field”.

2. Figure 5.1.2 shows trying to login with email and leaving the password field empty.

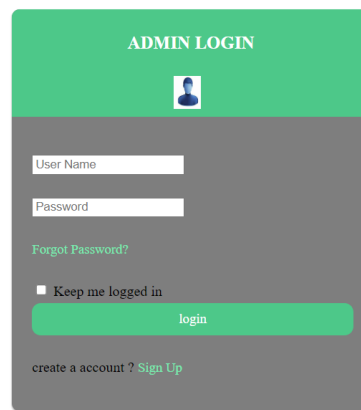
The expected output is to show “please enter password”.

3. Figure 5.1.3 shows trying to login with email and password.


The expected output is “successful login”, if email and password are valid.

4. Figure 5.1.4 shows trying to login with an email and wrong password.

The expected output is to show “invalid details, please try again”.



ADMIN LOGIN



User Name

Password

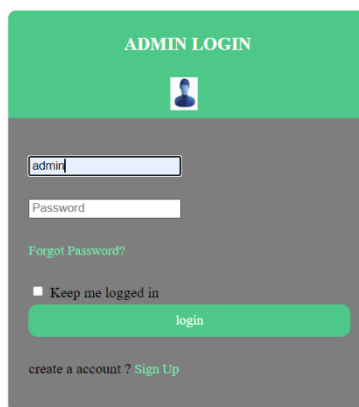
[Forgot Password?](#)

☐ Keep me logged in


[login](#)

create a account ? [Sign Up](#)

**Figure 5.1.1 both fields are empty**



ADMIN LOGIN



Password

[Forgot Password?](#)


☐ Keep me logged in


[login](#)

create a account ? [Sign Up](#)

**Figure 5.1.2 Password field is empty**

ADMIN LOGIN





Forgot Password?

☐ Keep me logged in

login

create a account ? [Sign Up](#)

Create Company

Company Name

Upload Image

Choose File

No file chosen

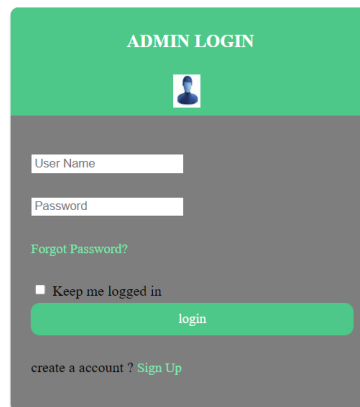
Save

Viewcontact


logout

**Figure 5.1.3 Successful login**

Invalid login Try Again



ADMIN LOGIN



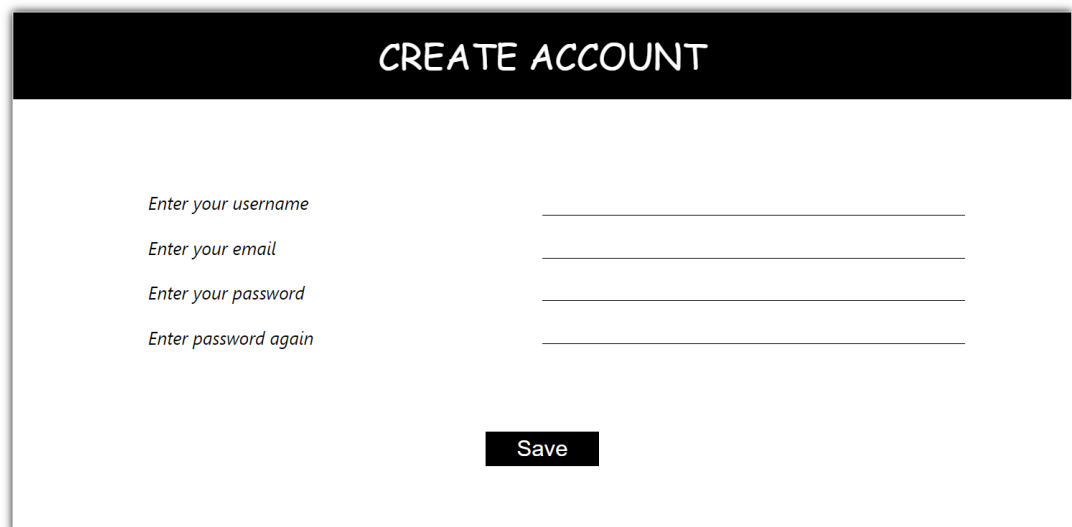
User Name

Password

[Forgot Password?](#)

