A

Project Report on

Bench Management

Submitted for the award of

the Degree of

Master of Computer Applications (MCA)

Submitted by

GOBIND

Regd. No.: - 2020PGCACA38

Under the supervision of

Dr. Koushlendra Kumar Singh
(Internal Guide)
Dept. of Computer Applications
NIT Jamshedpur

Mr. Sumit Agarwal
(External Guide)
Technical Program Manager
Accolite Digital

Department of Computer Science & Engineering

National Institute of Technology,

Jamshedpur May, 2023

Department of Computer Science and Engineering National Institute of Technology, Jamshedpur

(An Institution of National Importance under MHRD, Govt. of India, New Delhi)

Ref. No. ………………….   *Date*: …………….

CERTIFICATE

This is to certify that project entitled “Bench Management”, is a bonafide work of Gobind a student of Master of Computer Applications, sixth semester of the Department of Computer Science and Engineering, National Institute of Technology, Jamshedpur. This work has been carried out at “Accolite Digital” from 16th Feb 2023 to 20th April 2023. This report is submitted in partial fulfillment of the requirement for the award of degree of Master of Computer Applications and has been carried out under my supervision.

I wish all the best in his career and future endeavor.

Prof. Koushlendra Kumar Singh
(Internal Guide)
Dept. of Computer Science and Engineering,
NIT Jamshedpur

DECLARATION

I hereby declare that dissertation of the work titled, **“Bench Management”**, submitted towards requirements of project work for partial fulfilment of **Master of Computer Applications (MCA))** is an original work of mine and the report has not formed the basis for the award of any other degree, associateship, fellowship or similar titles. I also declare that wherever I have used materials such as data, theoretical analysis, and text from other sources, I have given due credit to them by citing source of the work in the thesis.

Gobind
2020PGCACA38

Place: …………

MCA. 6th Semester

Date: ………….

Dept. of Computer Science & Engineering
NIT Jamshedpur

ACCOLITE DIGITAL
Transforming The Future, Now

TO WHOMSOEVER IT MAY CONCERN

Date: 18/Apr/2023

This is to certify that **Mr. Gobind,** pursuing his Master's in Computer Applications at the National Institute of Technology, Jamshedpur is engaged as an intern at **Accolite Digital India Private Limited, Gurugram**. The current internship period is from 14th February 2023 to 14th August 2023.

Gobind is currently working on **Bench Management System** project.

Warm Regards,

Milind Mutalik
Head HR - Employee Experience
Accolite Digital India Pvt. Ltd

Signature ________________________

ACKNOWLEDGMENT

First of all, I thank all the Almighty God for his endless blessings. I would like to thank my Mentor **Mr. Sumit Agarwal (Technical Data Engineer)** for making me familiar with the intricacies of Software Development. The way he supported me and showed me the way during the training period.

Sincere thanks to all my colleagues at **“Accolite Digital”** for their support and assistance throughout the project. It is definitely a wonderful experience to have such a good surrounding of people who have helped me on every step where I might have fallen.

My heartfelt gratitude also goes to the H.O.D. D.C.A., **Dr Danish Ali Khan,** and Dean (Academic) **Prof. Amresh Kumar**, National Institute of Technology, Jamshedpur for providing me with the opportunity to avail the excellent facilities and infrastructure.

Thanking my family members and my friends will be too petty an effort. With no intention of belittling their contribution, I just want to say that it is all their love and blessings that keeps me going.

I also extend my apologies for all errors or omissions, which are solely my responsibility.

Last but not the least I am grateful to Planful for giving me this wonderful opportunity.

Gobind
2020PGCACA38
MCA. 6th Semester
Dept.of Computer Science & Engineering
NIT Jamshedpur

Place: ……………………

Date: ……………………

PREFACE

The aim of this project is to make a web application in which The Bench Manager can place the employees on Bench based on Project Requirements.

Bench Management System allows you to add new employees and update their Information, place a particular employee at a particular location.

The Bench Management is managed by admin user permission system where a admin having specific permission can do the related tasks .

The Purpose of this project is to create a system that could manage and monitor the Employees which are on Bench.

ABSTRACT

This project is all about to place the employees on the bench and place not on the bench based on the project requirements. It contains Front-end part for display the content and Back-end part for storing the information.

It also has implementation of Manager management system which is managed by the Admin and also have the employees management system which is managed by the both Admin and Manager.

The Purpose of this project is to create a system that could manage and monitor the Employees which are on Bench

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Bench Management

Gobind

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO THE PROJECT

Bench simply means:

Pool of resources rolled off from respective projects because either the project has ended or the resource has been moved against under performance.

Bench Management:

The key is maintain optimal rate , it enables to procure new business, gets in flexibility to pitch in resources to critical projects with in the TAT, Employee and organization get balanced time to develop skills and enhance capability between projects.

Utilization process works around Bench supply plan process, hiring request assessment, skill assessment, Gap assessment, Training and development, Resource allocation and bench resource termination/utilization in typical india IT services sector.

Balance between Capacity and Demand is what an ideal Bench is,but on ground organization players struggle to manage bench with percentage ranging between 10 to 30% anytime around the year with bench duration averaging between 60 days to 180 days and more in some critical cases.

1.2 OBJECTIVE

The main objective it to provide the ease in managing the data analysis plateform and details are following:

-

Module : - Employees Management System

- Creating Employees management system
- Creating management system as assigning roles and permissions
- Creating account according to the role.
- Creating one main Employee which is known as admin.
- Creating other Employees which are under the admin user.
- Creating result docs.
- Creating different type of analysis reporting.
- Creating the report for their better understanding.
- Restrictions are made through the JWT/Google authentication
- Creating User
- Creating Role
- Creating Permissions
- Edit Employee Details
- Assign Employee Role
- Edit Employee Roles
- Assign Employee Role
- Edit Employee Roles
- Activate/Deactivate Employee
- Assign Permissions to Roles

1.3 OVERVIEW OF THE ORGANISATION

ACCOLITE DIGITAL

Transforming The Future, Now

Transforming the Future, Now

We believe in solving real-world complex digital challenges with an innovative design thinking approach. As a best-in-class digital transformation services provider, we work with Fortune 500 clients to simplify their digital journeys. Our people are the heart of our business. We are a growing team of world-class technologists, passionate about helping our clients solve their most pressing technology challenges. Driving customer success is one of our core values, and we leverage our expertise to guide our clients from vision to reality, enabling them to respond to disruptive technologies and stay ahead of the curve.

Insights

Cloud Engineering and DevOps

Replatform: The Why and How

Cloud Engineering and DevOps

Rehost: The Art of Lift and Shift

Technology

Embracing digital transformation in the post pandemic era

Cloud Engineering and DevOps

Building your business case for Cloud Migration

CHAPTER 2

ANALYSIS

2.1 EXISTING SYSTEM

There was no existing system before this so there is very crucial need to make it. The work is managed by using the REST API. This allows you to control access to the report, update the report data, and monitor usage and performance.

2.2 DRAWBACK

There are few potential drawbacks to consider: -

- Cost: Bench Management can be expensive, particularly if you are embedding for a large number of users. The cost is based on the number of render anddata load operations, which can add up quickly.
- Complexity: Setting up and configuring Bench Management can be complex, particularly if you are not familiar with web development. You mayneed to hire a developer or consultant to help you get started.
- Maintenance: Once you have Bench Management, you need to maintain it to ensure that it continues to work properly. This can involve monitoring usage and performance, updating data sources, and addressing any issues that arise.

2.3 PROPOSTED TECHINCAL REQUIREMENT

- Table 1 : Software Requirements:-

Operating System	Window , MacOS , Linux
Programming Language	Javascript,Java
Technology	Microserverice
Tools	Vishual Sudio Code and Intellij
Databases	SQL
Frameworks	React

- Table 2 : Web Browsers

Browers	Requirement
Chrome	Version 60 and above
Safari	Version 10.1 and above
Firefox	Version 60 and above
Edge	Version 41 and above

2.4 PROJECT PLANNING AND PROJECT SCHEDULING

2.4.1 PROJECT PLANNING

After the project has been defined and the project team has been appointed, the project moves on to the second phase in the Project Management Life Cycle (PMLC): the detailed project planning phase. Project planning is at the heart of the project life cycle that tells everyone to get involved that know where it is going and how it is going. In the planning phase, the project plans are documented, the project deliverables and requirements are defined, and the project schedule is created. It involved creation of a set of plans to guide the team through the implementation and closure phases of the project. The plans are created during this phase are to manage time, cost, quality, changes, risk, and related issues. This helped to control staff and external suppliers to ensure the delivery of the project on time, within budget, and within schedule. Based on that the project is planned so that on estimated date the project is accomplished and proper documentation is completed**.**

2.4.2 PROJECT CONSTRAINTS

- Scope-Project outcome as defined in the contract.
- Timeline- Important milestones and client-imposed completion dates.
- Budget -Funding limits is based upon the time that is required and imposed by project sponsor i.e., Client Partner.
- Quality- Agreed quality metrics with Customer or conformance with internal quality standards.
- Resources- Availability of skilled human resources or materials.
- Risks- Uncertainties associated with the project.

For an example, a constraint on timeline may force the project to speed-up and skip certain tasks. This may impact the time-frame for quality checking and force the project team to compromise on the quality of the project outcome if they have to stick to the delivery date. It is not necessary that the project constraints should always be imposed by customer/sponsors. Some of the constraints can be attributed to external environment.

2.4.3 PROJECT APPROACH

Knowing the overall approach of a project is an important part of understanding how involvement is done and how the others are made to involved, such as the project team and business stakeholders. How to choose the right approach for a project is a large topic in itself. The methodology chosen depends on many things, including the structure and location of the project team, the technologies being used on the project, and the degree to which collaboration is a part of the company's culture. Here focus is on done on two of the most common types of approach. The important thing to note is that most approaches involve the same steps:

- Plan the overall strategy, approach, and team structure.
- Define the project requirements.
- Design interaction and visual concepts and evolve them into detailed specifications.
- Develop, test, and refine the solution.
- Deploy the solution via messaging, training, and a planned launch.
- Extend the project by making recommendations for improvements.

WATERFALL APPROACH

A waterfall approach involves treating the steps of a project as separate, distinct phases, where approval of one phase is needed before the next phase begins. For example, the Design phase does not begin in earnest until requirements have been approved by business stakeholders, who sign off on one or more requirements documents at the end of the Define phase.

AGILE APPROACH

Because change is constant, project teams are continually looking for more flexible approaches than the waterfall model. Many methodologies follow a more fluid approach, with some steps happening alongside each other; for example, versions of the website could be released on a rapid, iterative schedule using an agile or rapid development. An agile approach generally has a greater focus on rapid collaboration and a reduced focus on detailed documentation and formal sign-off.

2.4.4 PROJECT COMMUNICATION

The Project required a sound communication plan, but not as the other projects having same types of communication or the same methods for distributing the information. The communication plan documented the types of information needed from the stakeholders such as when the information should be distributed, and how the information will be delivered. The types of information which will be going to communicate typically include project status, project scope statements and updates, project baseline information, risks, action items, performance measures, project acceptance, and so on. It’s important that the information needed from the stakeholders should be determined as early in the planning phase of the project management life cycle as possible so that project planning documents can be developed, and who should receive copies of them and how they should be delivered is taken in consideration.

2.4.5 PROJECT SCHEDULING

Project scheduling to a project manager like landing an airplane is to a pilot. It is often the largest and most visible aspect in project. Given below are the six steps for scheduling:

1. Plan Schedule Management.
2. Define Activities.
3. Sequence Activities.
4. Estimate Activity Resources.
5. Estimate Activity Durations.
6. Deploy Schedule.

2.5 SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)

2.5.1 DOCUMENT PURPOSE

The purpose of the Software Requirements Specification is to outline the requirements for the enhancement in the live business and the innovative ideas project as the sample for future inclusion in live business as well as organizational purpose. When live business is concerned it is basically based on the tools and the framework that the organization is been working on cloud-based Enterprise Performance Management (EPM) applications for Planning, Consolidation, Reporting, and Analytics.

2.5.2 PRODUCT

SCOPE The following key requirements covered under the project are as follows:

1. Live Project.
    - Providing a new feature.
    - Removing the time consumption.
2. Innovation Idea.
    - Using the fasttechnology.

2.6 DATA MODEL, DATA FLOW DIAGRAM & OTHER SIMILAR MODELS & DATA DICTIONARY

2.6.1 FLOW CHARTS

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a step-by-step approach to solving a task.

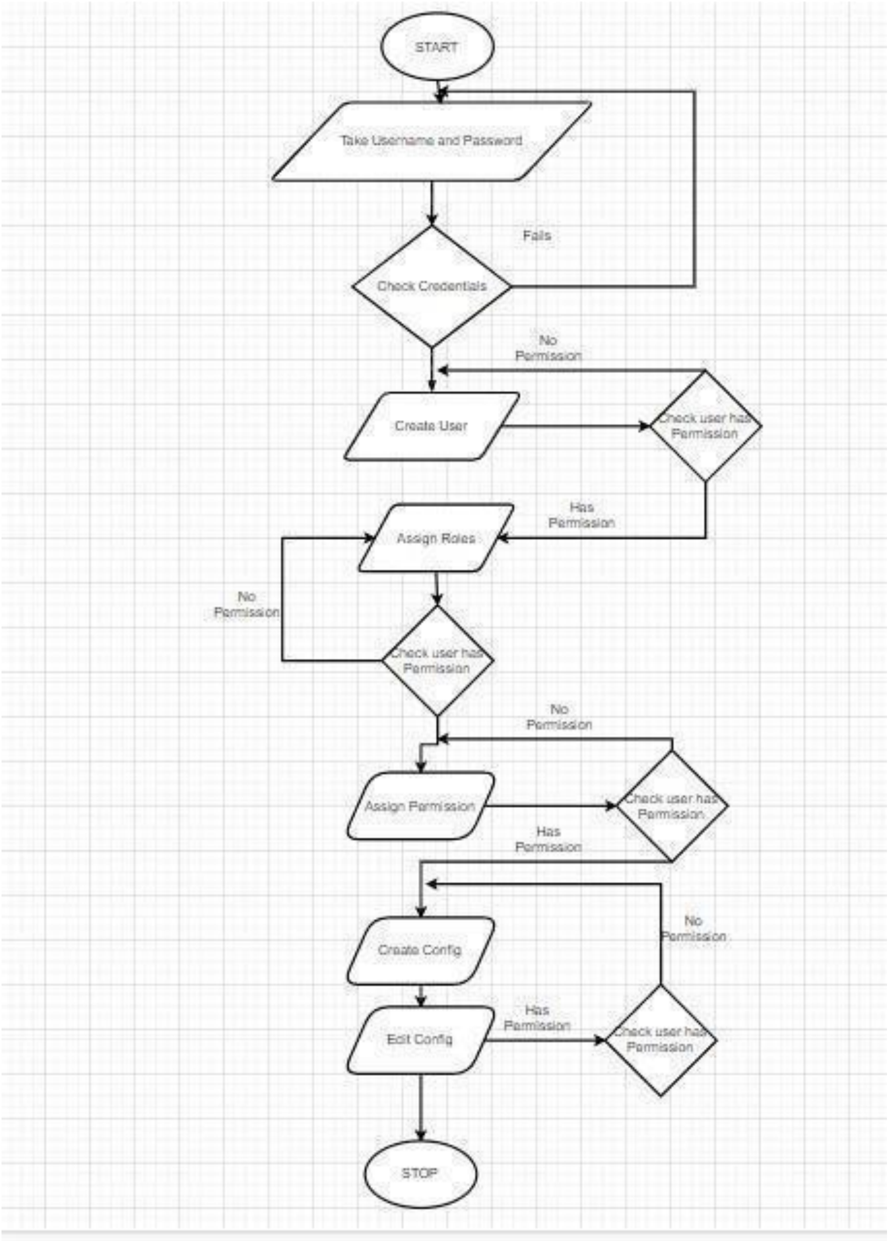


Figure 1: Flow Chart For Bench Management

2.6.2 DATA FLOW DIAGRAMS

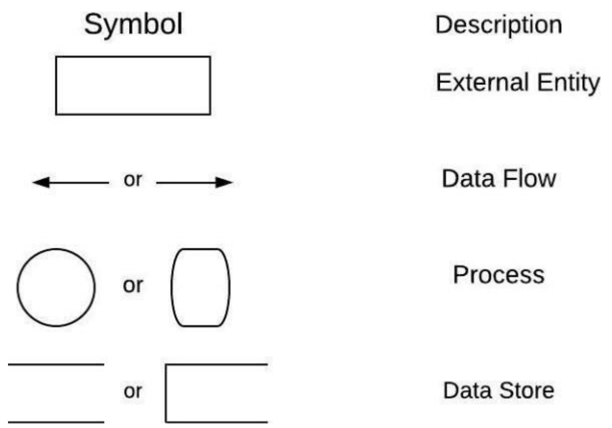
DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system’s structure charts. The **Basic Notation** used to **create a DFD’s** are as follows:

1. **Data Flow:** Data move in a specific direction from an origin to a destination.

