Project Report

Ceritficate

**ACKNOWELDGEMENTY**

* Our self Chodvadiya Neel S, Balar Axit M, Dhanani Rohan J. Have opportunity to express our knowledge. we would like to express our gratitude to all those who gave us the possibility to complete our project.
* Success is such a comprehensive project can not achieve single-handed. It is team effort that sails the ship to the coast. so we would like to express our sincere thanks to all the dignitaries who were involve in making this project the great joy and turning it our into successful piece of work.
* **Prof. Riddhi Vyas** – our professor and project co-ordinates has been very prudent to us threw out the collage studies. They are the person who has giving this direction to our work and the shape to our imagination. we express our regard to them from the core of our heart. we also like to thanks our all the professor who are always ready to give best guide. they are the person who give solution whenever needed.
* We would also like to acknowledge all the friends and colleagues, team member for the help and encouragement by them for time to time. The constant support and encouragement of my friend is deeply appreciates. The project indeed gave challenging and exhilarating experience in designing and developing the required system.

From,

Chodvadiya Neel S,

Balar Axit M,

Dhanani Rohan J.

**INDEX**

|  |  |
| --- | --- |
| **Chapter** | **Page No.** |
| **1. Introduction**   * 1. Project description   2. Project Profile   **2. Environment Description**   * 1. Hardware and Software Requirements   2. Technologies Used   **3. System Analysis and Planning**   * 1. Existing System and its Drawbacks   2. Feasibility Study   3. Requirement Gathering and Analysis   **4. Proposed System**  4.1 Scope  4.2 Project modules  4.3 Module vise objectives/functionalities Constraints  4.4 Expected Advantages  **5. Detail Planning**  5.1 Data Flow Diagram / UML  5.2 Process Specification / Activity Flow Diagram  5.3 Data Dictionary  5.4 Entity-Relationship Diagram / Class Diagram  **6. System Design**  6.1 Database Design  6.2 Directory Structure  6.3 Input Design  6.4 Output Design  **7. Software Testing**  **8. Limitations and Future Scope of Enhancements References** | **1-3**  2  3  **4-6**  5  6  **8-12**  9  10  12  **14-17**  15  15  16  17  **18-39**  19  26  32  39  **40-57**  41  56  57  57  **75-78**  **79-81** |

1. **Introduction**

* **Project Description**
* **Project Profile**

* **Project Description**

* A blog is a frequently updated online personal journal or diary. It is a place to express yourself to the world. A place to share your thoughts and your passions. Really, it’s anything you want it to be. For our purposes we’ll say that a blog is your own website that you are going to update on an ongoing basis.

* Blog is a short form for the word we blog and the two words are used interchangeably. Blogs range from the personal to the political, and can focus on one narrow subject or a whole range of subjects.
* It can also play an important role in student’s life. It can help in the promotion of critical and analytical thinking, increased access and exposure to quality content and a combination of solitary and social interactions with peers.
* Currently students in schools or colleges are unable to express their ideas, their talent or anything that can expressed for some benefits for everyone.

* The reason is, schools or colleges don’t have any proper medium to accomplish it. But proposed online blogging system can help in accomplish these things and even much more. We will cover the objectives of this online blogging system in next section.
* Internet has become reality and usage of internet become very much popular and there is tremendous increase of internet in all over the world for educational purpose. The Online Blogging System is easy to use, full-featured and much more.
* **Features :-**
* Online Blogging System is a perfect platform for students, teachers, institutional , administrative purposes.
* Before take any type of service, User have to register first.
* Home page display all categories of Blog and The Online Blogging System will allow the users to publish the writings, images if he/she should have credentials to login.
* After the user insert the blog, the blog goes to the admin and the admin check and verify the blog and then the blog is displayed to the rest of the users.
* User and admin can change their password.
* Admin can manage all type operation of system. Admin has every permission.
  + 1. **User Panel Features: -**
       - Login
       - Participant / cancel event registration.
       - View all activities.
       - Profile update.
       - Searching winner list.
       - Receive notice.
    2. **Admin Panel Features: -**
       - Login
       - Profile update
       - Change password
       - Create & manage event
       - Give rank to participated candidates
       - View event status
       - Send / receive notice
    3. **Blog Panel Features: -**
       - Login
       - Profile update
       - Change password
       - Confirm/Reject a delete event request
       - Manage student records
       - Manage Co-Ordinators
       - Check Co-Ordinator’s log
       - Send / receive notice
       - View event status
       - Backup Facility
    4. **System Admin Panel Features: -**
       - Login
       - Profile Update
       - Change Password
       - Create Notice
       - Manage College data

* **Project Profile :**

|  |  |
| --- | --- |
| Field | Details |
| Project Name: | IBlogger |
| Technology: | Node.js, React.js, Express.js, MongoDB, Html, Css, Bootstrap, Javascript. |
| Front End: | React.js |
| Back End: | Node.js / Express.js |
| Browser: | All browsers are supported. |
| Internal Guide: | Prof. Ridhhi Joshi |
| Platform: | Node.js -16.14.0 , React.js -17.0.2 |
| Tool Used For:- | Visual Studio Code, MongoDB Compass,  Postman. |
| Submitted To: | Shree Sambhubhai V Patel College  Of Computer Science & Business  Management |
| Developed By: | Chodvadiya Neel S.  Balar Axit M.  Dhanani Rohan J. |

1. **Environment Description**

* **Hardware and Software Requirements**
* **Technologies Used**

* **Hardware And Software Requirement :**

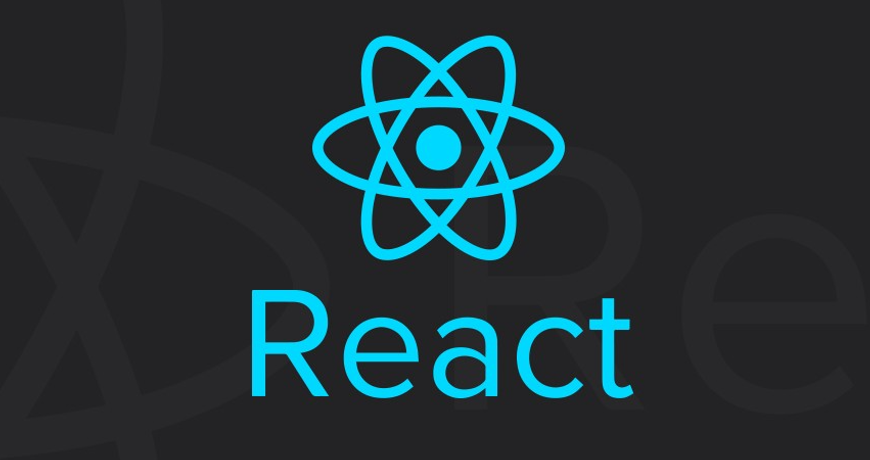
The efficient hardware and software configuration require running the system is as suggest below. The configuration suggested is for better performance. Same functionality or higher configuration will always better.