☐ Keep me logged in

create a account ? [Sign Up](#)



CREATE ACCOUNT

*Enter your username*

*Enter your email*

*Enter your password*

*Enter password again*

**Figure 5.1.4 Invalid Credentials**

**Table 5.1.1 Admin Login Test**

<b>Test No.</b>	<b>Data input</b>	<b>Expected Output</b>	<b>Actual Output</b>	<b>Pass / Fail</b>
1	All empty fields	Please fill out this field	please fill out this field	Pass
2	USERNAME and leaving PASSWORD empty	Please enter password	please fill out this field	Pass
3	Correct USERNAME and PASSWORD	Successful login	Successful Login	Pass
4	Incorrect PASSWORD	Invalid details try again	Invalid details try again	Pass

## **5.2 Test Strategy**

Field-testing is performed manually, and functional testing is written in detail.

### **Test Objectives**

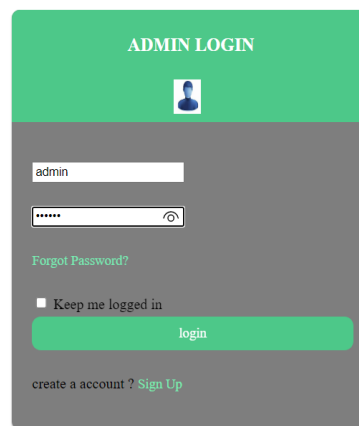
1. All field entries must work properly.
2. Pages must be activated from the identified link.
3. The entry screen, messages and responses must not be delayed.

### **Features to be tested**

1. Verify that entries are in correct format.
2. No duplicate entries should be allowed.
3. All links should take the user to the correct page.

### 5.3 Black Box Testing:

The base of the black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behaviour of the system. Now a days, it is becoming to route the testing work to a third party as the developer of the system knows too much of the internal logic and coding of the system which makes it unfit to rest application by the developer.



A screenshot of an 'ADMIN LOGIN' form. The form has a green header with the title 'ADMIN LOGIN' and a small user icon. Below the header, there is a text input field containing 'admin', a password input field with masked characters '\*\*\*\*\*' and a toggle icon, a green link for 'Forgot Password?', a checkbox for 'Keep me logged in', a green 'login' button, and a footer with the text 'create a account ? Sign Up'.

The admin has logged in successfully.

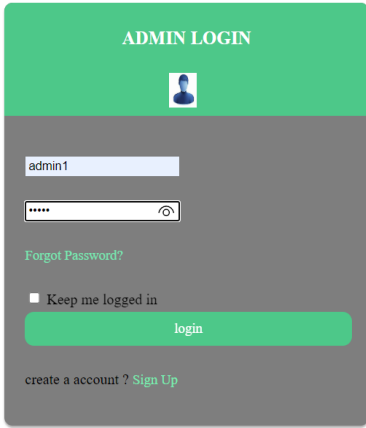
## 5.4 WHITE BOX TESTING:

White box testing requires access to source code. Though white box testing can be performed any time in the life cycle after the code is developed, it is a good practice to perform white box testing during unit testing phase.

In designing of database, the flow of specific inputs through the code, expected output and the functionality of conditional loops are tested.

1. Guarantee that all independent paths have been executed.
2. Executed all logical decisions on their true and false sides.
3. Executed all loops at their boundaries and within their operational bounds.
4. Executed internal data structures to ensure their validity.