2. **Process:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

3. **External Entity (Source or Destination):** External sources or destination of data, which may be People, programs, organizations or other entities.

4. **Data Store:** A data store represents the storage of persistent data required and/or produced by the process. Here are some examples of data stores: membership forms, database tables, etc.



TYPES OF DFD’s ARE: -

1. 0-Level DFD
2. 1-Level DFD
3. 2-Level DFD

In this project 0-Level DFD, 1-Level DFD are sufficient, as they cover all the things.

0- Level DFD or CONTEXT DIAGRAM

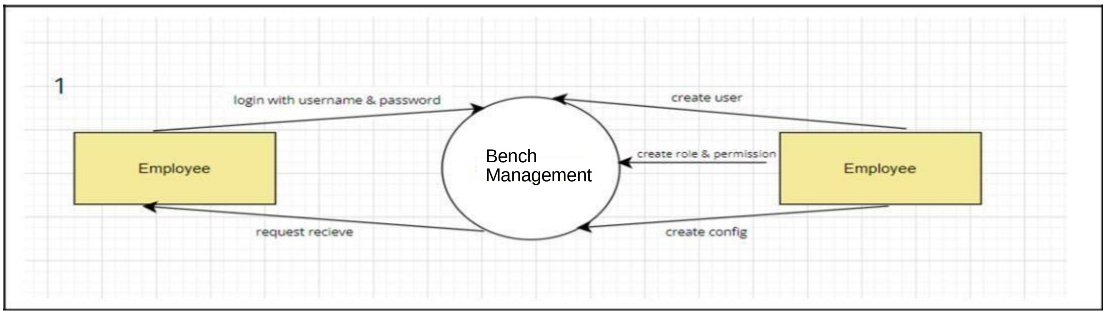


Figure 2: 0-Level DFD

1- LevelDFD

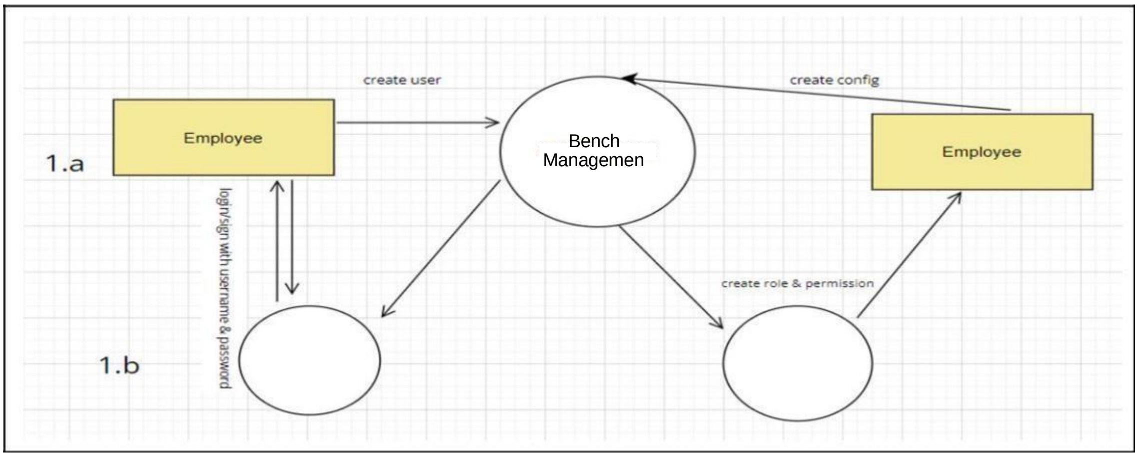


Figure 3: 1-Level DFD

2- LevelDFD

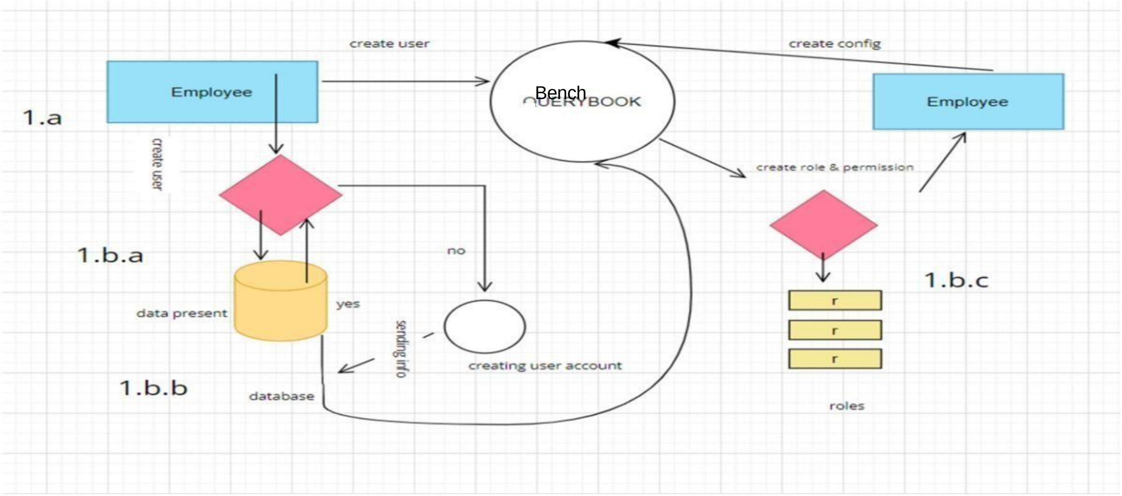


Figure 4: 2-Level DFD

2.6.3 ENTITY RELATIONSHIP DIAGRAMS (E-R DIAGRAM)

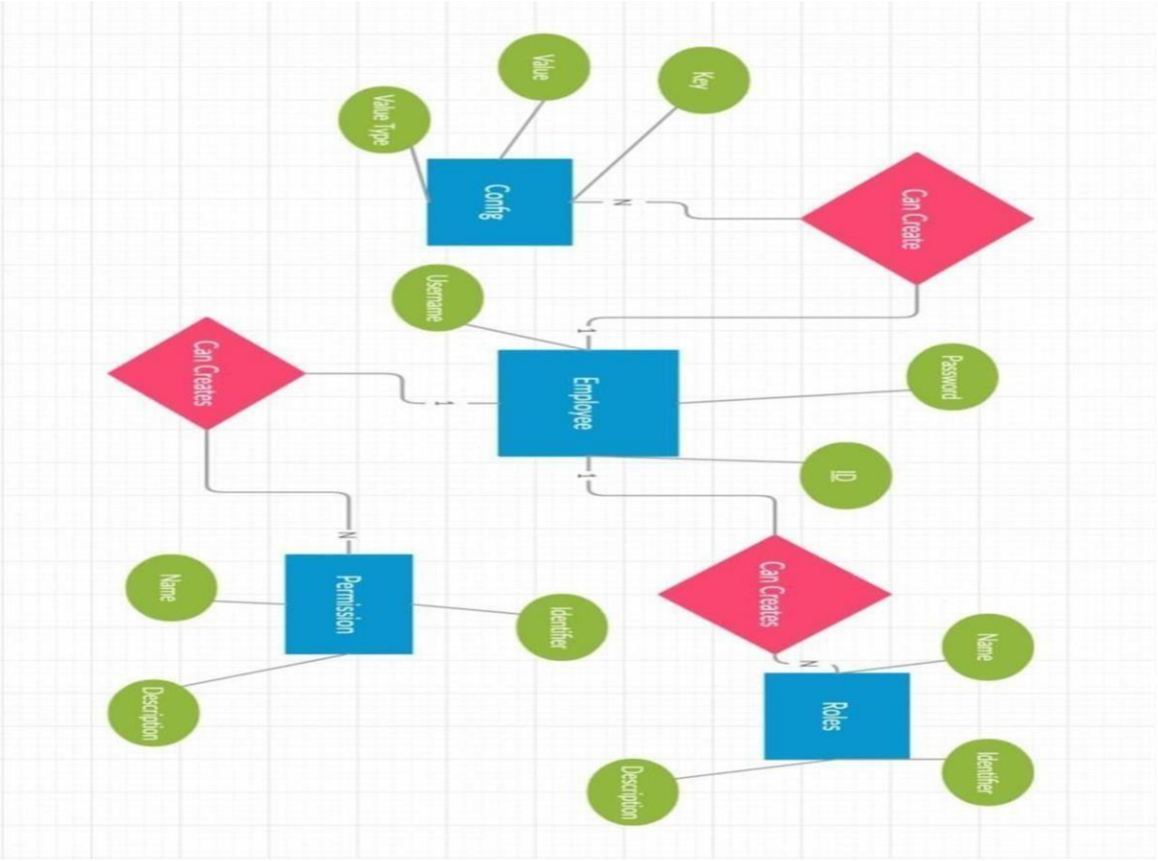


Figure 5: E-R Diagram

2.6.4 OBJECT ORI ENTED ANALYSIS (UML DIAGRAMS)

USE CASE DIAGRAM

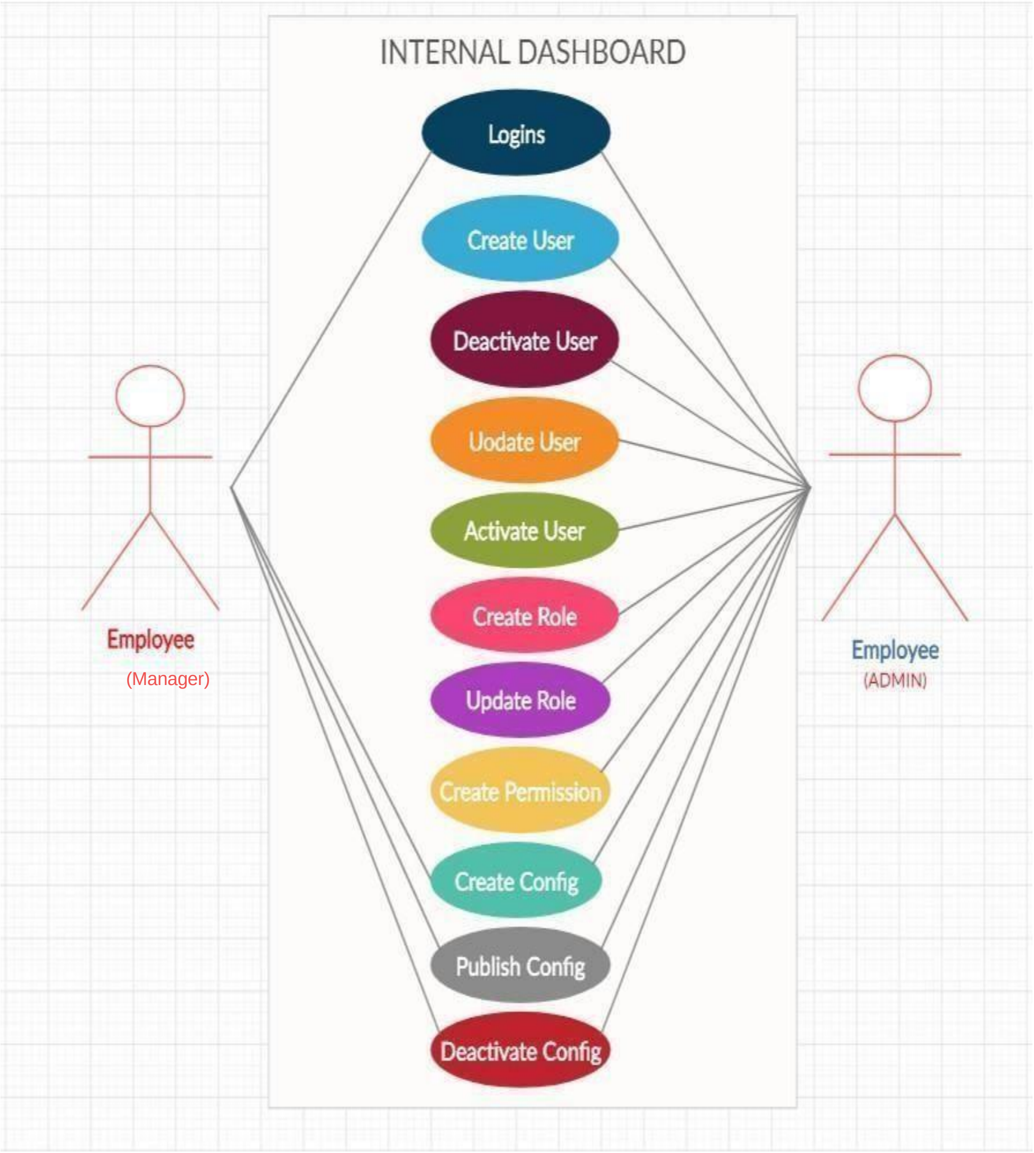


Figure 6 :Use Case Diagram

CHAPTER 3

DESIGN

3.1PROGRAM STRCUTURE

In the flexibility of uses the interface has been developer a graphics concepts in mind, in which user can interact with the system clearly system design is good then user can easily interact with the system and it will be beneficial for the system because more user will interact with system because of good design so program structure should be good of the system.

The Project has been divided into three modules as given below: -

Module 1 – Creating WEB API (Backend)

- Creating Employee management system
- Creating management system as assigning roles and permissions
- Restrictions are made through the JWT/Google authentication
- Creating Employee
- Creating Role
- Creating Permissions
- Edit Employee Details
- Assign Employee Role
- Edit Employee Roles
- Activate/Deactivate Employee
- Assign Permissions to Roles

3.2 DATA INTEGRITY & CONSTRAINTS

3.2.1 DATA INTEGRITY

Data integrity refers to the validity of data, which means data is consistent and correct. In the data warehousing field, we frequently hear the term, "Garbage In, Garbage Out." If

there is no data integrity in the data warehouse, any resulting report and analysis will not be useful. In a data warehouse or a data mart, there are three areas of where data integrity needs to be enforced.

3.2.2 DATABASE LEVEL

We can enforce data integrity at the database level. Common ways of enforcing data integrity include:

- **Referential Integrity:** The relationship between the primary key of one table and the foreign key of another table must always be maintained. For example, a primary key cannot be deleted if there is still a foreign key that refers to this primary key.
- **Primary key / Unique Constraint:** Primary key and the UNIQUE constraint are used to make sure every row in a table can be uniquely identified.
- **Not NULL:** Column defined as NOT NULL, can’t have null value.
- **Nullable:** Column which is not defined as NOT null and it has not been declared with Primary Key and Unique Constraints can contain null values.
- **Valid Values:** Only allowed values are permitted in the database. For example, if a column can only have positive integers, a value of '-1' cannot be allowed.