* **Client Side :**

* Internet enabled device with web-browser.
* **Server side :**
* Node.js - 14.16.1
* MongoDB Database
* **Development Side :**
* **Processor :** Intel core i5 11th generation
* **O.S :** Windows 10
* **Memory :** 8.00 GB
* **Hard disk :** 1TB
* **Web Browser :** Google Chrome (Recommended)

* **Technologies Used :**

* **Overview of React.js :-**



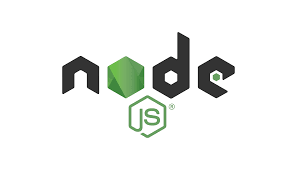
* React is a JavaScript library for building user interfaces.

.

* React is a library for building compassable user interfaces. It encourages the creation of reusable UI components, which present data that change over time.
* React abstracts away the DOM from you, offering a simpler programming model and better performance.
* React can also render on the server using Node, and it can power native apps using React Native.
* React implements one- way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

**React Advantages :-**

* React Uses virtual DOM which is a JavaScript object. This will improve apps performance, since JavaScript virtual DOM is faster than the regular DOM.
* React Can be used on client and server side as well as with other frameworks.
* React Component and data patterns improve readability, which helps to maintain larger apps.
* **Overview of Node.js :-**



* Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine).
* Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.
* Following are some of the important features that make Node.js the first choice of software architects.
* **Asynchronous and Event Driven** :

All APIs of Node.js libraries are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.

* **Very Fast** :

Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.

* **Overview of JAVASCRIPT** :-



* Java script is a programming language that can be included on web pages to make them more interactive. You can Use it to check or modify the Content of forms, change images. Open new widows and write dynamic page content.
* This allows you make parts of your web pages appear or disappear or move around on the page.
* JavaScript is a client side, interpreted, object oriented, high level scripting language, while Java is a client side, compiled, object oriented high level language.
* **JavaScript Feature** :-
  + JavaScript is more flexible language.
  + JavaScript make possible all type of validation and security.
  + JavaScript has set of function.
* **Overview of CSS** :-



* CSS is the languages we use to style an HTML document.CSS describe how HTML elements should be displayed. This tutorial will teach you CSS from basic to advance.
* **Benefits of CSS** :-
* Greater typography and page layout controls
* With style sheets, you can specify traditional typography features that you could never do with HTML alone (even with its presentational extensions).
* Less Work
* Not only can format all similar elements in a document with a single style rule, external style sheets make it possible to edit the appearance of an entire site at once with a single style sheet edit.
* Potentially smaller documents
* Redundant font tags and nested tables make for bloated documents. Stripping Presentational HTML out of the document save on file size.
* **Overview of HTML** :-



* Hyper Text Markup Language (HTML) is the set of markup symbols or codes inserted into a file intended for display on the Internet. The markup tells web browsers how to display a web page's words and images.
* Each individual piece markup code (which would fall between "<" and ">" characters) is referred to as an element, though many people also refer to it as a tag.
* Some elements come in pairs that indicate when some display effect is to begin and when it is to end.
* Hyper Text Markup Language is the computer language that facilitates website creation.
* The language, which has code words and syntax just like any other language, is relatively easy to comprehend and, as time goes on, increasingly powerful in what it allows someone to create.
* **Overview of BootStarp** :-



* Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.
* It would be easy to send you over to their Getting Started page and call it a day. Their setup guide is indeed a host of useful information - links to CDNs, explanations on how to install with Bower, NPM, and Composer, information on integration with Auto prefixer and LESS, a bunch of templates, licenses, and translations.
* Bootstrap employs a handful of important global styles and settings that you’ll need to be aware of when using it, all of which are almost exclusively geared towards the normalization of cross browser styles.
* It is a front-end framework used for easier and faster web development.
* It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others. It can also use JavaScript plug-ins.It facilitates you to create responsive designs.
* **Overview of MongoDB :-**



* MongoDB is a [source-available](https://en.wikipedia.org/wiki/Source-available) [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [document-oriented database](https://en.wikipedia.org/wiki/Document-oriented_database) program. Classified as a [NoSQL](https://en.wikipedia.org/wiki/NoSQL) database program, MongoDB uses [JSON](https://en.wikipedia.org/wiki/JSON)-like documents with optional [schemas](https://en.wikipedia.org/wiki/Database_schema). MongoDB is developed by [MongoDB Inc.](https://en.wikipedia.org/wiki/MongoDB_Inc.) and licensed under the [Server Side Public License](https://en.wikipedia.org/wiki/Server_Side_Public_License) (SSPL).
* MongoDB Atlas also includes powerful features to enhance reliability for your mission-critical production databases, such as continuous backups and point-in-time recovery.
* MongoDB Atlas makes it easy to control access to your database. Your database instances are deployed in a unique Virtual Private Cloud (VPC) to ensure network isolation.
* MongoDB Atlas automates infrastructure provisioning, setup, and deployment so your teams can get the database resources they need, when they need them. Patches and minor version upgrades are applied automatically.

1. **System Analysis and Planning**

* **Existing System and its Drawbacks**
* **Feasibility Study**
* **Requirement Gathering And Analysis**

* **Existing System and its Drawbacks :**

* **Our aim is to reach people through the latest news and students also have a way of learning new languages by instant blog.**

* **Feasiblity Study :**

* All projects are feasible, provided that unlimited resources and infinite time are available. However, in real world that can sound like dream. Especially computer-based systems are likely to be bounded by limited resources as well as time. Feasibility & risk is related in many ways. If the risk is great then feasibility of producing quality software is reduced.

### * Technical Feasibility :-

* This project does require that much of higher & advanced technology. It’s requires database Interaction and it requires to be accessed via web or internet. This can be easily done. It must be developed within the four months of period excluding the time period for the testing and validation, verification. Thus, it seems that the project is technically feasible to do.

### * Economic Feasibility :-

* In This project, we will require to not purchasing much of the tools like web cameras to take the photograph of a person or shut Video of that person, but as it will be web-enabled we do not have any extra cost of setting up a network. This is also feasible economically.

### * Operational Feasibility :-

* The web application can be beneficial only if it satisfies the organization requirement; in such a way that resource utilization & optimum outcome is justified. A web application should not only be robust but should also be able to work Simultaneously with other systems. Operational feasibility means that web application should not affect any existing system during the development phase or even in the implementation phase

### * Management Feasibility :-

### .

* Management feasibility ratio and aspect of management News. Here all level of management rise of related basic feasibility and gives their decision for all those feasibilities to use or not is not comes in basic feasibility of system, bus how the requirement of that all, after that all manager.