Invalid login Try Again



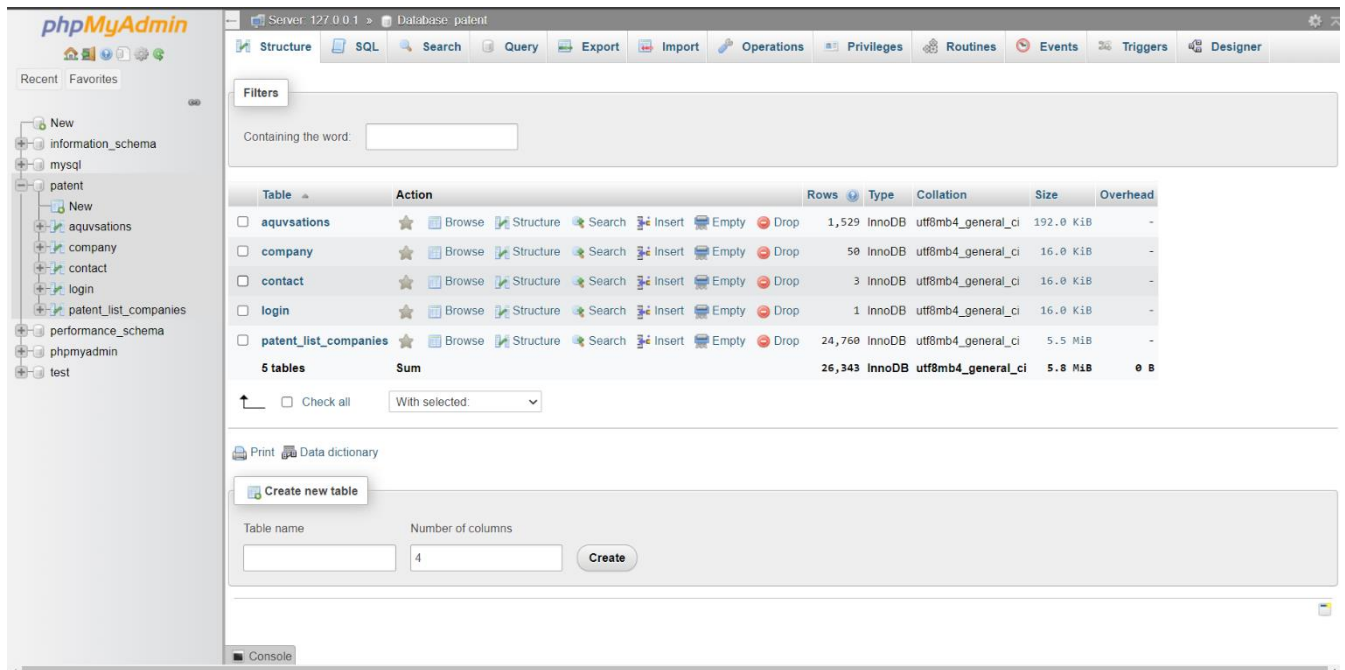
The image shows a web form titled "ADMIN LOGIN" with a green header. Below the header is a user icon. The form has two input fields: a username field containing "admin1" and a password field with masked characters "\*\*\*\*\*" and a toggle icon. Below the password field is a link "Forgot Password?". There is a checkbox labeled "Keep me logged in" which is unchecked. A green "login" button is positioned below the checkbox. At the bottom of the form, there is a link "create a account ? Sign Up".

The login has failed because inappropriate values are entered.

## 5.5 Database Testing:

This test involves testing whether the values entered through the form gets stored and saved in the database correctly or not.

Figure 5.3.1 shows that entered details are stored correctly in the database or not.

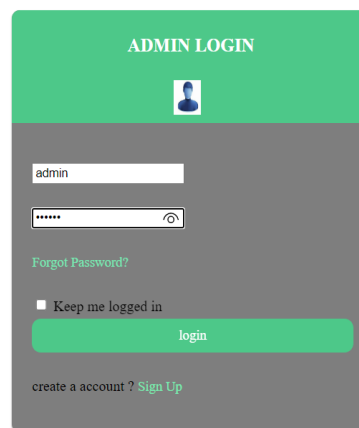


**Figure 5.3.1 DATABASE-patent**



## 5.6 Acceptance Testing:

Testing to verify a product meets customer specified requirements. The acceptance test suite is run against supplied input data. Then the results obtained are compared with the expected results of the client. A correct match was obtained.

A screenshot of an 'ADMIN LOGIN' form. The form has a green header with the title 'ADMIN LOGIN' and a small user icon. Below the header, there are two input fields: the first contains the text 'admin', and the second contains six dots for a password, with an eye icon to its right. Below the password field is a link that says 'Forgot Password?'. Underneath that is a checkbox labeled 'Keep me logged in'. A green 'login' button is positioned below the checkbox. At the bottom of the form, there is a link that says 'create a account ? Sign Up'.