Data integrity refers to the validity of data, which means data is consistent and correct. In the data warehousing field, we frequently hear the term, "Garbage In, Garbage Out."
If

there is no data integrity in the data warehouse, any resulting report and analysis will not be useful. In a data warehouse or a data mart, there are three areas of where data integrity needs to be enforced.

3.2.3 TYPES OF DATA INTEGRITY

This section describes the rules that can be applied to table columns to enforce different types of data integrity.

- **Null Rule:** A null rule is a rule defined on a single column that allows or disallows inserts or updates of rows containing a null in that column.
- **Unique Column Values:** A unique value rule defined on a column (or set of columns) allows the insert or update of a row only if it contains a unique value in that column (or set of columns).
- **Primary Key Values:** A primary key value rule defined on a key (a column or set of columns) specifies that each row in the table can be uniquely identified by the values in the key.

3.2.4 REFERENTIAL INTEGRITY RULES

A referential integrity rule is a rule defined on a key (a column or set of columns) in one table that guarantees that the values in that key match the values in a key.

Referential integrity also includes the rules that dictate what types of data manipulation are allowed on referenced values and how these actions affect dependent values. The rules associated with referential integrity are:

- **Restrict:** Disallows the update or deletion of referenced data.
- **Set to Null:** When referenced data is updated or deleted, all associated dependent data is set to NULL.
- **Set to Default**: When referenced data is updated or deleted, all associated dependent data is set to a default value.
- **Cascade:** When referenced data is updated, all associated dependent data is correspondingly updated. When a referenced row is deleted, all associated dependent rows are deleted.

3.3 PROCEDURAL DESIGN

Our simple 5 step design process and project plan helps ensure your new solution:

1. Pre-Planning
2. Discovery
3. Design
4. Develop
5. Implementation and Follow-up

Applied properly, our steps of the design process can save you money (as well as headaches and re-work). And with collaboration and excellent customer support you are guaranteed to have a great experience and a solution you love.

This simple project plan has demonstrated that it applies equally well to your web site design process, database design process, or any other process you care to tackle.

1. **Pre-Planning:** Although it is not glamorous, core to the success of every project is investing time to answer key questions before you start. Your answers will provide

the purity of purpose and message that will drive where the design of the solution should go.

2. **Discovery:** Details of the project including a clear understanding of your purpose, goals and objectives, expectations, time lines, and budgets are communicated during this step of the process with resulting milestones and deliverables.
3. **Design:** Once the project is properly defined, the Design step begins. This is the prototype phase of the project. Prototype concepts are basic design ideas; the first look at your project's possibilities.
4. **Develop:** Once the design is complete, we begin to develop the physical code for your project according to your specifications. This is the nuts and bolts of the project life cycle. Depending upon the nature of your project, common tasks that are undertaken during this phase include coding, environment testing, stress testing, backup and recovery procedures, and user manuals.
5. **Implementation Launch and Follow-Up:** Once the development step is complete, your solution is ready to deploy. Upon final approval and invoicing, we deliver the project files and make sure everything works as promised.

3.4 UI OR INPUT AND OUTPUT

➔ The UI for signup & login –

- ### 3.4.1 Landing Page

-Login page

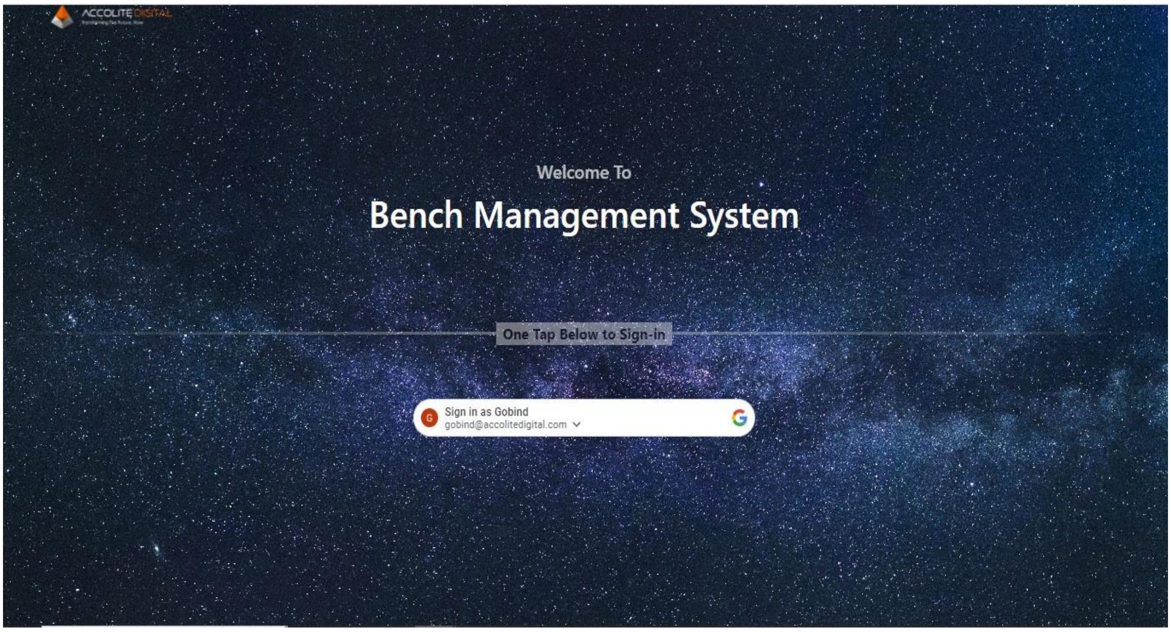
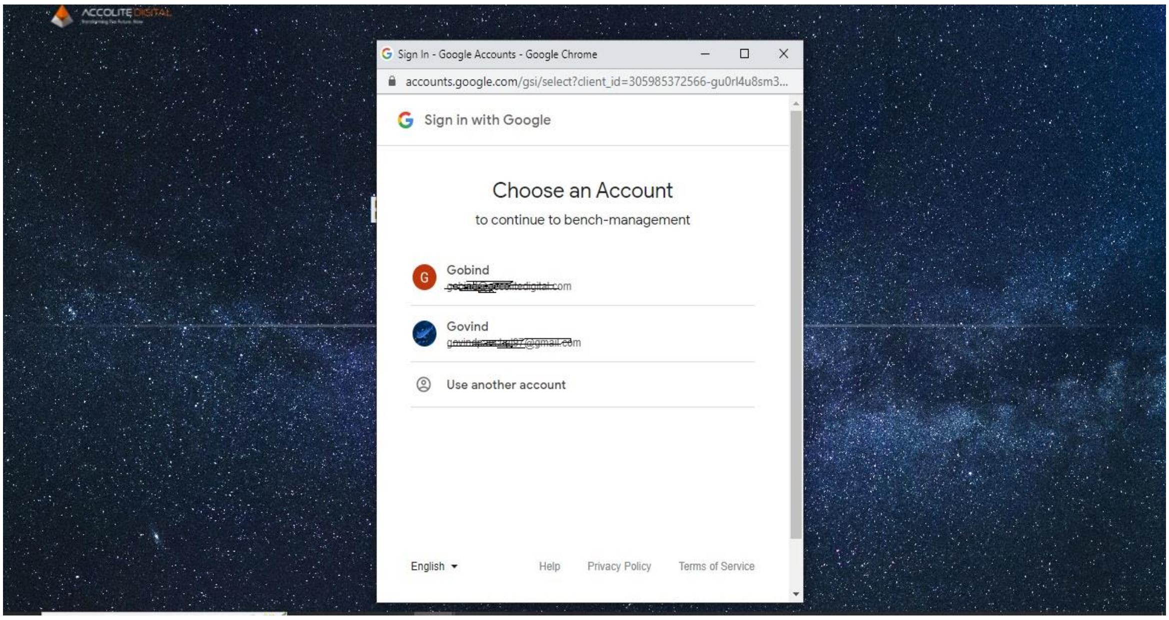


Figure 7:Login Page

-Google Authentication


Sign In - Google Accounts - Google Chrome

accounts.google.com/gsi/select?client_id=305985372566-gu0rl4u8sm3...

Sign in with Google

Choose an Account

to continue to bench-management

Gobind
[redacted]

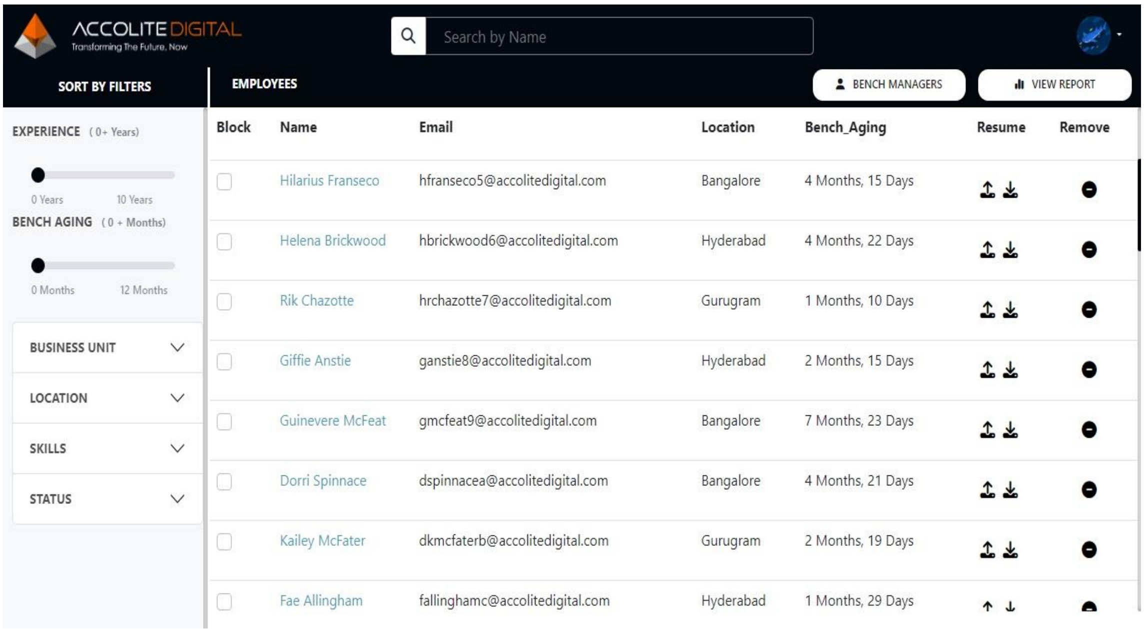
Govind
[redacted]

Use another account

Figure 8:Google Authentication

- ## Admin View

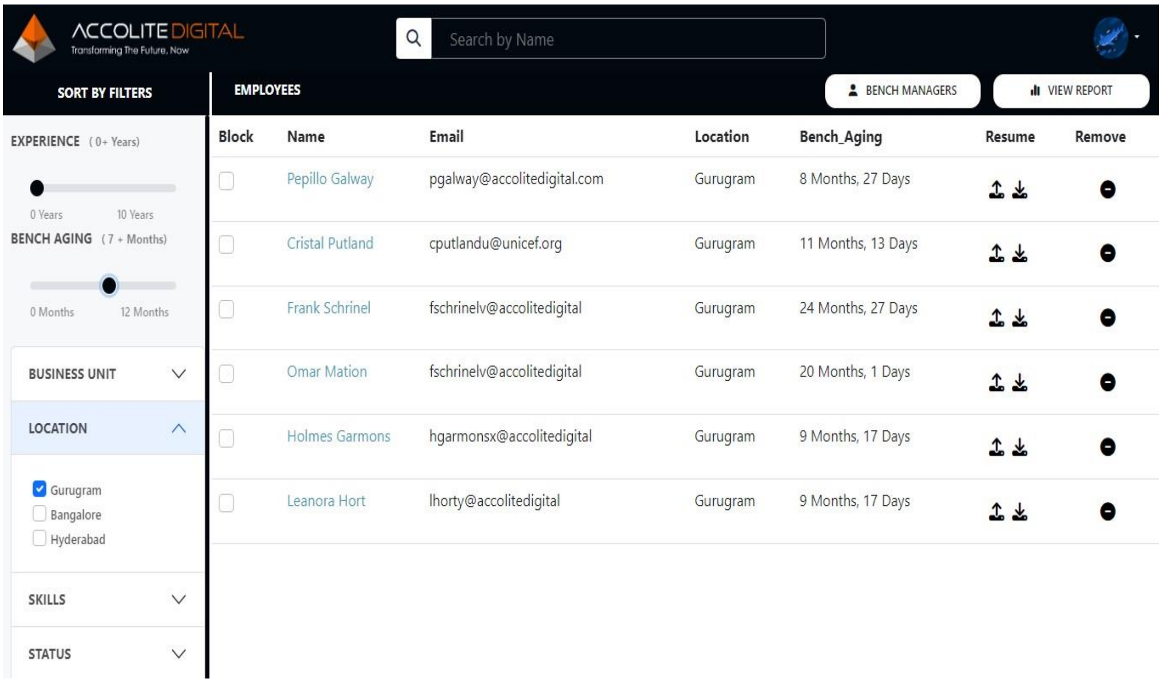
-By Default View



Block	Name	Email	Location	Bench_Aging	Resume	Remove
□	Hilarius Franseco	hfranseco5@accolitedigital.com	Bangalore	4 Months, 15 Days	↥ ⤓	●
□	Helena Brickwood	hbrickwood6@accolitedigital.com	Hyderabad	4 Months, 22 Days	↥ ⤓	●
□	Rik Chazotte	hrchazotte7@accolitedigital.com	Gurugram	1 Months, 10 Days	↥ ⤓	●
□	Giffie Anstie	ganstie8@accolitedigital.com	Hyderabad	2 Months, 15 Days	↥ ⤓	●
□	Guinevere McFeat	gmcfeat9@accolitedigital.com	Bangalore	7 Months, 23 Days	↥ ⤓	●
□	Dorri Spinnace	dspinnacea@accolitedigital.com	Bangalore	4 Months, 21 Days	↥ ⤓	●
□	Kailey McFater	dkmcfaterb@accolitedigital.com	Gurugram	2 Months, 19 Days	↥ ⤓	●
□	Fae Allingham	fallinghamc@accolitedigital.com	Hyderabad	1 Months, 29 Days	↥ ⤓	●