### * Time Feasibility :-

* Time feasibility describes the time cost for converting the existing system to web application. As a for completion. This System is Develop under the time period of just four months so we can say that this system is time Feasible.

* **Requirement Gathering and Analysis :-**

* **Requirement Gathering :**
* Gathering requirements for the project is the most important part of the software development.
* There are two main reasons for specifying requirements.

1. It guides design and decision-making processes towards a correct solution.
2. Provides a basis of testing the implemented solution is correct.

* For gathering basic needs and functionalities, we had meeting with people personally to understand required functionalities also noted down some useful feature to be included.
* Every successful system passes through requirement gathering because without any requirement the development process is like adding bugs in empty system.
* Requirement gathering and analysis makes the whole picture of the system and because of that we can identify the features, the modules, the functionality of the system.
* **Requirement Analysis :**
* System should be able to provide registration facility.
* System should provide safe login facility.
* System should provide forgot password facility.
* System should manage email Contact, Skill.
* System should able to add or remove user and blog company and also manage its information.
* System should able to manage Review, Designation, Staff.
* We analyzed our gathered information and we have decided our system should have following functionalities :-
  + **Common models**
    - Login
    - Forget Password (Co-Ordinator / Admin)
    - Add Blog
    - View Profile
    - Like Blog

### Module 1 ( USER )

* + Login
  + Logout
  + Forgot Password
  + Manage profile
  + Add Feedback

### Module 2 ( BLOG )

* + Add Blog
  + View Blog
  + Delete blog
  + Update Blog
  + Like Blog
  + Comment blog
  + Search Blog

### Module 3 ( ADMIN )

* + Login
  + Logout
  + View all events
  + Manage user
  + Manage blog
  + Manage feedbacks
  + Manage profile
  + Search Blog & User
  + Manage blog & user reports

### Module 4 ( FEEDBACK & COMMENT )

* + Add Feedback by user
  + Add comment by user
  + View Feedback & Comment
  + Send Feedback email

1. **System Analysis and Planning**

* **Project Scope**
* **Project modules**
* **Module wise objectives/functionalities Constraints.**
* **Expected Advantages**

* **Project Scope :-**

The Online Blogging System will allow the users to publish the writings, videos, images or audios if he/she should have credentials to login. The main users of this project are students, teachers and administrators.

From an end-user perspective, the Online Blogging System project consists of following functional elements:

* **Dashboard:** It is the default page of the site and we can access this option from left hand side anytime. All links are available on this page. We can also find the Quick Draft and Activity section here.
* **Blog:** Here we can see all the published contents by clicking on “All Blogs” option and we can also publish new content by “Create Blog” option.
* **Media:** We can see the uploaded media items ( images ) by clicking on “Add Image” option and we can also add new media item from local system with the help of “Add Image” button.
* **Comments:** In this section, we can check that who, when and what has been commented.
* **Profile:** In this section, we can personalize our profile like Password Change, Profile Picture Change, Display Name, Nickname etc.
* **Tools:** This option is having additional plugins to install, which may enhance the current functionality.
* **Feedbacks:** User can give feedbacks about our website and blog.
* **Home Button:** It is located at the top right portion and we can use this button to check the timeline where we can view the contents published by everyone.
* **Project Category** : It is an application that can be accessed over web.

* **Project Modules :-**

1. **User**
2. **Admin**
3. **Blog**
4. **Comment**
5. **Experts**

* **Modules Wise Objectives/functionalities Constraints :-**

1. **User :-**

* User can register It self.
* User can login
* User can logout
* User can manage their profile.
* User can add blog
* User can view blog
* User can like blog
* User can change password

1. **Admin :-**

* Admin can upload/delete/edit blog
* Admin can add/delete/edit User
* Admin can manage user feedback
* Admin can manage their profile
* Admin can change password

1. **Blog :-**

* User & Admin can add blog
* User & Admin can view blog

1. **Comment :-**

* User & Admin can add comment on blog.
* User & Admin can view like on blog.

1. **Experts :-**

* All User give feedback about our website and blog to experts.

* **Expected Advanteges :-**

#### **Easy to use**

* The system is easy to use for every user such as user, co- ordinator, admin and system-admin. Easily manage their work with amazing features like add blog, sending email, verify and unverified blog, Notify for every update in event etc.

#### **User friendly interface**

* The interface of every panel is minimal and user friendly. Every panel like user, co-ordinator, admin and system-admin is fully responsive and run in most of browser such as Google chrome, Mozilla Firefox, Opera etc.

#### **Easy to manage**

* It is easy to manage every activity and blog for admin and co- ordinator. Not for only admin and co-ordinator but also user can manage their blog in website.
* Ex. user can also delete your blog. Admin and Co - Ordinator can view the user and its blog and update the user blog.

#### **Report generating feature**

* Admin can generate report of user detail and total blog details.
* Ex. Admin can view particular blog user, verify and unverified the blog after check it, read the top 10 blogs.
* In blog system, Admin can view the total user of our system and total blog of our system, and also view blog total likes, generate the feedback users report.

#### **Security**

* The system is secured itself because of functionality and features of Node.js – 16.18.2. Not only security of Node.js but we have done encryption and decryption of Password when it passes data from one page to another.
* Ex. Password encryption-decryption

#### **Reduce time & manpower**

* On desk registration system has many drawbacks like difficulty in managing blogs, difficulty in registration, time consuming and it takes lots of manpower.
* In this system, all the task is done through online with the help of internet and various functionality of system.
* Ex. Create and manage blog through the system, user can create and read the blog through the system etc.

1. **Detail Planning**

* **Data flow diagram/UML**
* **Process specification / activity flow diagram**
* **Data dictionary**
* **Entity-relationship diagram / class diagram**

* **Data Flow Diagram/UML :-**

* A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and Outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