### Test cases:

**Test case1:** Admin Login (successful)

**Test case:** Admin logged (successful).

**Test Description:** Provide Admin right by checking the details.

**Pre-Condition:** Database connectivity.

**Action Performed:** Entered valid login details without leaving any field.

**Expected Results:** Successful login.

**Condition Verified:** Yes.

**Result:** Success.

## 5.7 TEST RESULTS

Test cases are done on every page on this website. All the test cases are passed successfully. No defects were encountered

# CHAPTER 6

## RESULTS

The result screenshots are as follows:

### 6.1 Screenshot for Homepage

Below figure 6.1 shows the homepage of the website, the feature of the homepage is to allow the customer and Admin to navigate through the site. This page shows the navigating categories. There are five categories:

1. Home
2. About us
3. Companies
4. Areas
5. Contact us
6. Admin Login

Each category links to an individual page containing the items related to the category to which it is assigned.

Path: localhost/ patent

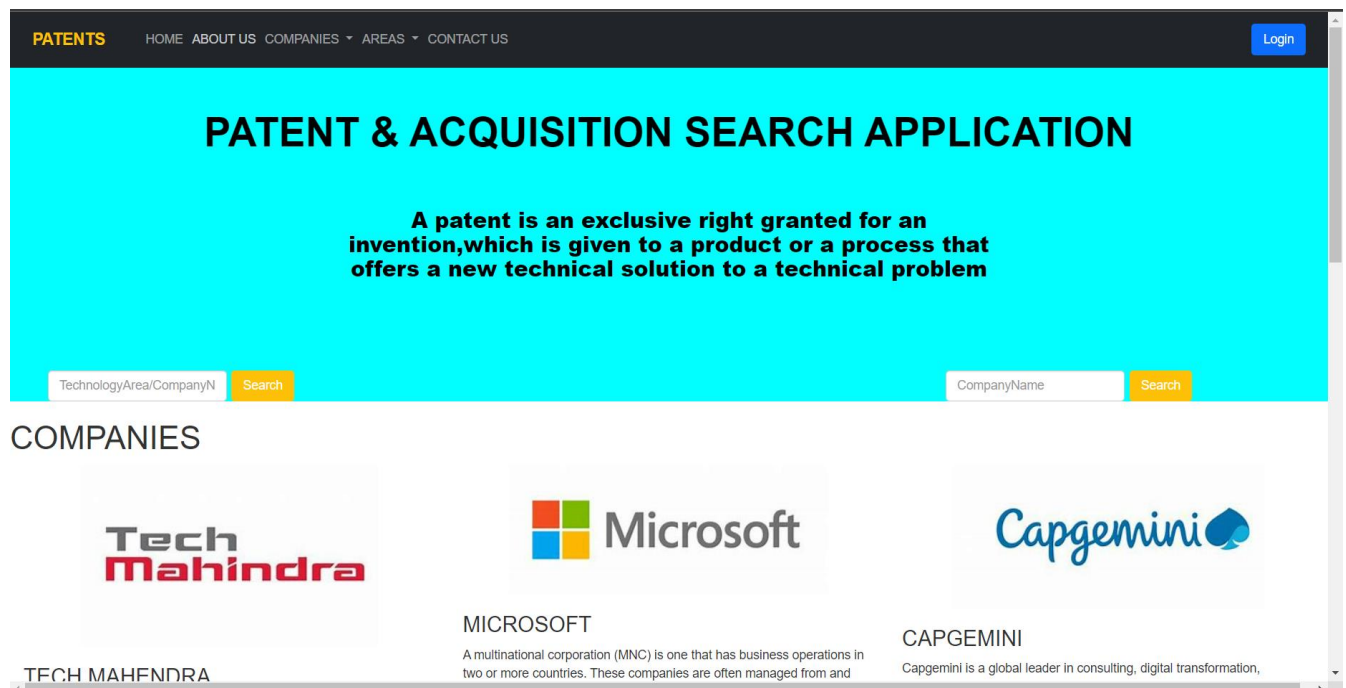
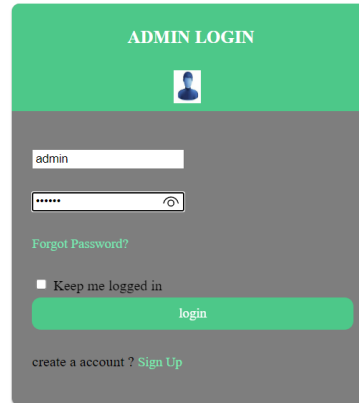


Figure 6.1: Screenshot of Homepage

## 6.2 Screenshot for login successfully

The Figure 6.2 show that admin is successfully logged in and can view required details about the sales of the products and other details.