Figure 9:Admin By Default View

-Admin View With Applied Filters

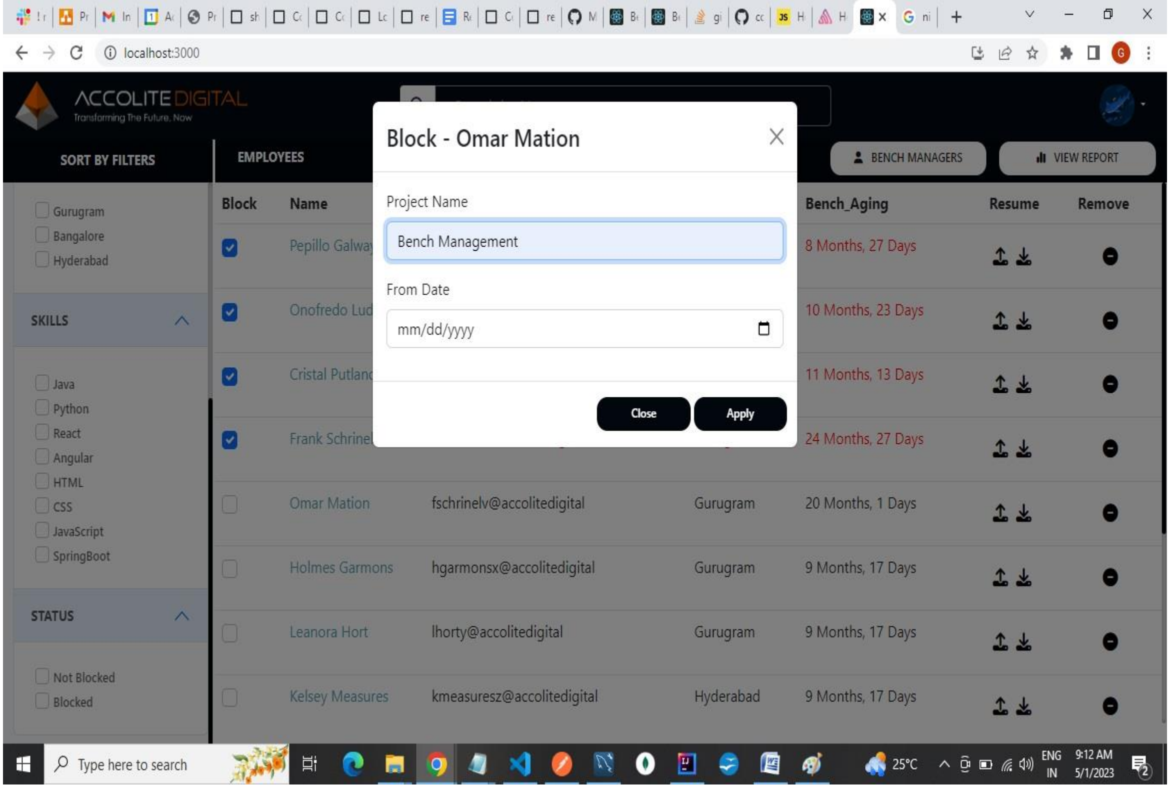


☑ Gurugram
□ Bangalore
□ Hyderabad

Block	Name	Email	Location	Bench_Aging	Resume	Remove
□	Pepillo Galway	pgalway@accolitedigital.com	Gurugram	8 Months, 27 Days	↥ ⤓	●
□	Cristal Putland	cputlandu@unicef.org	Gurugram	11 Months, 13 Days	↥ ⤓	●
□	Frank Schrinel	fschrinelv@accolitedigital	Gurugram	24 Months, 27 Days	↥ ⤓	●
□	Omar Mation	fschrinelv@accolitedigital	Gurugram	20 Months, 1 Days	↥ ⤓	●
□	Holmes Garmons	hgarmonsx@accolitedigital	Gurugram	9 Months, 17 Days	↥ ⤓	●
□	Leanora Hort	lhorty@accolitedigital	Gurugram	9 Months, 17 Days	↥ ⤓	●

Figure 10: Admin View With Applied Filters

-Admin View with Blocked Status



Block - Omar Mation

Project Name: Bench Management

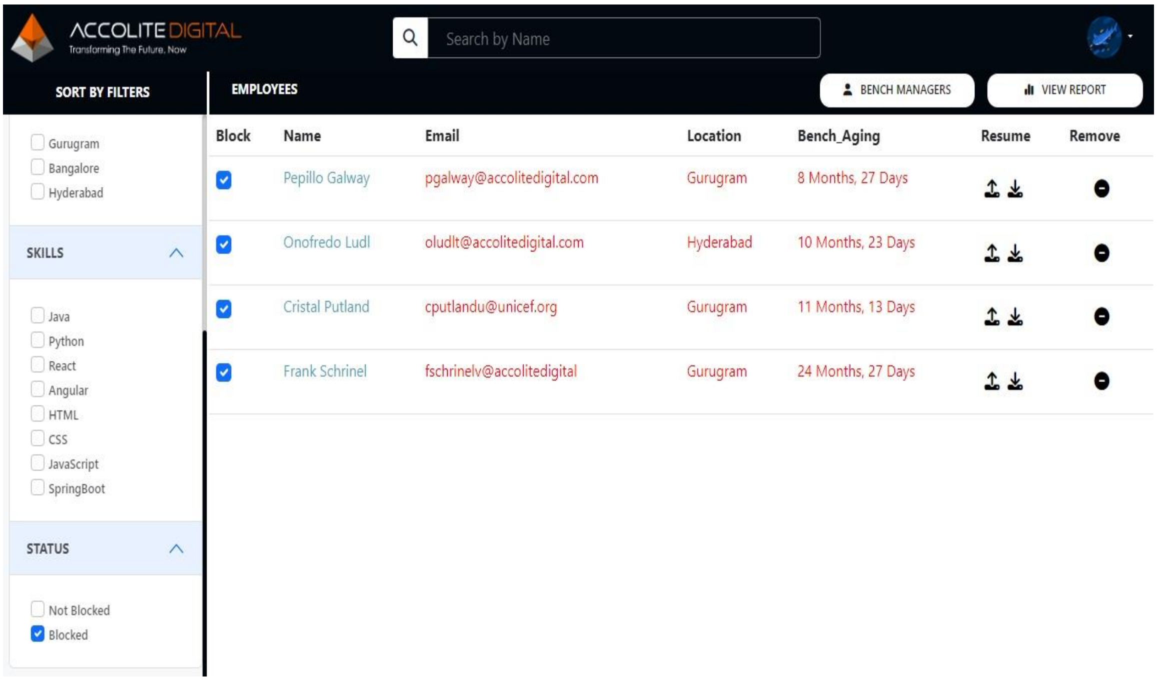
From Date: mm/dd/yyyy

Close Apply

Block	Name			Bench_Aging	Resume	Remove
☑	Pepillo Galwa			8 Months, 27 Days	↥ ↧	●
☑	Onofredo Lud			10 Months, 23 Days	↥ ↧	●
☑	Cristal Putlan			11 Months, 13 Days	↥ ↧	●
☑	Frank Schrine			24 Months, 27 Days	↥ ↧	●
□	Omar Mation	fschrinelv@accolitedigital	Gurugram	20 Months, 1 Days	↥ ↧	●
□	Holmes Garmons	hgarmonsx@accolitedigital	Gurugram	9 Months, 17 Days	↥ ↧	●
□	Leanora Hort	lhorty@accolitedigital	Gurugram	9 Months, 17 Days	↥ ↧	●
□	Kelsey Measures	kmeasuresz@accolitedigital	Hyderabad	9 Months, 17 Days	↥ ↧	●

Figure 11: Admin View with Blocked Status

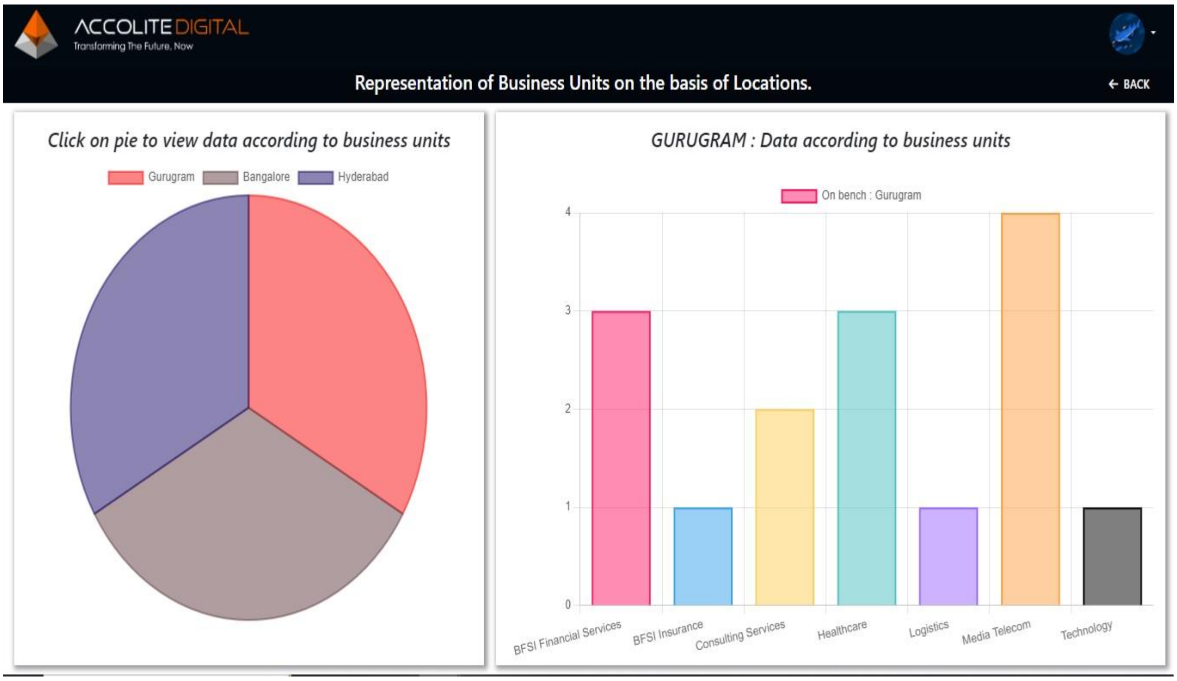
-Admin View with Blocked Employees



Block	Name	Email	Location	Bench_Aging	Resume	Remove
☑	Pepillo Galway	pgalway@accolitedigital.com	Gurugram	8 Months, 27 Days	↥ ↧	●
☑	Onofredo Ludl	oludlt@accolitedigital.com	Hyderabad	10 Months, 23 Days	↥ ↧	●
☑	Cristal Putland	cputlandu@unicef.org	Gurugram	11 Months, 13 Days	↥ ↧	●
☑	Frank Schrinel	fschrinelv@accolitedigital	Gurugram	24 Months, 27 Days	↥ ↧	●

Figure 12: Admin View with Blocked Employees

-Admin Representation of Business Unit Based On Location

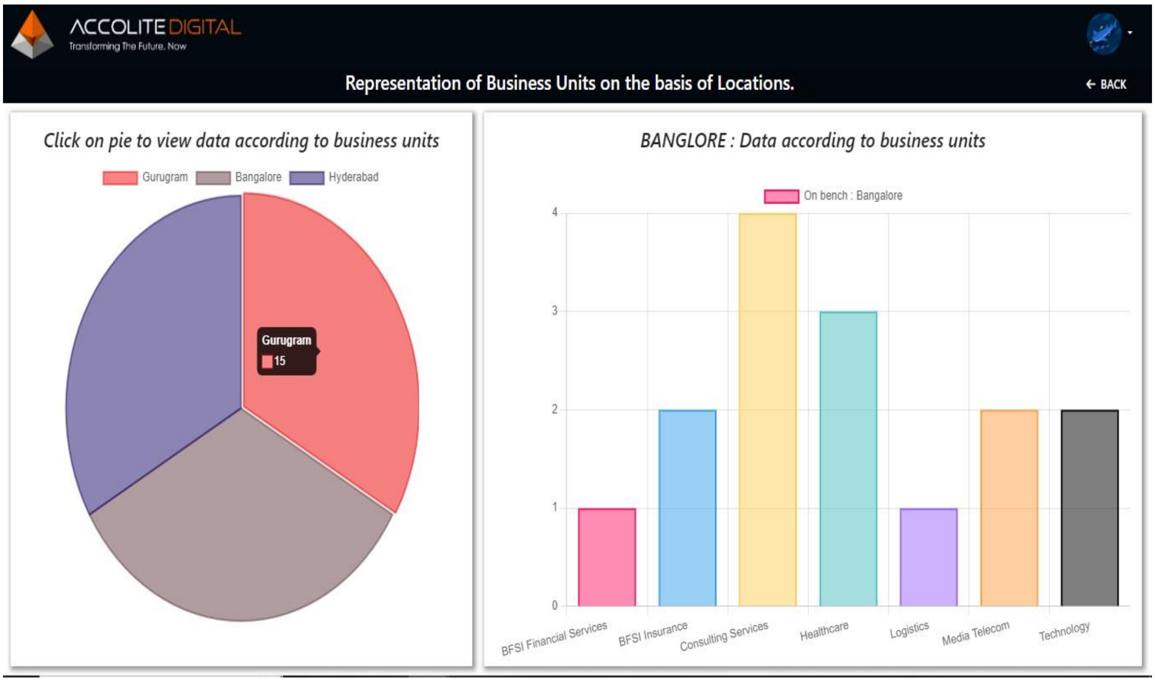


Representation of Business Units on the basis of Locations.

Click on pie to view data according to business units

GURUGRAM : Data according to business units

Figure 13: Business Unit Based On Location



Representation of Business Units on the basis of Locations.

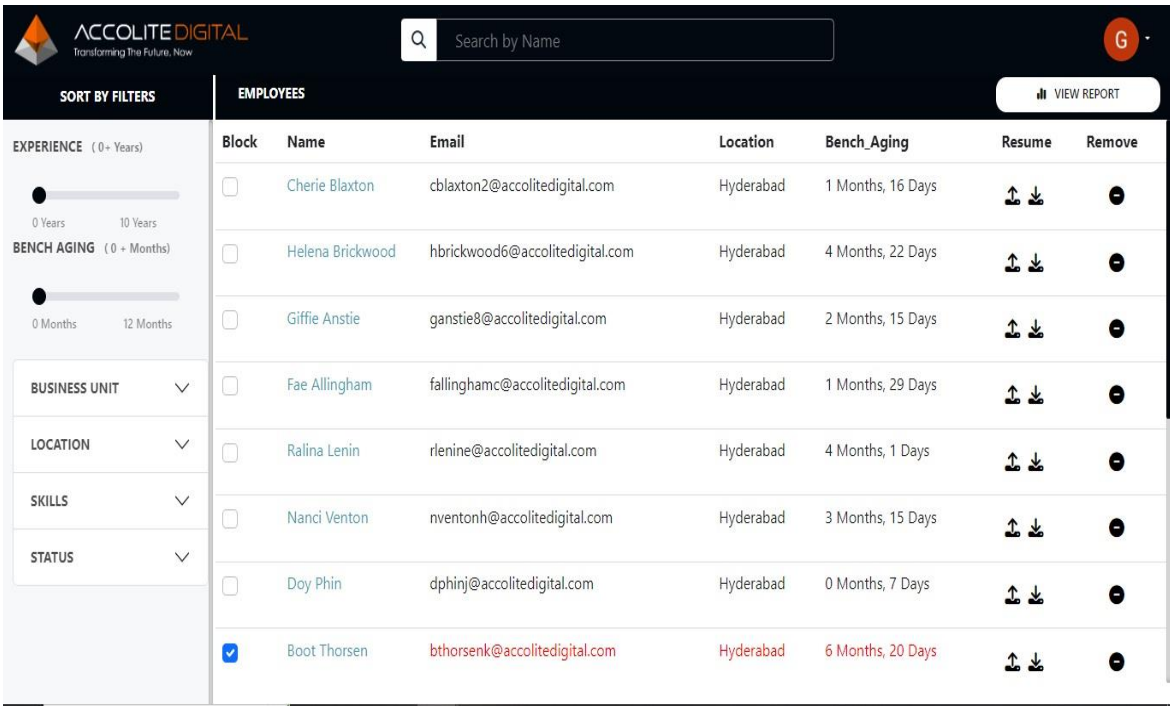
Click on pie to view data according to business units

BANGLORE : Data according to business units

Figure 14

- Manager View

-Manager By Default View



EXPERIENCE (0+ Years)