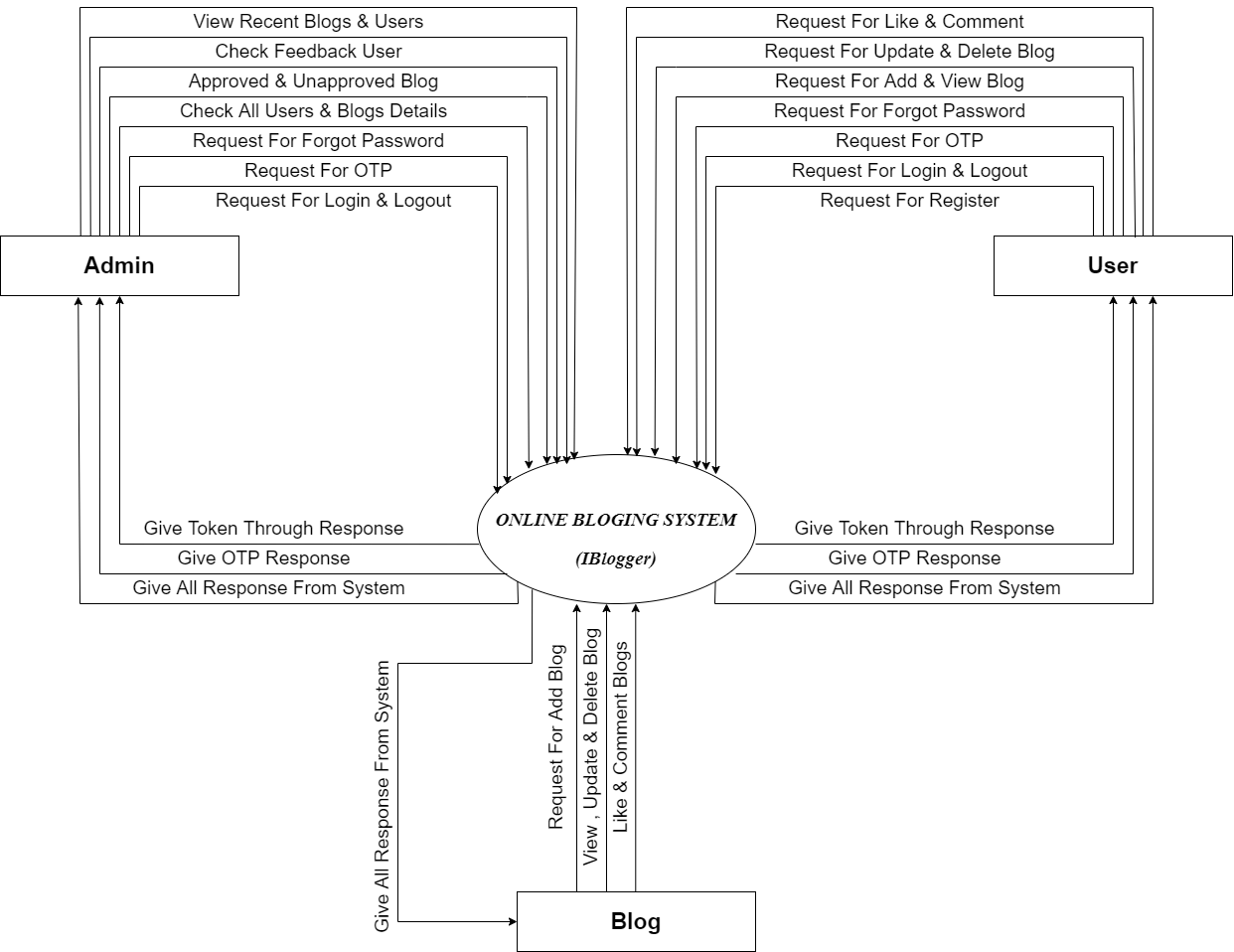
### Strict landing page policy :

* A data flow diagram needs to be simple because user has to go through it, understand suggest correction or changes.
* External entity
* Data flow
* Process
* Data stores

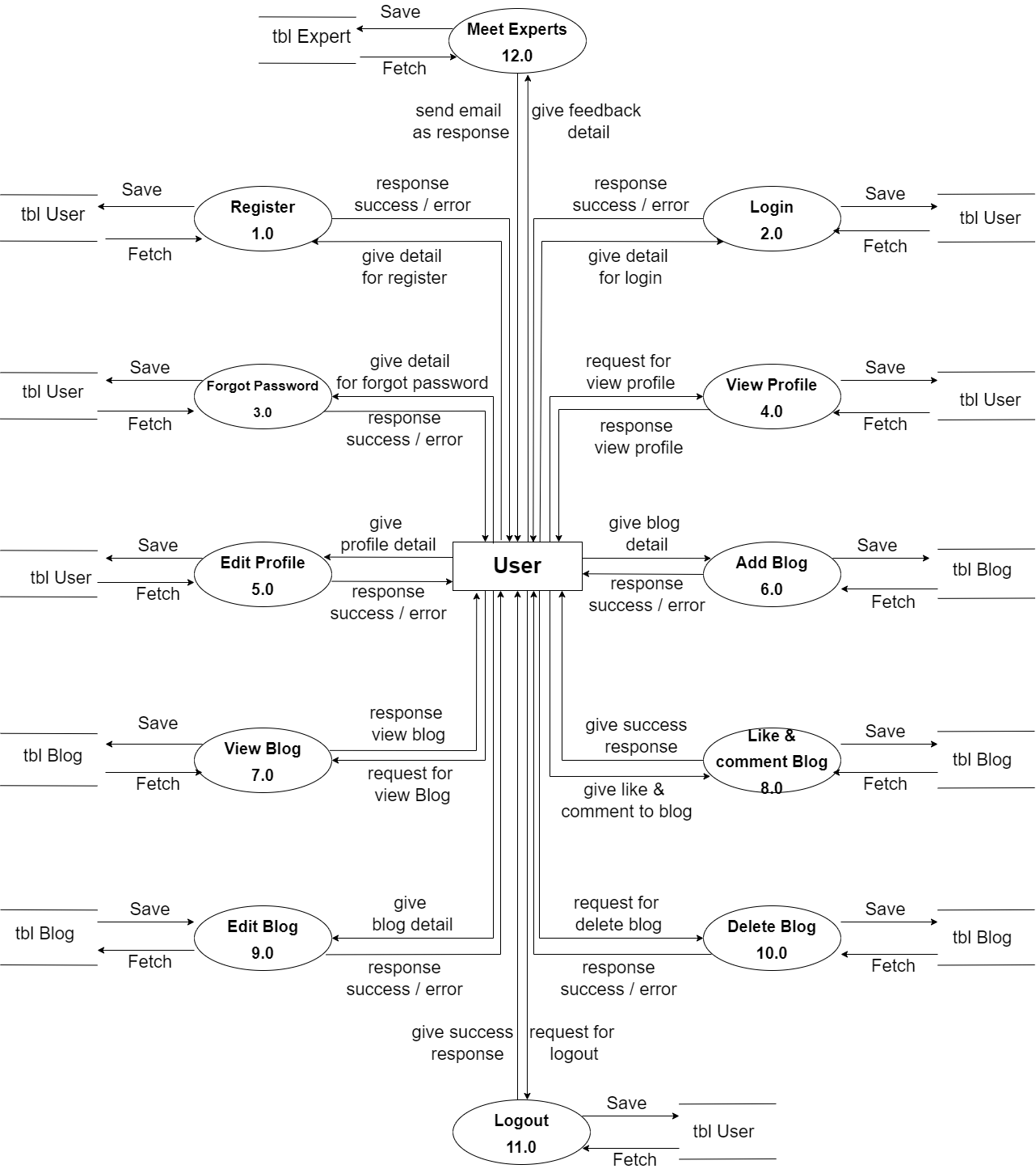
### Symbols used in DFD:

* + 1. **Process** : -
    - Flow of data is transformed.
    - A Process represents some amount of work using behavior of data.
    - Transformation of data from one form to another.
    - A circle represents a process.
    - The area of the circle is divided into two parts.
    - The process number is written in the top portion and process name is written the bottom portion.
    - E.g. verify credits, update inventory file, etc.
    1. **External Entity :-**
    - A source or destination of data which is external to the system.
    - As the name suggest it line outside the context of the system. It represented by solid squares.
    - The people/department that will be giving the data and/or receiving information.
    - It can be referred to as external entities as they do not accomplish any word done by the system.
    - The data coming in and the report produced are referred as the input and Output of the system respectively.
    1. **Data Flow :-**
    - Line is packet of data. it may be of form of documentation, letter etc.
    - A data flow portrays an interface among different components in the flow diagram.
    - It represents the data of the data as it flows through a system. An arrow represents it.
    - The arrow head point in the direction in which the data moves.
    1. **Data Stores :-**
    - Any stores data but no reference to the physical method of storing.
    - If there is a logical requirement for the data to be stored, it is held in a data store.
    - A data store there for, is a repository of the data.

### Context Level DFD



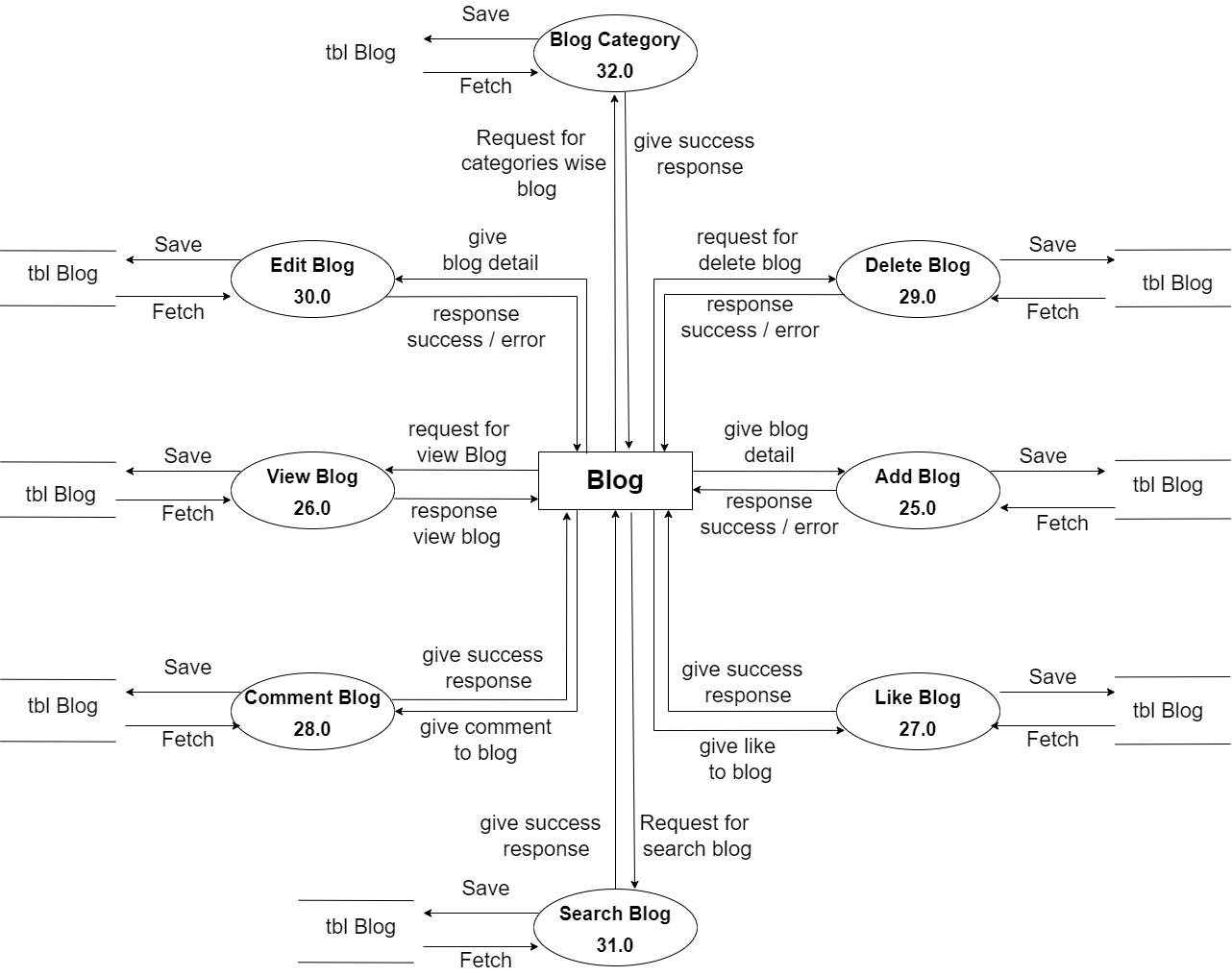
### 1st Level DFD ( User )



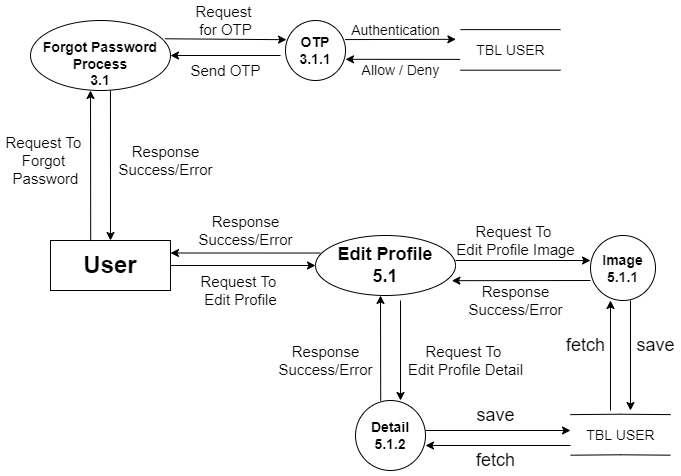
### 1st Level DFD ( Admin )