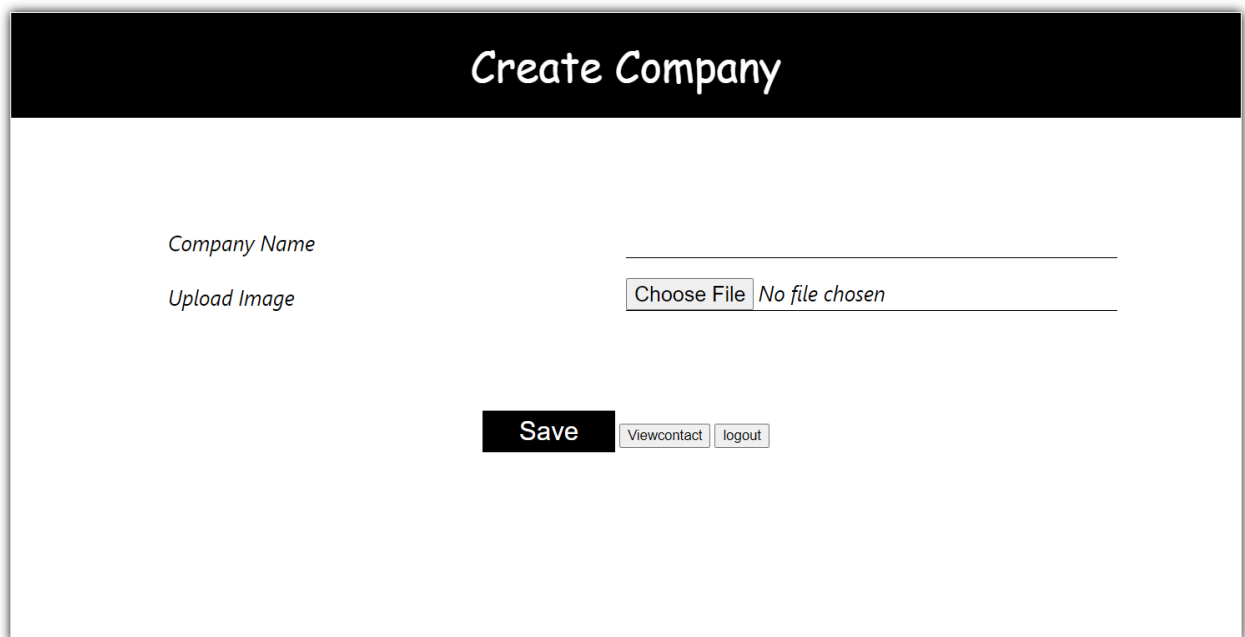


ADMIN LOGIN

[Forgot Password?](#)

☐ Keep me logged in

[create a account ? Sign Up](#)



Create Company

Company Name

Upload Image  No file chosen

**Figure 6.2: Screenshot for login successfully**

### 6.3 Screenshot for Add Company

The figure 6.3 represents that admin can add companies by selecting category, company with details.



The screenshot shows a web form titled "Create Company" in a black header bar. Below the header, the form has a white background. It contains two input fields: "Company Name" and "Upload Image". The "Company Name" field is a simple text input. The "Upload Image" field is a file upload control with a "Choose File" button and the text "No file chosen". At the bottom of the form, there are three buttons: a black "Save" button, a "Viewcontact" button, and a "logout" button.