0 Years 10 Years

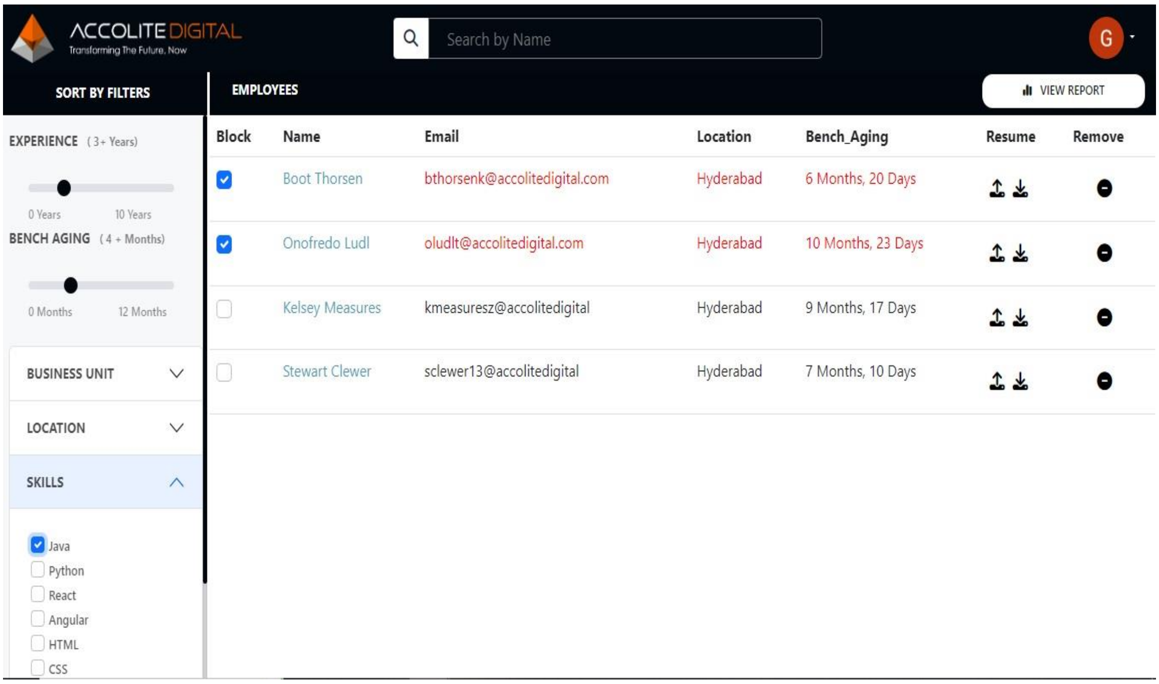
BENCH AGING (0 + Months)

0 Months 12 Months

Block	Name	Email	Location	Bench_Aging	Resume	Remove
□	Cherie Blaxton	cblaxton2@accolitedigital.com	Hyderabad	1 Months, 16 Days	↑ ↓	⊖
□	Helena Brickwood	hbrickwood6@accolitedigital.com	Hyderabad	4 Months, 22 Days	↑ ↓	⊖
□	Giffie Anstie	ganstie8@accolitedigital.com	Hyderabad	2 Months, 15 Days	↑ ↓	⊖
□	Fae Allingham	fallinghamc@accolitedigital.com	Hyderabad	1 Months, 29 Days	↑ ↓	⊖
□	Ralina Lenin	rlenine@accolitedigital.com	Hyderabad	4 Months, 1 Days	↑ ↓	⊖
□	Nanci Venton	nventonh@accolitedigital.com	Hyderabad	3 Months, 15 Days	↑ ↓	⊖
□	Doy Phin	dphinj@accolitedigital.com	Hyderabad	0 Months, 7 Days	↑ ↓	⊖
☑	Boot Thorsen	bthorsenk@accolitedigital.com	Hyderabad	6 Months, 20 Days	↑ ↓	⊖

Figure 15: Manager By Default View

-Manager View With Applied Filters



EXPERIENCE (3+ Years)

0 Years 10 Years

BENCH AGING (4 + Months)

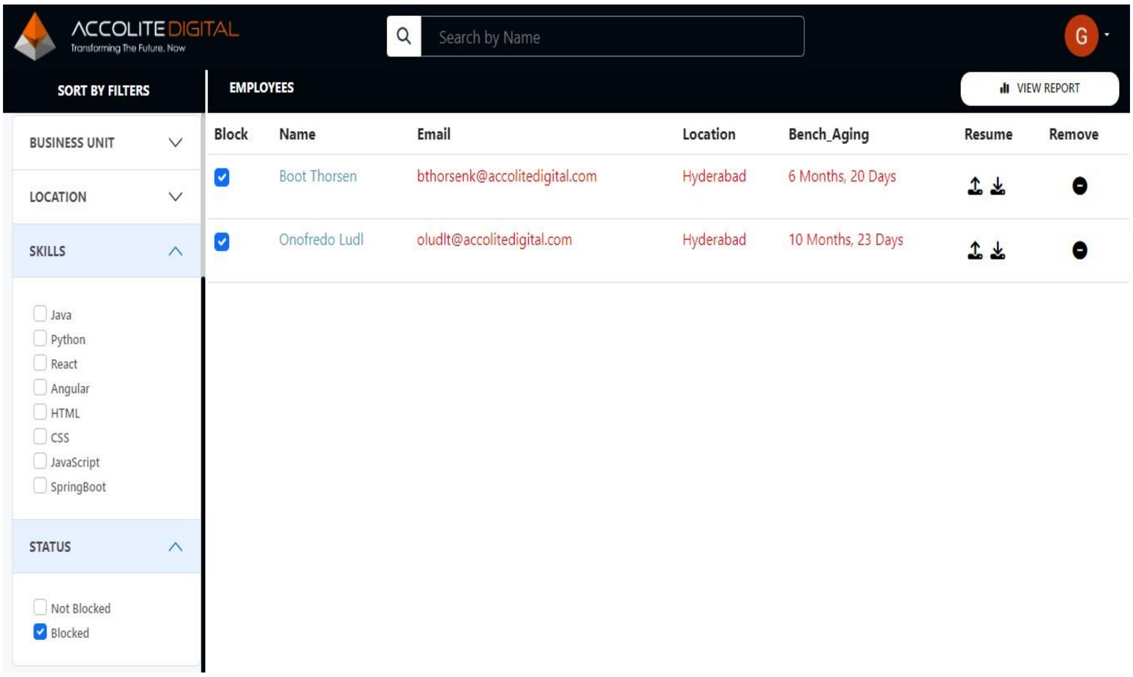
0 Months 12 Months

☑ Java
□ Python
□ React
□ Angular
□ HTML
□ CSS

Block	Name	Email	Location	Bench_Aging	Resume	Remove
☑	Boot Thorsen	bthorsenk@accolitedigital.com	Hyderabad	6 Months, 20 Days	↑ ↓	⊖
☑	Onofredo Ludl	oludlt@accolitedigital.com	Hyderabad	10 Months, 23 Days	↑ ↓	⊖
□	Kelsey Measures	kmeasuresz@accolitedigital	Hyderabad	9 Months, 17 Days	↑ ↓	⊖
□	Stewart Clewer	sclewer13@accolitedigital	Hyderabad	7 Months, 10 Days	↑ ↓	⊖

Figure 16: Manager View With Applied Filters

-Manager View With Blocked Status



BUSINESS UNIT

LOCATION

SKILLS

☐ Java
☐ Python
☐ React
☐ Angular
☐ HTML
☐ CSS
☐ JavaScript
☐ SpringBoot

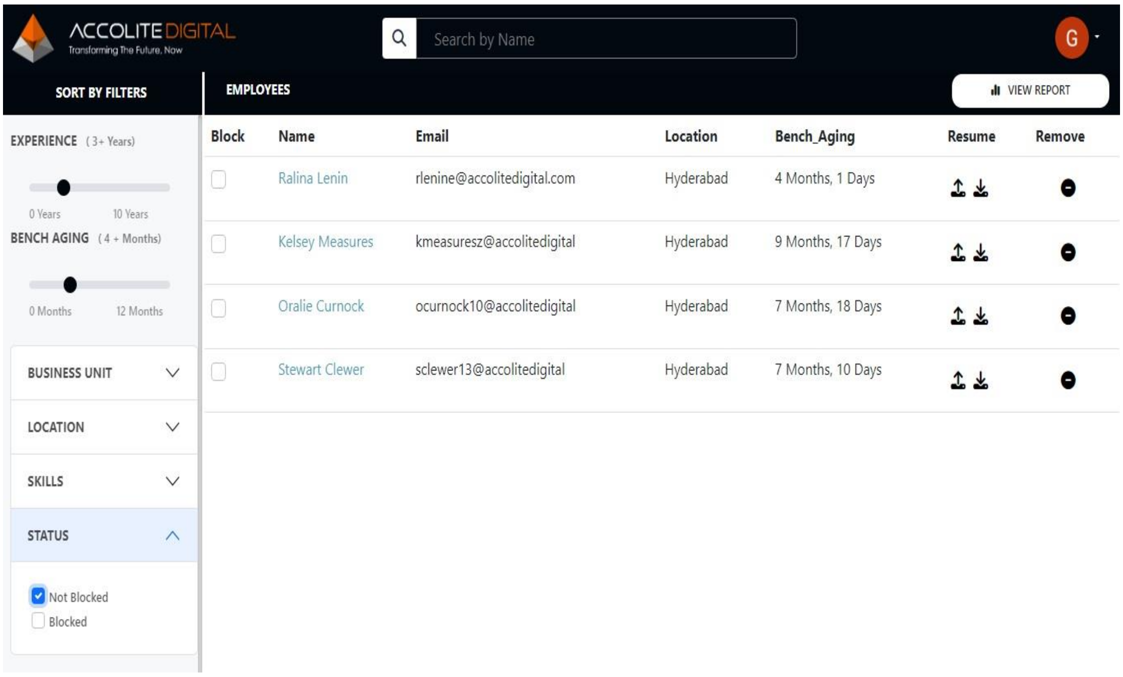
STATUS

☐ Not Blocked
☑ Blocked

Block	Name	Email	Location	Bench_Aging	Resume	Remove
☑	Boot Thorsen	bthorsenk@accolitedigital.com	Hyderabad	6 Months, 20 Days	↥ ↧	⊖
☑	Onofredo Ludl	oludlt@accolitedigital.com	Hyderabad	10 Months, 23 Days	↥ ↧	⊖

Figure 17: Manager View With Blocked Status

-Manager View With Not Blocked Status



EXPERIENCE (3+ Years)

0 Years 10 Years

BENCH AGING (4 + Months)

0 Months 12 Months

BUSINESS UNIT

LOCATION

SKILLS

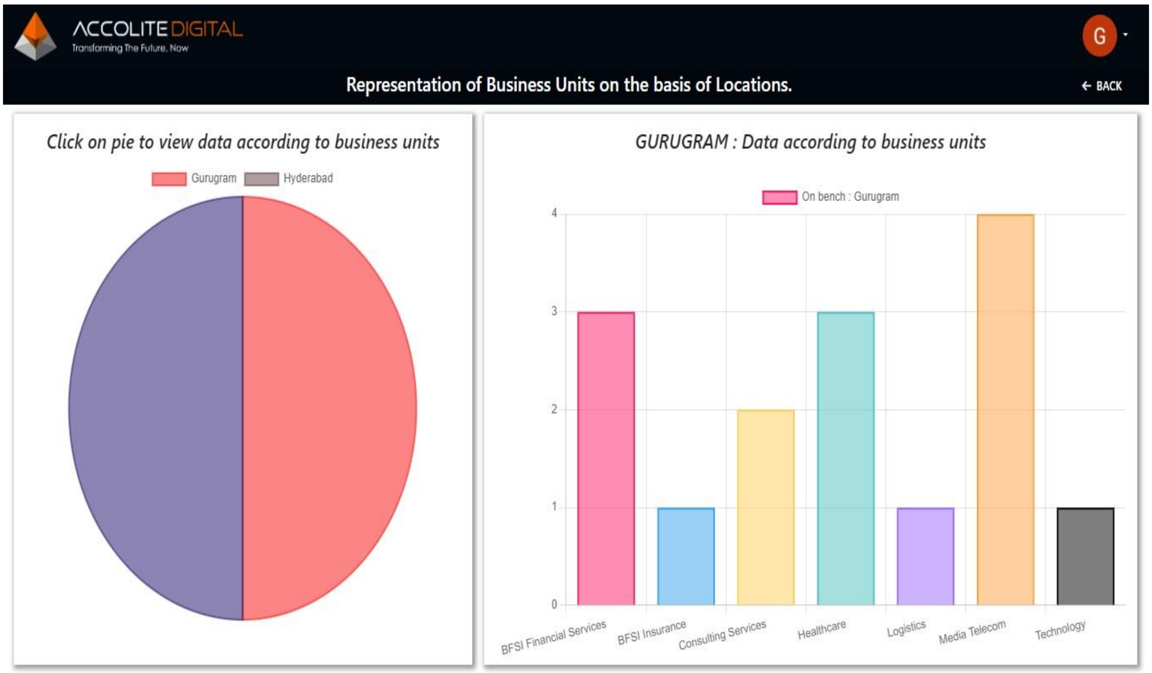
STATUS

☑ Not Blocked
☐ Blocked

Block	Name	Email	Location	Bench_Aging	Resume	Remove
☐	Ralina Lenin	rlenine@accolitedigital.com	Hyderabad	4 Months, 1 Days	↥ ↧	⊖
☐	Kelsey Measures	kmeasuresz@accolitedigital	Hyderabad	9 Months, 17 Days	↥ ↧	⊖
☐	Oralie Curnock	ocurnock10@accolitedigital	Hyderabad	7 Months, 18 Days	↥ ↧	⊖
☐	Stewart Clewer	sclewer13@accolitedigital	Hyderabad	7 Months, 10 Days	↥ ↧	⊖

Figure 18: Manager View With Not Blocked Status

-Manager Representation of Business Unit Based On Location

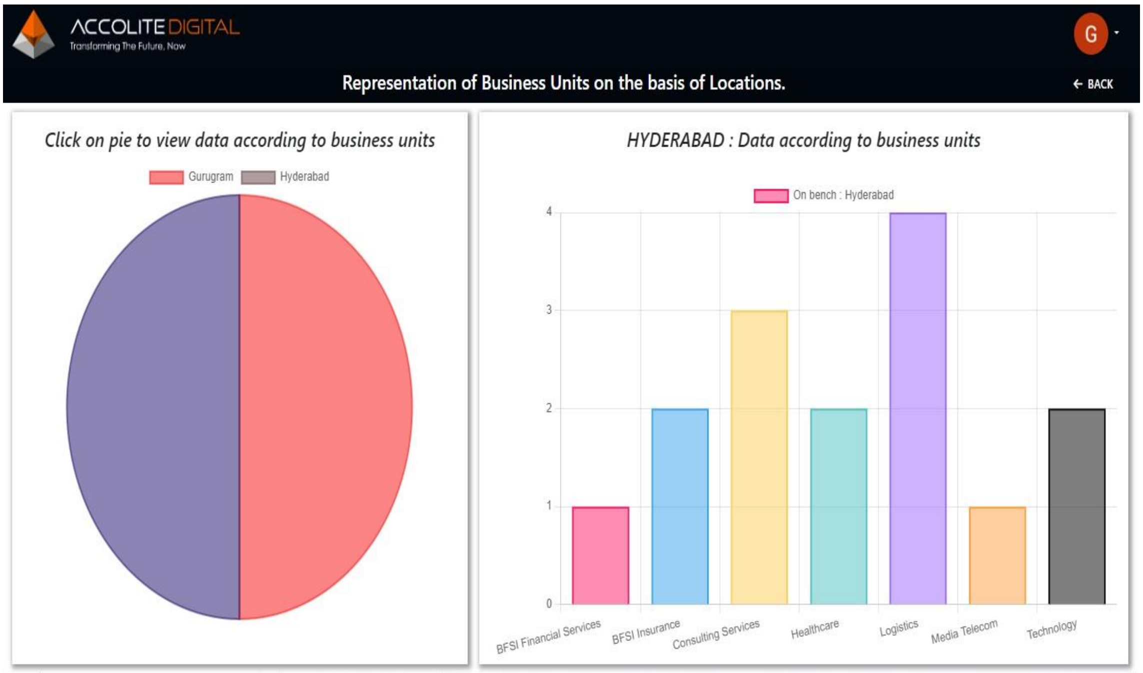


Representation of Business Units on the basis of Locations.