### 

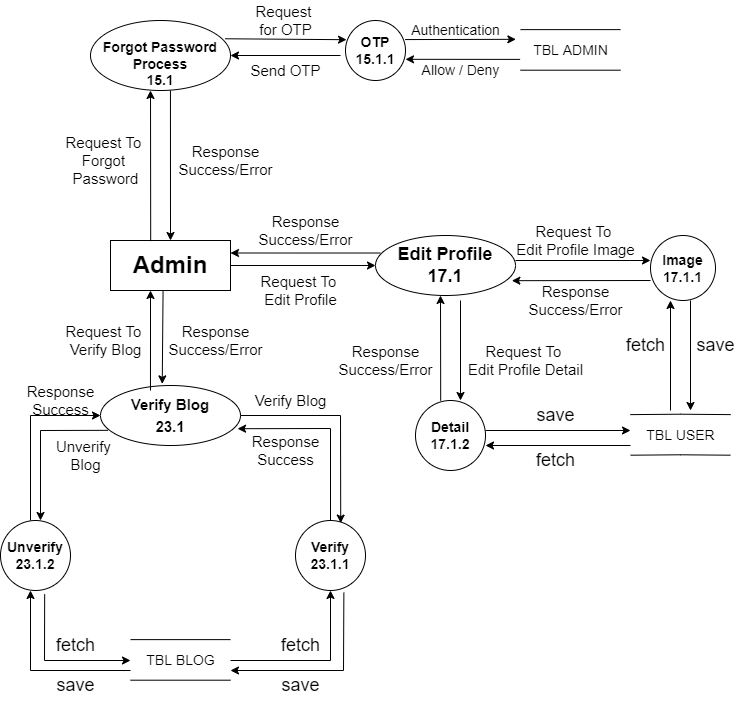
### 1st Level DFD ( Blog )



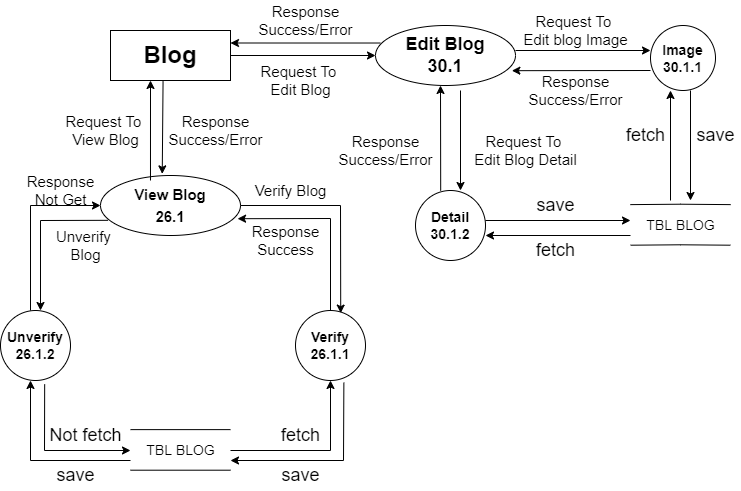
* **2st Level DFD ( User )**



* **2st Level DFD ( Admin )**



* **2st Level DFD ( Blog )**



* **Process Specification :-**

### User Register

### 

### User Login

### 

### User Forgot Password

### 

### User View Profile

### 

### User Edit Profile

### 

### User Add Blog

### 

### User View Blog

### 

### User Like & Comment Blog

### 

### User Edit Blog

### 

### User Delete Blog

### 

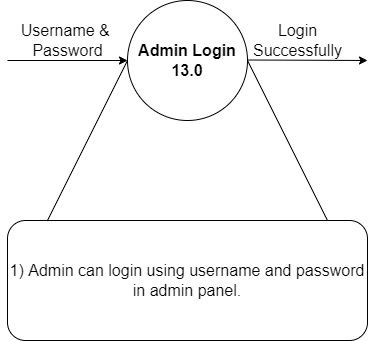
### User Logout

### 

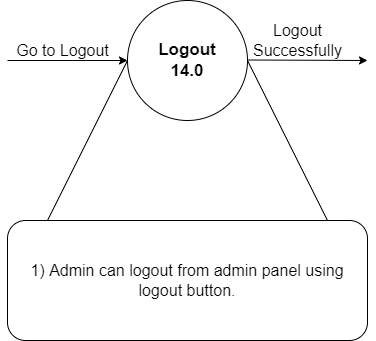
### User Meet Experts

### 

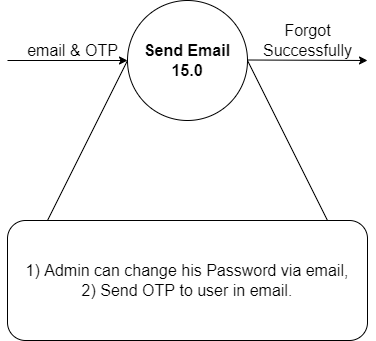
* **Admin Login**



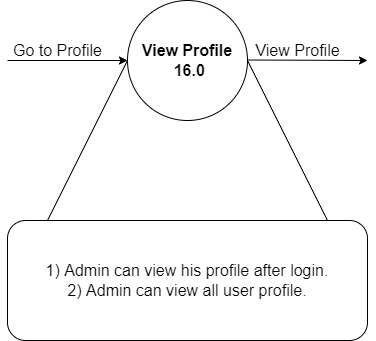
* **Admin Logout**



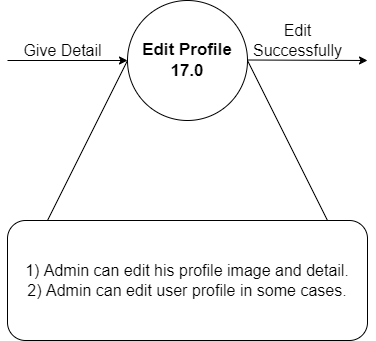
* **Admin Send Email For Forgot password**