**Create Company**

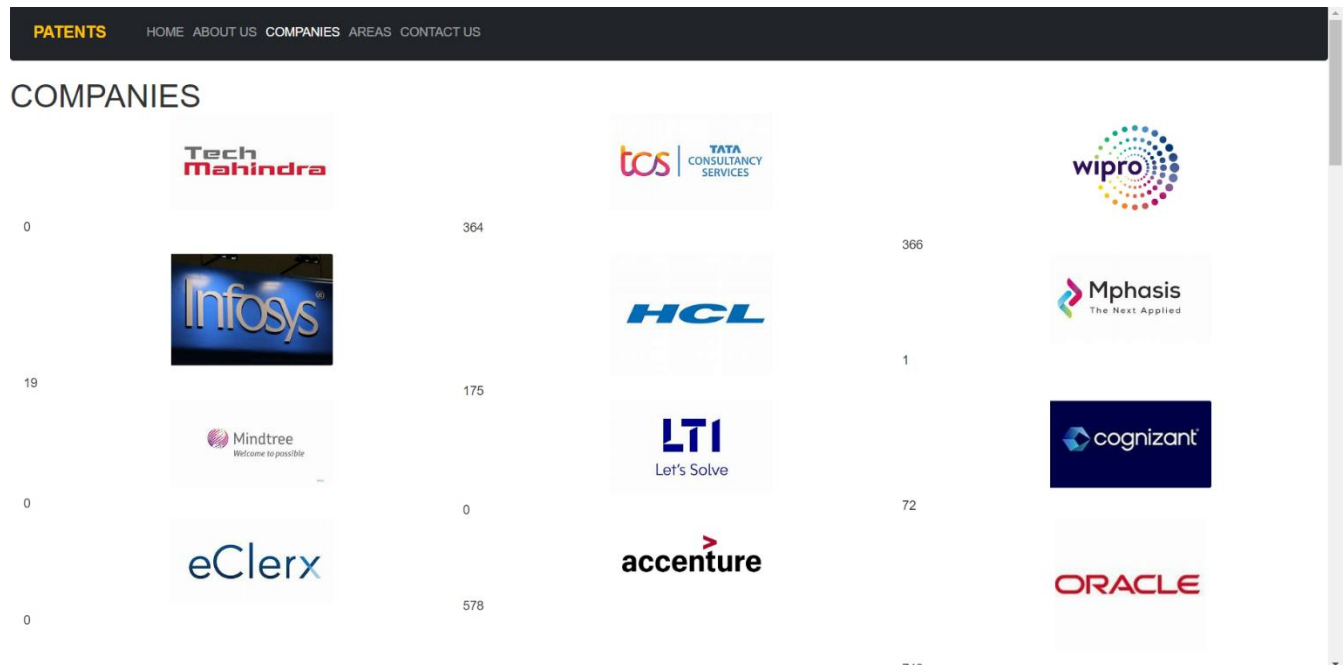
Company Name

Upload Image  No file chosen

**Figure 6.3: Screenshot for Add Company**

## 6.4 Screenshot for Company Events

The figure 6.4 represents that admin can manage products. Here admin can manage Operations like add remove and update products.



**Figure 6.4: Screenshot for company Events**

# **CHAPTER 7**

## **CONCLUSION**

- The project we have developed is a web application for patent search and analytic tool. This application helps in searching company patents.
- This application is convenient, effective and easy to search patents and acquisition.
- This application give solution to most of the problems that were identified when performing the task like maintaining the patents manually.

## **CHAPTER 8**

### **FUTURE SCOPE & ENHANCEMENTS**

**The admin can enhance this in future by:**

- In the current Project, the filed data of the company count is displayed in companies' page. So, if admin want to search for new company, he shall download that company image from the browser and collect the patent information of the same company.
- Already exist companies patent count will be increased.

## **CHAPTER 9**

### **BIBLIOGRAPHY**

- [1] Applying UML and patterns: An Introduction to object-oriented Analysis and Design by Craig Larman; PHI; Third Edition; ISBN-10:0131489062, ISBN-13:978-0131489066
- [2] Database Modelling and Design, by Toby J. Teorey, Sam S. Lightstone, Tom Nadeau;5th Edition; Morgan Kaufmann; ISBN-10:0123820200, ISBN-13:978-0123820204
- [3] Database Design using Entity-Relationship Diagrams by Richard Earp;2nd Edition, April 16,2007; Auerbach Publications; ISBN-10:1439861765, ISBN-13:978-1439861769
- [4] Structured systems Analysis and Design: Data flow Approach, by V.B Kaujalgi, 2nd Edition. Orient Black swan; ISBN-10:0863113230, ISBN-13:978-0863113239
- [5] Software Testing, by Ron Patton;2nd Edition; Sam's publishing; ISBN-10:0672327988, ISBN 13:978-0672327988
- [6] Software Testing concepts and Tools, by Nageshwar Rao Pusuluri; 2nd Edition; Dreamtech press;ISBN-10:8177227122,ISBN-13:978-8177227123