Click on pie to view data according to business units

GURUGRAM : Data according to business units

Figure 19: Business Unit Based On Location



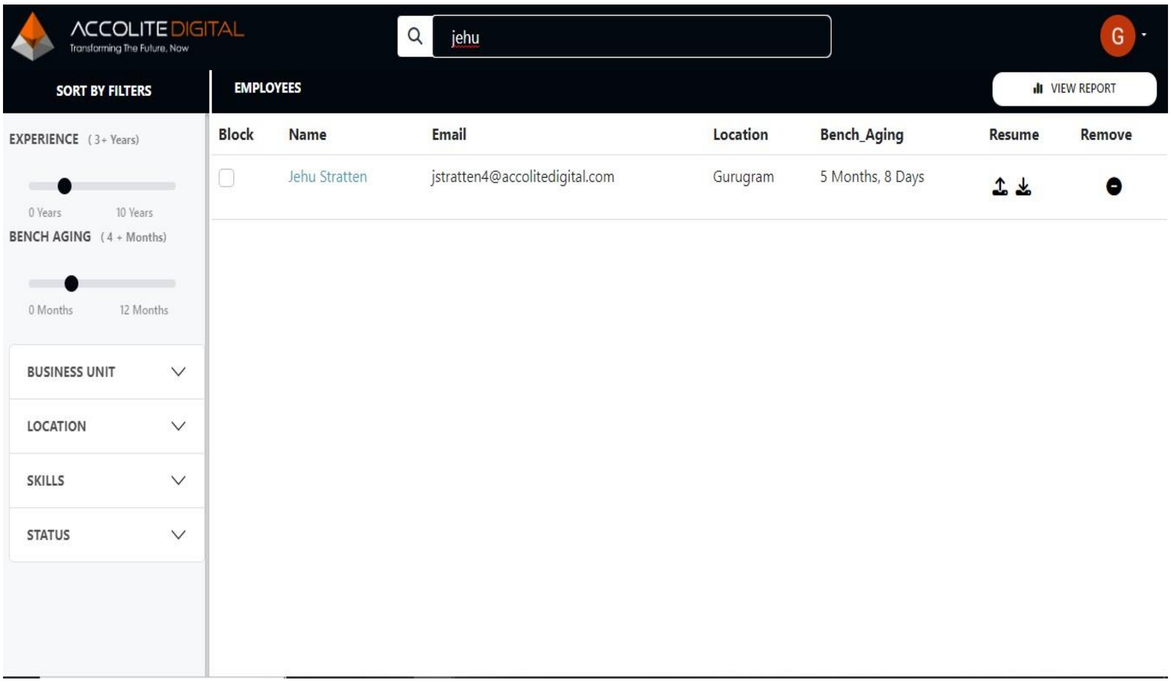
Representation of Business Units on the basis of Locations.

Click on pie to view data according to business units

HYDERABAD : Data according to business units

Figure 20

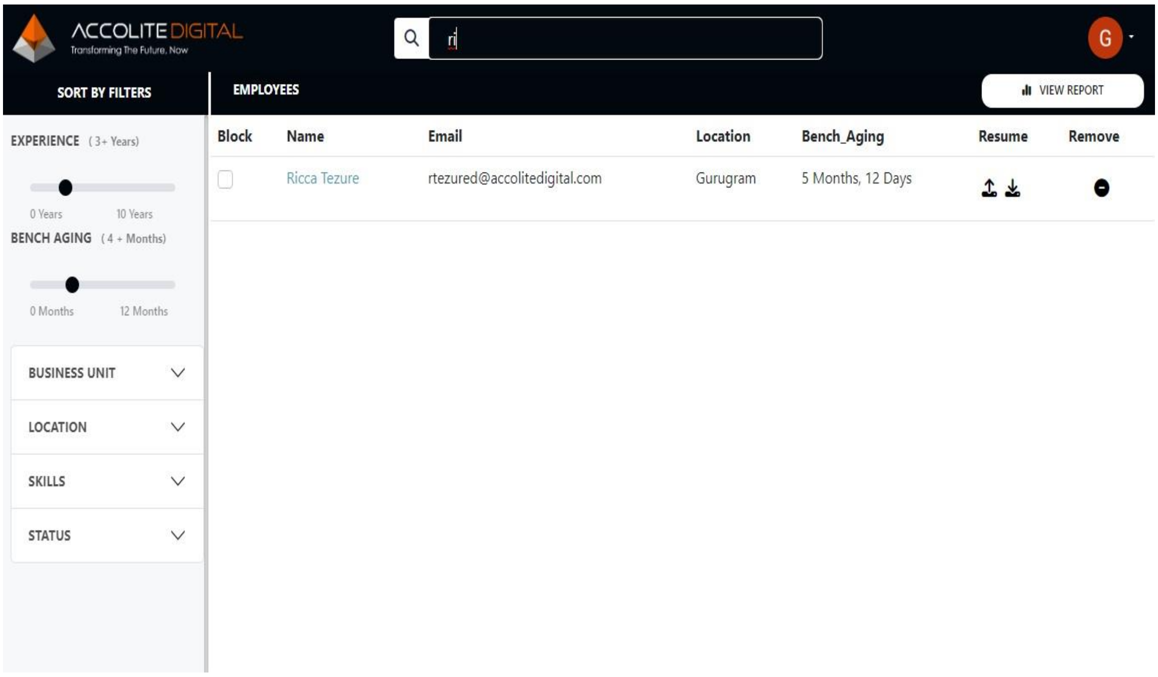
- Manager Search Employees By Names



jehu

Block	Name	Email	Location	Bench_Aging	Resume	Remove
□	Jehu Stratten	jstratten4@accolitedigital.com	Gurugram	5 Months, 8 Days	⭱ ⭳	●

Figure 21: Manager Search Employees By Names

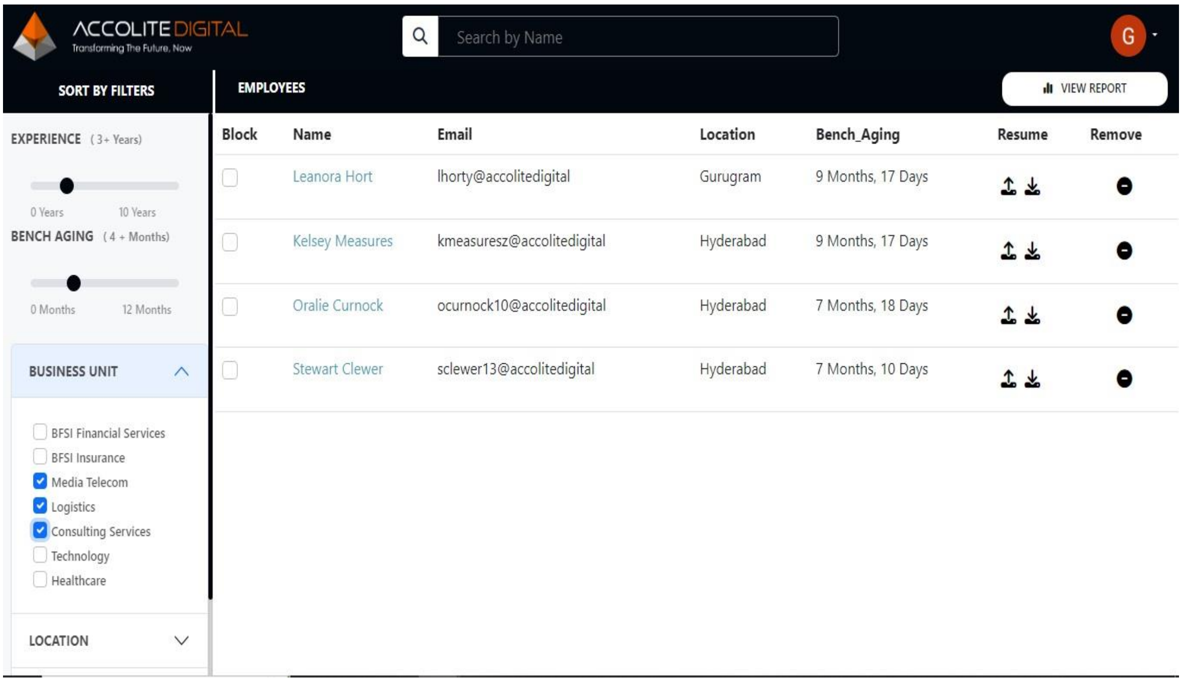


ri

Block	Name	Email	Location	Bench_Aging	Resume	Remove
□	Ricca Tezure	rtezured@accolitedigital.com	Gurugram	5 Months, 12 Days	⭱ ⭳	●

Figure 22

-Manager View with Business Units



ACCOLITE DIGITAL
Transforming The Future, Now

Search by Name

G

SORT BY FILTERS

EMPLOYEES

VIEW REPORT

EXPERIENCE (3+ Years)

0 Years 10 Years

BENCH AGING (4 + Months)

0 Months 12 Months

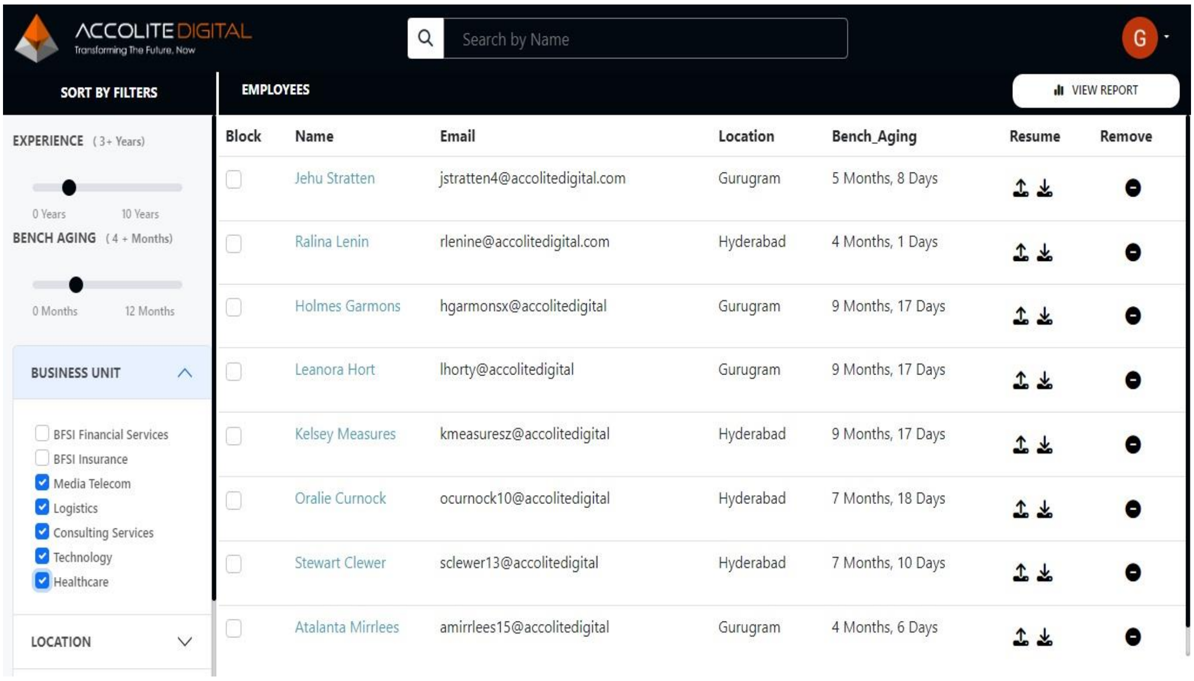
BUSINESS UNIT

- □ BFSI Financial Services
- □ BFSI Insurance
- ☑ Media Telecom
- ☑ Logistics
- ☑ Consulting Services
- □ Technology
- □ Healthcare

LOCATION

Block	Name	Email	Location	Bench_Aging	Resume	Remove
□	Leanora Hort	lhorty@accolitedigital	Gurugram	9 Months, 17 Days	⇧ ⇩	●
□	Kelsey Measures	kmeasuresz@accolitedigital	Hyderabad	9 Months, 17 Days	⇧ ⇩	●
□	Oralie Curnock	ocurnock10@accolitedigital	Hyderabad	7 Months, 18 Days	⇧ ⇩	●
□	Stewart Clewer	sclewer13@accolitedigital	Hyderabad	7 Months, 10 Days	⇧ ⇩	●

Figure 23: Manager View with Business Units



ACCOLITE DIGITAL
Transforming The Future, Now

Search by Name

G

SORT BY FILTERS

EMPLOYEES

VIEW REPORT

EXPERIENCE (3+ Years)

0 Years 10 Years

BENCH AGING (4 + Months)

0 Months 12 Months

BUSINESS UNIT

- □ BFSI Financial Services
- □ BFSI Insurance
- ☑ Media Telecom
- ☑ Logistics
- ☑ Consulting Services
- ☑ Technology
- ☑ Healthcare

LOCATION

Block	Name	Email	Location	Bench_Aging	Resume	Remove
□	Jehu Stratten	jstratten4@accolitedigital.com	Gurugram	5 Months, 8 Days	⇧ ⇩	●
□	Ralina Lenin	rlenine@accolitedigital.com	Hyderabad	4 Months, 1 Days	⇧ ⇩	●
□	Holmes Garmons	hgarmonsx@accolitedigital	Gurugram	9 Months, 17 Days	⇧ ⇩	●
□	Leanora Hort	lhorty@accolitedigital	Gurugram	9 Months, 17 Days	⇧ ⇩	●
□	Kelsey Measures	kmeasuresz@accolitedigital	Hyderabad	9 Months, 17 Days	⇧ ⇩	●
□	Oralie Curnock	ocurnock10@accolitedigital	Hyderabad	7 Months, 18 Days	⇧ ⇩	●
□	Stewart Clewer	sclewer13@accolitedigital	Hyderabad	7 Months, 10 Days	⇧ ⇩	●
□	Atalanta Mirrlees	amirrlees15@accolitedigital	Gurugram	4 Months, 6 Days	⇧ ⇩	●

Figure 24: Manager View with Business Units

CHAPTER 4

CODING

4.1 COMMENTS AND DESCRIPTION

Classes names and Variables names are meaning full, so anyone can easily understand this. The function names will start with small letters.

- ✓ Comments should add to the clarity of code.
- ✓ Avoid decoration, i.e., do not use banner-like comments
- ✓ Document why something is being done, not just what**.**

4.2 STANDARDIZATION OF CODING

The purpose of code is to facilitate the identification, retrieval of the items and information. A code is an oriented collection of symbols design to provide unique identification of an entry or attribute. Code is built with manually exclusive features. Codes in all cases specify object which are physical or on performance characteristics. They are used to give optimal distraction and other information. Codes are used for identifying, accessing, storing and matching records. The codes insure that only one value of the code with a single meaning is correctly applied to give entity or attribute as described in various ways. Code can also be design in a manner easily understood and applied by the user. This coding standard includes naming conversions of variables, constants and objects, standardized formats or labelling and commenting code, spacing, formatting and indenting.

Standardization has a positive impact on any business. In the Software Industry too, there are certain coding standards that are needed for successful **Software Development**. For most of the organizations, the seamless functioning of software programs is essential for their growth.

It is seen that good quality software and code is not as easy as pie. It requires consistent efforts and sheers the focus of the software development team to meet quality goals. This is an essential thing for the software project. The developers are often seen side-stepping the quality standards when they are required to complete their tasks in a short span of time.