* **Admin View Profile**



* **Admin Edit Profile**



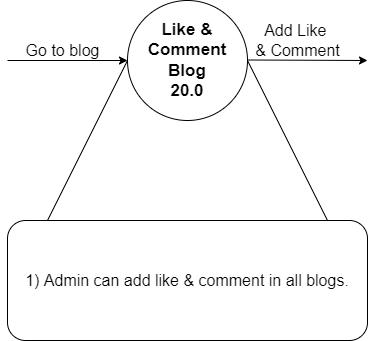
* **Admin Add Blogs**



* **Admin View Blogs**



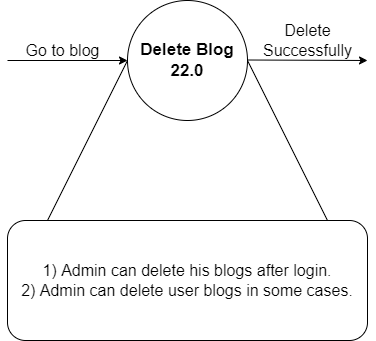
* **Admin Like & Comment blogs**



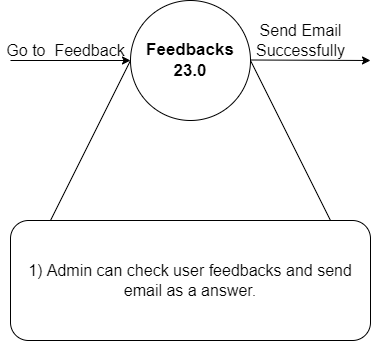
* **Admin Edit Blogs**



* **Admin Delete Blogs**



* **Admin Check Feedbacks**



* **Admin Verify Blogs**



* **Add Blogs**



* **View Blogs**



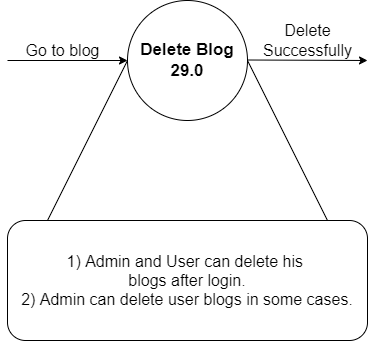
* **Like Blogs**



* **Comment Blogs**



* **Delete Blogs**



* **Edit blogs**



* **Data Dictionary :-**

* **User :-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Description |
| User\_id | Autogenerate | Null | Primary | User\_id is unique id of user data. |
| Name | String | Not Null |  | It displays user’s name. |
| Email | Email form | Not Null |  | It displays user’s email. |
| Gender | String | Not Null |  | It displays user’s gender. |
| Phone | Number | Not Null |  | It displays user’s phone number. |
| Username | String | Not Null |  | It displays user’s username. |
| Password | Password (encrypted) | Not Null |  | It displays user’s password. |
| Images | Files | Not Null |  | It displays user’s profile image. |
| Image path | String | Null |  | It displays user’s image path. |
| Token | String  (Autogenerate) | Null |  | It displays user’s login token. |
| Date & Time | String  (Autogenerate) | Null |  | It displays user’s register date and time. |

* **Admin :-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Description |
| Admin\_id | Autogenerate | Null | Primary | Admin\_id is unique id of admin data. |
| Name | String | Not Null |  | It displays admin’s name. |
| Email | Email form | Not Null |  | It displays admin’s email. |
| Gender | String | Not Null |  | It displays admin’s gender. |
| Phone | Number | Not Null |  | It displays admin’s phone number. |
| Username | String | Not Null |  | It displays admin’s username. |
| Password | Password (encrypted) | Not Null |  | It displays admin’s password. |
| Images | Files | Not Null |  | It displays admin’s profile image. |
| Image path | String | Null |  | It displays admin’s image path. |
| Token | String  (Autogenerate) | Null |  | It displays admin’s login token. |
| Date & Time | String  (Autogenerate) | Null |  | It displays admin’s register date and time. |

* **Blog :-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Description |
| Blog\_id | Autogenerate | Null | Primary | Admin\_id is unique id of admin data. |
| Source\_id | Autogenerate | Null | Foreigen | Source\_id is unique id of user and admin data. |
| Title | String | Not Null |  | It displays blog’s title. |
| Image | file | Not Null |  | It displays blog’s image file. |
| Tags | String | Not Null |  | It displays blog’s tags. |
| Short description | String | Not Null |  | It displays blog’s Short description. |
| Long  Description | String | Not Null |  | It displays blog’s long description. |
| Image path | String | Null |  | It displays blog’s image path. |
| Like | Number | Null |  | It displays blog’s total likes. |
| Comments | String | Not Null |  | It displays blog’s total comments. |
| Date & Time | String  (Autogenerate) | Null |  | It displays blog’s created date and time. |

* **Comment :-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Description |
| comment\_id | Autogenerate | Null | Primary | Comment\_id is unique id of comment data. |
| blog\_id | Autogenerate | Null | Foreigen | blog\_id is unique id of blog data. |
| Source\_id | Autogenerate | Null | Foreigen | blog\_id is unique id of user data. |
| Name | String | Not Null |  | It displays user name they push comment on blogs. |
| Comment | String | Not Null |  | It displays comment’s of blogs. |
| Tags | String | Not Null |  | It displays tag of comments. |
| Date & Time | String  (Autogenerate) | Null |  | It displays comment created date and time. |

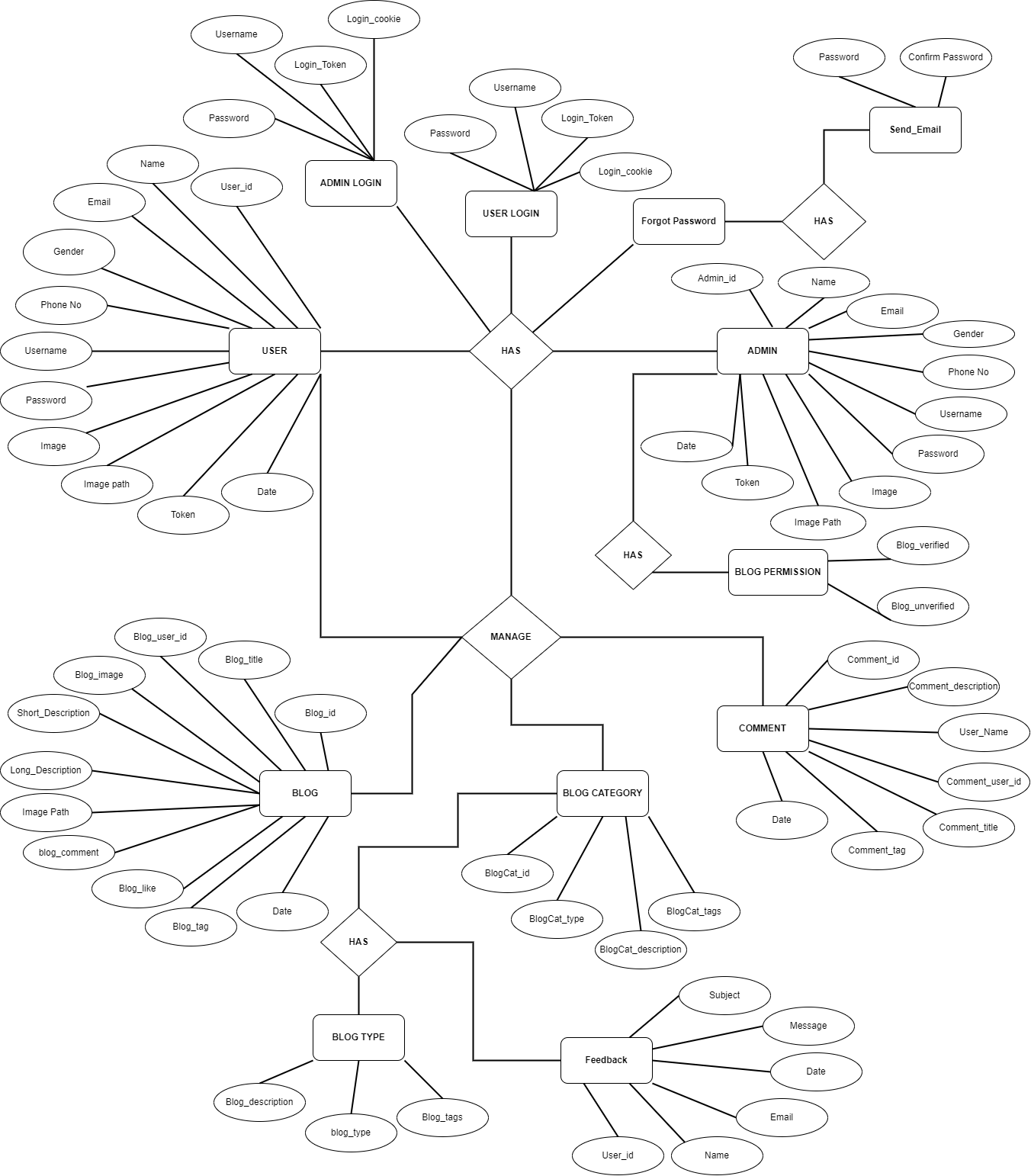
* **Forgot Password :-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Description |
| User\_id | Autogenerate | Null | Foreigen | User\_id is unique id of user data. |
| Email | Email form | Not Null |  | It displays user email they save in use database. |
| OTP | String | Not Null |  | It displays unique OTP for security |
| Password | Password  (encrypted) | Not Null |  | It displays new password of user. |
| Confirm Password | Password  (encrypted) | Not Null |  | It displays new confirm password of user. |
| Date & Time | String  (Autogenerate) | Null |  | It displays user data updated date and time. |