4.2.1 CODING STANDARDS

They are a series of procedures that can be defined for a particular programming language specifying a programming style, the methods, & different procedures. These procedures can be for various aspects of the program written in that language. They can be considered as essential attributes of software development.

A coding standard makes sure that all the developers working on the project are following certain specified guidelines. The code can be easily understood and proper consistency is maintained.

Consistency has a positive impact on the quality of the program and one should maintain it while coding. Also, it should be taken care that the guidelines are homogeneously followed across different levels of the system and they do not contradict each other. The finished program code should look like that it has been written by a single developer, in a single session.

Why coding standards are important in Software Development?

If the coding standards are not defined, developers could be using any of their own methods, which might lead to certain negative effects such as:

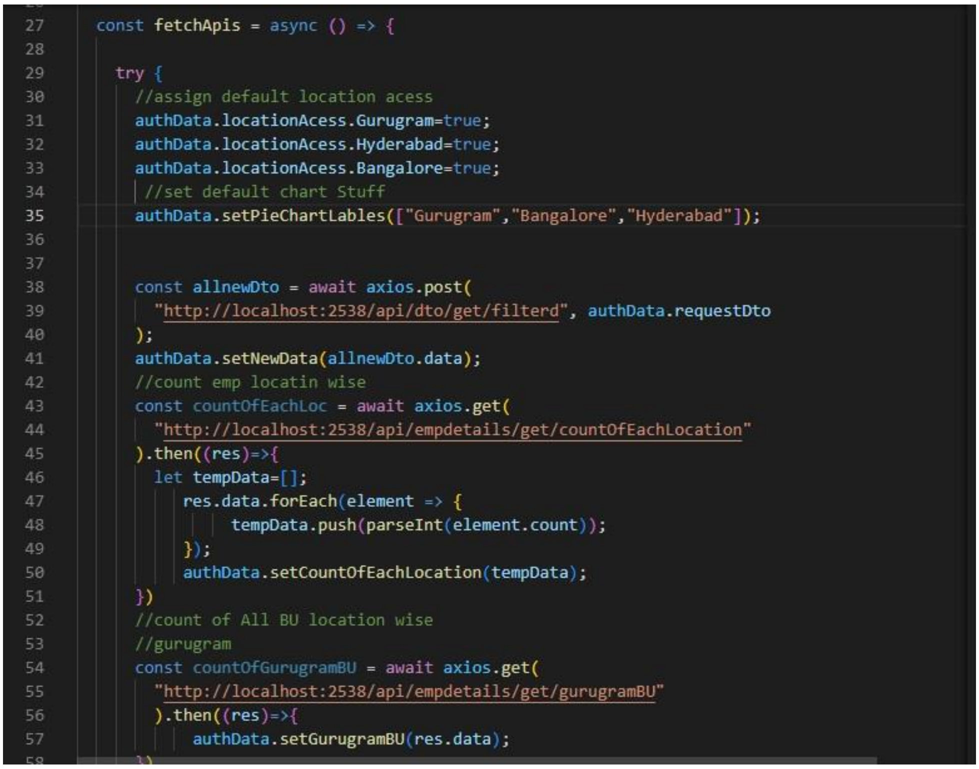
1. **Security Concerns**: Software becomes vulnerable to attacks if it is inconsistent, contains bugs and errors in logic. Most of the aforementioned problems arise due to the faulty programming code that might have resulted from poor coding practices.
2. **Performance Issues**: Poor coding has an adverse effect on the performance of the site. The performance issues comprise a multitude of things like when the user is interacting with the site, server response issues, reusability & flow of the code, etc.

When the coding standards are implemented, these problems can be easily overcome giving you a secure site with minimum or no performance issues.

While formulating a code, the following should be kept in mind

1. The code should be easy to be read, for this:
    - Try to define different sections of the code by segmenting blocks of code into a paragraph.
    - Make use of indentation for indicating the start and end of the control structures along with a clear specification of where the code is between them.
2. There should be consistency in the naming convention of the variables throughout the code. Also, the data should be described that is there in the code
3. Name the functions according to what they perform
4. The code should be such that one should be able to understand it even after returning to it after some time gap, without that person having to look at every line of it
5. Follow a specific method for commenting on the work
6. The language functions that are complex or the structure that is difficult to be comprehended should be avoided

Refer below images to follow coding standards:

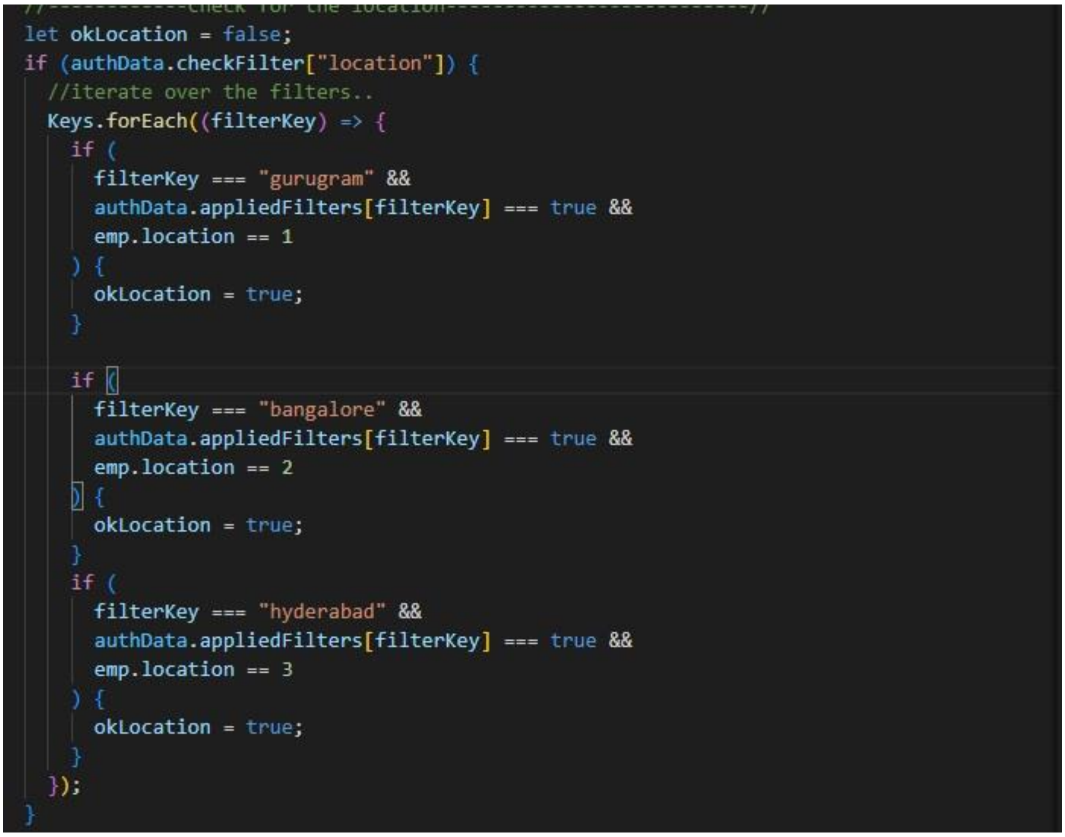


```
const fetchApis = async () => {

  try {
    //assign default location acess
    authData.locationAcess.Gurugram=true;
    authData.locationAcess.Hyderabad=true;
    authData.locationAcess.Bangalore=true;
     //set default chart Stuff
    authData.setPieChartLables(["Gurugram","Bangalore","Hyderabad"]);


    const allnewDto = await axios.post(
      "http://localhost:2538/api/dto/get/filterd", authData.requestDto
    );
    authData.setNewData(allnewDto.data);
    //count emp locatin wise
    const countOfEachLoc = await axios.get(
      "http://localhost:2538/api/empdetails/get/countOfEachLocation"
    ).then((res)=>{
      let tempData=[];
         res.data.forEach(element => {
              tempData.push(parseInt(element.count));
         });
         authData.setCountOfEachLocation(tempData);
    })
    //count of All BU location wise
    //gurugram
    const countOfGurugramBU = await axios.get(
      "http://localhost:2538/api/empdetails/get/gurugramBU"
      ).then((res)=>{
          authData.setGurugramBU(res.data);
    })
```

Image 1 : **follow commenting on the work**



```
//-----------check for the location------------------------//
let okLocation = false;
if (authData.checkFilter["location"]) {
  //iterate over the filters..
  Keys.forEach((filterKey) => {
    if (
      filterKey === "gurugram" &&
      authData.appliedFilters[filterKey] === true &&
      emp.location == 1
    ) {
      okLocation = true;
    }

    if (
      filterKey === "bangalore" &&
      authData.appliedFilters[filterKey] === true &&
      emp.location == 2
    ) {
      okLocation = true;
    }
    if (
      filterKey === "hyderabad" &&
      authData.appliedFilters[filterKey] === true &&
      emp.location == 3
    ) {
      okLocation = true;
    }
  });
}
```

Image 2 : **Use of Indentation on the work**

There are many advantages to the following coding standards while coding the software.

Advantages of Implementing Coding Standards in Software Development

- ✓ **Enhanced Efficiency**
- ✓ **Risk of project failure is reduced**
- ✓ Minimal Complexity
- ✓ Easy to Maintain
- ✓ Bug Rectification
- ✓ A Comprehensive Look
- ✓ Cost-Efficient

In nutshell, coding standards play a vital role in any successful software development.

Best practices that are used to write better codes

- ✓ **Code Comments and Proper Documentation**
- ✓ **Use of Indentation**

- ✓ **Avoid Commenting on Obvious Things**
- ✓ **Grouping Code**
- ✓ **Proper and Consistent Scheme for Naming**
- ✓ **Principle of DRY**
- ✓ **Deep nesting structure should be avoided**
- ✓ **Use short line length**
- ✓ **Proper organization of files and folder**
- ✓ **OOPs vs. Procedural programming**
- ✓ **Open source code readability**
- ✓ **Refactoring of code**

4.3 ERROR HANDLING, PARAMETER PASSING AND CALLING

Member Function Header:

Member function documentation should include the following:

- ✓ What and why the member function does what it does
- ✓ What member function must be passed as parameters
- ✓ What a member function returns?
- ✓ Known bugs
- ✓ Any exception that a member function throws
- ✓ Visibility decisions (if questionable by other developers)
- ✓ How a member function changes the object – it is to helps a developer to understand how a member function invocation will affect the target object.
- ✓ Include a history of any code changes
- ✓ Examples of how to invoke the member function if appropriate.
- ✓ Applicable pre conditions and post conditions under which the function will work properly.

These are the assumptions made during writing of the function.

Fig:– Error Handling Diagrams

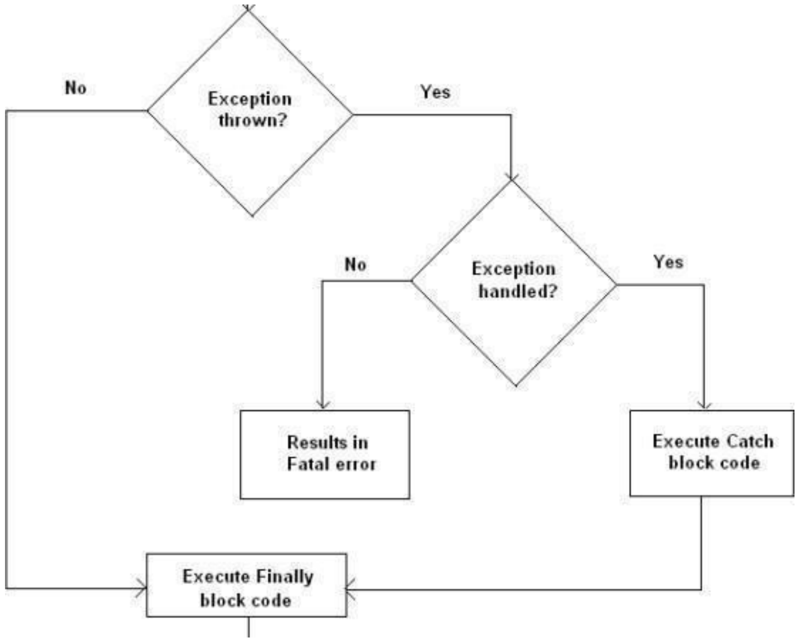


Figure 25: Error Handling Diagrams

CHAPTER 5

SECURITY

Security is the degree of resistance to, or protection from, harm. It applies to any vulnerable and valuable asset, such as a person, dwelling, community, nation, or organization.

As noted by the Institute for Security and Open Methodologies (ISECOM) in the OSSTMM 3, security provides "a form of protection where a separation is created between the assets and the threat." These separations are generically called "controls," and sometimes include changes to the asset or the threat.

Internet Security is a branch of computer security specifically related to the Internet often involving browser security but also network security on a more general level as it applies to other applications or operating systems on a whole. Its objective is to establish rules and measures to use against attacks over the Internet. The Internet represents an insecure channel for exchanging information leading to a high risk of intrusion or fraud, such as phishing. Different methods have been used to protect the transfer of data, including encryption.

5.1 DATABASE OR DATA SECURITY

Data entering a data processing system and the programs processing the data must be kept secure. The following requirements are met to ensure security:

- ✓ The data and programs are protected from theft, fire, disk corruption and other types of physical destruction. Duplicate copies are kept in a fire -proof vault in a place away from the data processing center. This is particularly important for financial data.
- ✓ Data is re-construct able in case of loss despite precautions. Backup copies of master files and transaction files are kept.

- ✓ The system is being tamper -proof. Password system and file security keys are used to bar unauthorized access. If password system is broken by a clever programmer a secrecy transformation is used to transform the stored data.
- ✓ Any person gaining access to a file should be identified. Thus an attempt to access data is logged and identity is also recorded.

So purpose of security is to protect the organization valuable resources, such as information, hardware, and software. Through the selection and application of appropriate safeguards, security helps the organization's mission by protecting its physical and financial resources, reputation, legal position, employees, and other tangible and intangible assets.

Awareness of the types of data loss and the risks associated with losing data is essential for preventing data loss that can be a major cost to your business.

- □ Human Error
- □ Viruses & Malware
- □ Hard Drive Damage
- □ Power Outages
- □ Computer Theft
- □ Liquid Damage
- □ Disasters
- □ Software Corruption
- □ Hard Drive Formatting
- □ Hackers and Insiders

5.2 USER AND ACCESS RIGHTS

Access Rights are the permissions an individual user or a computer application holds to read, write, modify, delete or otherwise access a computer file; change configurations or settings, or add or remove applications. The level of access rights often depends on the user's position or supervisory role in the company.

The benefits of access control are pretty obvious when you think about it. Having a building-wide system can protect your employees, patrons, information, equipment and other assets without question. Before we discuss the types of access control, let's talk a little bit more about the benefits and how access control works.

5.2.1 BENEFITS OF ACCESS CONTROL

The benefits of access control include:

- ✓ **Knowing Who's Coming and Going at All Times**
- ✓ Keep Track of Employees
- ✓ Secure Sensitive Documents and Data
- ✓ Reduce Theft and Accidents
- ✓ Multi-Property Protection
- ✓ No More Worrying About Keys

5.2.2 HOW ACCESS CONTROL WORKS

1. Access control readers give access to the building based on established credentials. Things like a key card, key fob, or biometrics like fingerprints are all considered established credentials.
2. Door readers are connected to a network. Every person who needs access has a code tied to their credential & the system recognizes that they are authorized to be in the building.
3. Software tracks who enters and exits the building and has the ability to alert security supervisors, business owners, etc. when someone enters the building after hours or there is a break-in.

THE THREE TYPES OF ACCESS CONTROL SYSTEMS

In brief, access control is used to identify an individual who does a specific job, authenticate them, and then proceed to give that individual only the key to the door or workstation that they need access to and nothing more. Access control systems come in three variations:

1. Discretionary Access Control (**DAC**)
2. Mandatory Access Control (**MAC**)

3.Role-Based Access Control (**RBAC**)

1. DISCRETIONARY ACCESS CONTROL (DAC)

Discretionary Access Control is a type of access control system that holds the business owner responsible for deciding which people are allowed in a specific location, physically or digitally.

The drawback to Discretionary Access Control is the fact that it gives the end-user complete control to set security level settings for other users and the permissions given to the end-user are inherited into other programs they use which could potentially lead to malware being executed without the end-user being aware of it.

2. MANDATORY ACCESS CONTROL (MAC)

Mandatory Access Control is more commonly utilized in organizations that require an elevated emphasis on the confidentiality and classification of data (i.e. military institutions). MAC doesn’t permit owners to have a say in the entities having access in a unit or facility, instead, only the owner and custodian have the management of the access controls. MAC will typically classify all end users and provide them with labels which permit them to gain access through security with established security guidelines.

3. ROLE-BASED ACCESS CONTROL (RBAC)

Also known as Rule-Based Access Control, RBAC is the most demanded in regard to access control systems. Not only is it in high demand among households, RBAC has also become highly sought-after in the business world.

RBAC makes life much easier because rather than assigning multiple individuals particular access, the system administrator only has to assign access to specific job titles.

5.3 STEPS TAKEN AGAINST HACKING OF SYSTEM

The Open Web Application Security Project (OWASP) is a nonprofit foundation that works to improve the security of software. Our programming includes:

- ✓ Community-led open source software projects
- ✓ Over 275 local chapters worldwide
- ✓ Tens of thousands of members
- ✓ Industry-leading educational and training conferences

We are an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. All of our projects, tools, documents, forums, and chapters are free and open to anyone interested in improving application security. The OWASP Foundation launched on December 1st, 2001, becoming incorporated as a United States non-profit charity on April 21, 2004.

Small businesses are responsible for preventing these crimes from harming company property and stealing consumer information. Here are some steps you can take to protect your company's computer integrity.

1. Use a firewall
2. Install antivirus software
3. Install an anti-spyware package
4. Use complex passwords
5. Keep your OS, apps and browser up to date
6. Ignore spam
7. Back up your computer
8. Shut it down
9. Use virtualization
10. Secure your network
11. Use two-factor authentication
12. Use encryption

5.4 SQL INJECTION ATTACK

A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data (Insert/Update/Delete), execute administration operations on the database (such as shutdown the DBMS), recover the content of a given file present on the DBMS file system and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.

Solution: - To handle the SQL injection, query is not created by concatenation of string, queries are executed by using Prepared Statement reference object and value passed.

CHAPTER 6

TESTING

6.1 TEST CASE DESIGN

A test case provides the description of inputs and their expected outputs to observe whether the software or a part of the software is working correctly. IEEE defines test case as 'a set of input values, execution preconditions, expected results and execution post conditions, developed for a particular objective or test condition such as to exercise a particular program path or to verify compliance with a specific requirement.' Generally, a test case is associated with details like identifier, name, purpose, required inputs, test conditions, and expected outputs.

Additional information that may be included:

- ✓ Test case ID
- ✓ Test case description
- ✓ Test step or order of execution number
- ✓ Related requirement(s)
- ✓ Depth
- ✓ Expected Result
- ✓ Resource Name
- ✓ Actual Result
- ✓ Status
- ✓ Remarks

- **Test Case ID:** Test case ID is the serial number of test cases according to the screens and the scenarios applies in the test cases.
- **Test Case Description:** Test case description is the explanation of test cases which shows under which scenario test cases will execute.
- **Test Step:** It shows the navigation and order of sequence of test cases.
- **Related requirements:** This is the collections of required things which are essential for executing related test cases.

- **Depth**: It shows the brief description and effects of the related test cases.
- **Expected Result**: It is the expected effect of test case guess by testers before executing test case.
- **Resource Name**: This is the name of testers who writes and executes test cases.
- **Actual Result**: This is the actuality that shows exactly what happened after executing test case.
- **Status**: This is the status of the test cases before or after executing test cases. It should be like
  - ✓ PASS
  - ✓ FAIL
  - ✓ HOLD
  - ✓ INVALID
  - ✓ TIP (Testing in Progress)
- **Remarks**: It is generally the reason of failed, hold, invalid and TIP test cases.

6.2 TEST REPORTS

Test Report is needed to reflect testing results in a formal way, which gives an opportunity to estimate testing results quickly. It is a document that records data obtained from an evaluation experiment in an organized manner, describes the environmental or operating conditions, and shows the comparison of test results with test objectives.

6.2.1 DIFFERENT LEVELS OF TESTING

UNIT TESTING: The main idea of unit testing is testing software with a small piece of source the software consists of "units" which are separate testable parts of the product. An individual program, class, method, function etc. can be such "unit". It allows checking whether a unit behaves as the developer intended and whether a unit corresponds to the design specifications. Unit testing provides an ability of independent testing for each software unit. It should run at the push of the button. It should run quickly too.

INTEGRATION TESTING: Integration Testing is executed to establish whether the components interact with each other consort to the specification or not. Integration testing in large refers to joining all the components resulting in the complete system. It is further performed by the developer or the software Tester or by both. This was an essential testing for those modules, which invoked other child modules. After each module is developed and tested separately, they have been integrated and have been tested for any errors. Various modules have been integrated with other modules and even with service oriented architecture modules for enhancement.

USER ACCEPTANCE TESTING: Acceptance testing is a test conducted to determine if the requirements of a specification or a contract are met. It may involve Chemical tests, Physical tests or performance tests. In System engineering it may involve black – box testing performed on a system prior to its delivery. Software developers often distinguish acceptance testing by the system provider from acceptance testing by the customer prior to accepting transfer of ownership. In the case of software, acceptance testing performed by customer is known as user acceptance testing, end user testing, or field testing.

BLACK BOX TESTING: Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. Tester is mainly concerned with the validation of output rather than how the output is produced (functionality of the item under test is not important from tester's pod).

WHITE BOX TESTING: White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. Tester validates the internal structure of the item under consideration along with the output. Programming knowledge and implementation knowledge (internal structure and working) is required in White Box testing, which is not necessary in Black Box testing.

PERFORMANCE TESTING: Performance testing is defined as the technical investigation done to determine or validate the speed, scalability, and/or stability characteristics of the product under test. Performance-related activities, such as testing and tuning, are concerned with achieving response times, throughput, and resource-utilization levels that meet the performance objectives for the application under test. Because performance testing is a general term that covers all of its various subsets, every value and benefit listed under other performance test types in this chapter can also be considered a potential at benefit of performance testing.

6.2.2 TEST CASES

A test case, in software engineering, is a set of conditions or variables under which a tester will determine whether an application, software system or one of its feature is working as it was originally established for it to do. In some settings, it could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written - when they are usually collected into test suites.

6.2.3 Test Approach

Testing can be done in two ways:

- ✓ Bottom up approach
- ✓ Top down approach

BOTTOM UP APPROACH: Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

TOP DOWN APPROACH: This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

6.2.4 VALIDATION: The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

6.2.5 STLC (Software Testing Life Cycle)

TEST PLANNING:

1. Test Plan is defined as a strategic document which describes the procedure how to perform various testing on the total application in the most efficient way
2. This document involves the scope of testing
3. Objective of testing
4. Areas that need to be tested
5. Areas that should not be tested
6. Scheduling Resource Planning
7. Areas to be automated, various testing tools

TEST DEVELOPMENT

1. Test case Development (check list)
2. Test Procedure preparation. (Description of the Test cases)
3. Implementation of test cases. Observing the result

RESULT ANALYSIS:

1. **Expected value:** is nothing but expected behaviour of application.
2. **Actual value:** is nothing but actual behaviour of application.

Bug Tracing: Collect all the failed cases, prepare documents.

Reporting: Prepare document (status of the application)

TESTTING TOOL USED

- Go-sqlmock
- Testify
- Testing
- Assert

6.3 DEBUGGING AND CODE IMPROVEMENT

In ideal worlds, all programmers would be so skilled and attentive to detail that they would write bug-free code. Unfortunately, we do not live in an ideal world. As such, debugging, or tracking down the source of errors and erroneous result, is an important task that all developers need to perform before they allow end-user to use their applications. We will discuss some techniques for reducing the number of bugs in code up front.

There are three Categories of Bugs-

1 **Syntax error:** These errors occur when code breaks the rule of the language, such as visual Basic sub statement without a closing End sub, or a forgotten closing curly braces ({}) in C#. Theses error the easiest to locate. The language complier or integrated development environment (IDE) will alert you to them and will not allow you to compile your program until you correct them.

2 **Semantic error:** These errors occur in code that is correct according to rules of the compiler, but that causes unexpected problems such as crashes or hanging on execution. A good example is code that execute in a loop but never exists the loop, either because the loop depends on the variable whose values was expected to be something different than it actually was or because the programmer forgets to increment the loop counter. Another category of errors in this area includes requesting a field from a dataset, there is no way to tell if the field actually exists at compile time. These bugs are harder to detect and are one type of running error.

3 **Logic error:** Logic errors are like semantic errors, logic errors are runtime error. That is, they occur while the program is running. But unlike semantic errors, logic errors do not cause the application to crash or hang. Logic error results in unexpected values or output. This can be a result of something as simple as a mistyped variables name that happens to match another declared variable in the program. This type of error can be extremely difficult to track down to eliminate.

PREVENTING DEBUG WRITE REDABLE CODE

Develop and make consistent use of naming and coding standards. It not that important which standard we use, such as Hungarian notation or Pascal, Casing (First Name) or other naming conventions, as long as we use one. We should also strive for consistency in our comments and encourage liberal commenting code.

Create Effective Test Plan

The only effective way to eliminate logic error is to test very path of your application with every possible data value that a user could enter. This is difficult to manage without effective planning. We should create our test plan at the same time we are designing the application, and we should update these plans as you modify the application design

Defect Testing

Defect testing is testing for situation where the program does not meet its fictional specification. Performance testing tests a system's performance or reliability under realistic loads. This may go some way to ensuring that the program meets its non-functional requirements.

6.4 EXECUTION OF TESTING

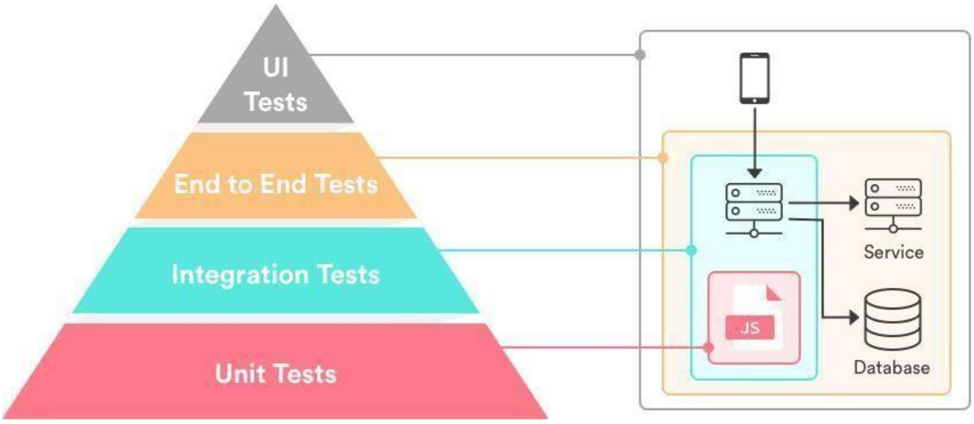
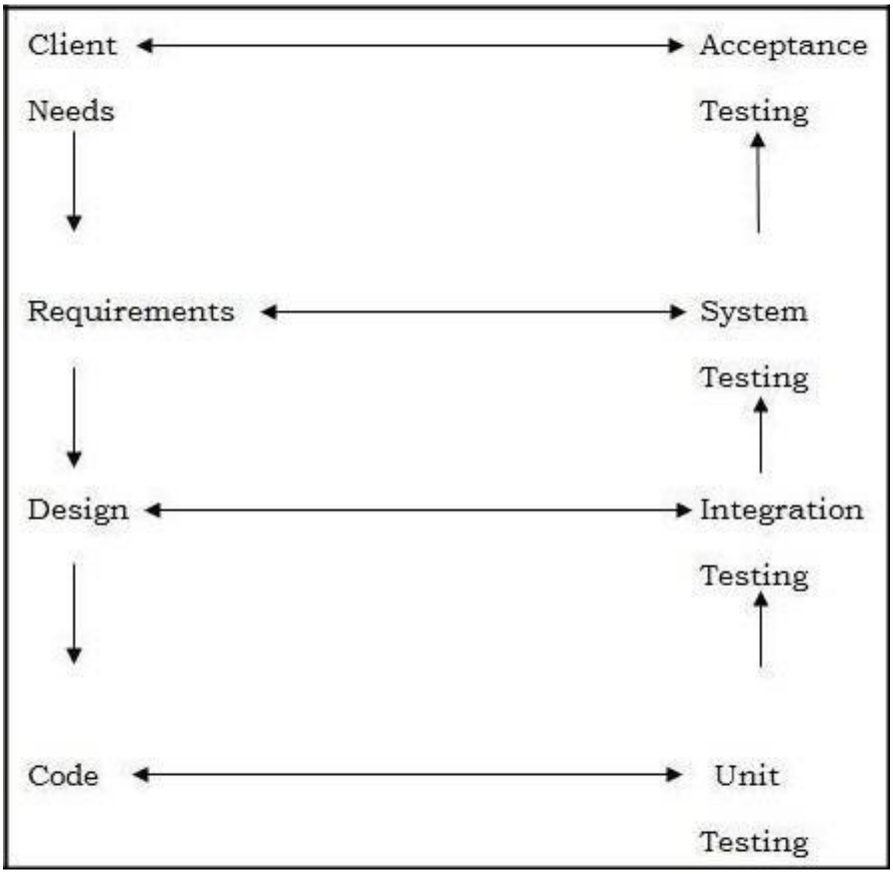


Figure 26 :Execution of Testing



CHAPTER 7

CONCLUSION AND FUTURE SCOPE FOR FURTHER DEVELOPMENT

7.1 CONCLUSION

This project has allowed me to learn about a lot of new technologies that I was unaware of. I got to know about how a project is developed in a company and all the different stages it goes through during its lifecycle. I came to know about how the thought process of developing a project differs from a college project and a company project. I also got to know the importance of documentation along with the project itself. My contribution while doing the handling the backend of project. I had also completed some of the pending work which leads to its completion so we could move further to the next project.

With this assignment of task **Bench Management** , I was able to learn how a production grade codes are written as well as why time complexity is so crucial to any server.

7.2 FUTURE WORK

-There is a need to add more filter functionality so that every manual task can be achieved.

- Currently Project is working on Temporary Data. We can connect it with any Organization Real Time Data. Which is our end goal.

REFERENCES

WEB REFERENCES

- https://spring.io/guides/tutorials/react-and-spring-data-rest/
- https://www.w3schools.com/js/DEFAULT.asp
- https://www.baeldung.com/get-started-with-java-series
- https://www.javatpoint.com/sql-tutorial
- https://www.geeksforgeeks.org/reactjs-tutorials/
- https://www.geeksforgeeks.org/best-way-to-master-spring-boot-a-complete-roadmap/
- https://www.geeksforgeeks.org/spring-boot-architecture/
- https://www.geeksforgeeks.org/easiest-way-to-create-rest-api-using-spring-boot/
- https://www.baeldung.com/spring-boot-start

BOOK REFERENCES

- The Road to Learn React by Robin Wieruch
- A Smarter Way to Learn JavaScript by **Mark Myers**