* **Feedback :-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Description |
| user\_id | Autogenerate | Null | Primary | user\_id is unique id of  feedback data. |
| Name | String | Not Null |  | It displays user name they give feedback. |
| Email | Email Form | Not Null |  | It displays user email. |
| Subject | String | Not Null |  | It displays subject of feedback topic. |
| Message | String | Not Null |  | It displays message of feedback. |
| Date & Time | String  (Autogenerate) | Null |  | It displays created date and time. |

* **Entity-Relationship Diagram :-**



1. **System Design**

* **Database Design**
* **Directory Structure**
* **Input Design**
* **Output Design**

* **Database Design :-**

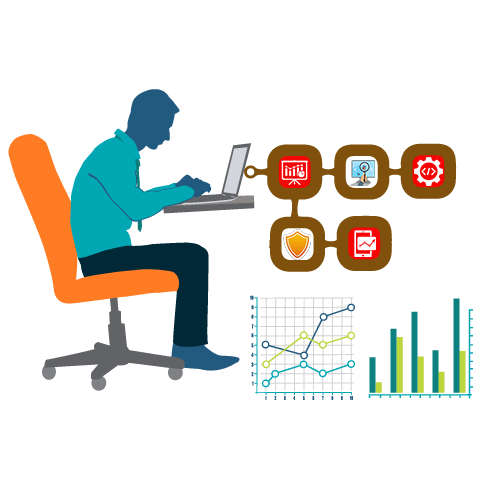
**e**

1. **Software Testing**

* **Software Testing**
* **Test Cases**

* **Software Testing :-**

* Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is an exposure of a system to trial input to see whether software meets correct output. Testing cannot be determined whether software meets user’s needs, only whether it appears to confirm to requirements. Testing can show that a system is free of errors, only that it contains error. Testing finds errors, it does not correct errors. Software success is a quality product, on time and within cost. Through testing can reveal critical mistakes. Testing should therefore,
* Validate Performance.
* Detects errors.
* Identify.
* Inconsistencies.



* **Test Objective :**
* There Is Strong Evidence That Effective Requirement Management Leads To Overall Project Cost Savings. The Three Primary Reasons For This Are:
* Errors in requirement typically cost over 10 times more to repair than other errors.
* Requirement errors typically comprise over 40% of all error in a software project.
* The Testing Procedure Should Care For All Of These, As Well As, In Order To Attain A Flawless, Error-Free And Efficient Functioning System; Too, Software Testing Is An Important Phase Of Any Software Development Life Cycle. Various Reports And Data Used For The Same Are The Core Of The System. The Testing, Therefore, Becomes Important In Order To Maintain The Cost As Well As Improve Performance And Consistency. The Testing Procedure For The System Has Been Divided In To Various Parts Ranging For Single Unit Testing To Entire System Testing.
* **Testing Principles :**
* All test should be traceable to customer requirements.
* Tests should be planned long before testing begins.
* The pareto principle applies to software testing.
* Testing should begin “in the small” and progress toward testing “in the large”.
* Exhaustive testing is not possible.
* **Unit Testing :**
* Unit Testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output.
* **Integration Testing :**
* **Integration Testing** is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing.
* **Top-down integration :**
* Top-down integration testing is an integration testing technique used in order to simulate the behaviour of the lower-level modules that are not yet integrated. Stubs are the modules that act as temporary replacement for a called module and give the same output as that of the actual product.
* **Bottom-down integration :**
* Bottom-up testing is a specific type of integration testing that tests the lowest components of a code base first. More generally, it refers to a middle phase in software testing that involves taking integrated code units and testing them together, before testing an entire system or code base.
* **Validation Testing :**
* The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements.
* Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.
* **System Testing :**
* System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.
* **Recovery Testing :**
* Recovery testing is a type of non-functional testing technique performed in order to determine how quickly the system can recove rafter it has gone through system crash or hardware failure. Recovery testing is the forced failure of the software to verify if the recovery is successful.
* **Security Testing :**
* Security testing is a process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended.
* **Stress Testing :**
* Stress testing a Non-Functional testing technique that is performed as part of performance testing. During stress testing, the system is monitored after subjecting the system to overload to ensure that the system can sustain the stress.
* **Sanity Testing :**
* Sanity testing is the subset of regression testing and it is performed when we do not have enough time for doing testing. Sanity testing is the surface level testing where QA engineer verifies that all the menus, functions, commands available in the product and project are working fine.
* **White Box Testing :**
* White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing [software](https://en.wikipedia.org/wiki/Software) that tests internal structures or workings of an application, as opposed to its functionality (i.e. [black-box testing](https://en.wikipedia.org/wiki/Black-box_testing)).
* In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.
* The tester chooses inputs to exercise paths through the code and determine the expected outputs.
* This is analogous to testing nodes in a circuit, e.g. [in-circuit testing](https://en.wikipedia.org/wiki/In-circuit_test) (ICT). White-box testing can be applied at the [unit](https://en.wikipedia.org/wiki/Unit_testing), [integration](https://en.wikipedia.org/wiki/Integration_testing) and [system](https://en.wikipedia.org/wiki/System_testing) levels of the [software testing](https://en.wikipedia.org/wiki/Software_testing)process.
* Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today.
* It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test.
* Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

1. **Limitations and future Scope of Enhancements, References**

* **Limitations of the system :-**

* **Future scope of enhancements :-**

* **References :-**