CHAPTER 1: OVERVIEW OF THE COMPANY

1.1HISTORY

UNNATI INFORMATICS has championed technological innovation. With competent teams of developers, project managers, and strategists, we help our clients overcome their business challenges with customized software development. Our services and solutions help your business. We contribute to the success of some of the powerful leading brands in the modern market.

We are a software development organization and deliver state-of-the-art IT business solutions to our clients across the globe. We combine technical expertise and business acumen to deliver great results.

Our CSR initiatives are in line with our corporate values and guide our company in philanthropic pursuits; we support initiatives in areas such as education, health, and social welfare, which lead to the sustainable long-term development of communities.

We proactively work for society since we believe that corporate success and socialwelfare are interdependent; our mission is to facilitate employee-driven social initiatives that make a positive contribution to society and drive innovation.

Vision: To become The Best IT Solutions and Services platform by giving optimal solutions to Businesses and Educational Institutions for their consistent growth.

Mission: To provide software solutions that are functional, reliable, maintainable,and cost-friendly to our existing and growing client and customer base. To consistently cater to their growing needs for an optimal solution, ensuring excellent support and service platform to give a hassle-free experience in achieving their dreams.

1.2DIFFERENT SCOPE OF WORK

- **Design**

  UI/UX

  Frontend

  Wireframe

- **Development**

  Web development Mobile

  App developmentDigital

  commerce

- **Project Training**

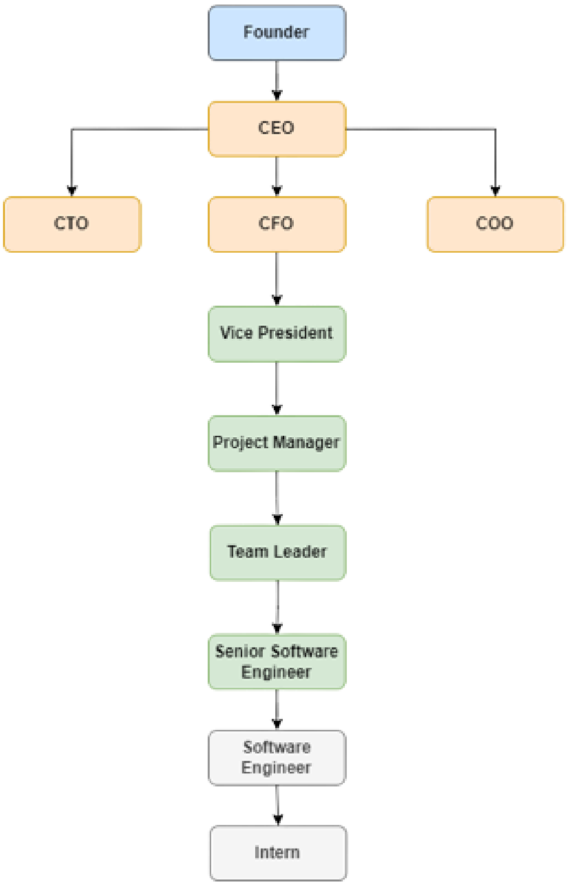
  PHP
  MEAN Stack/ MERN Stack
  Python Django
  Spring Boot

- **Deployment**

  Biometric attendance system

  School management software

1.3 ORGANIZATION CHART



Unnati Informatics LLP

(Figure 1.1 Organization Chart)

1.4 CAPACITY OF OFFICE

UNNATI INFORMATICS LLP is big enough to handle the medium-scale projects. It has almost 50+ staff including all junior developers, senior developers, project managers, interns, and the CEO. It has a large space and is equipped with modern more than 35 computers. At present time company is planning to expand its area by opening another branch in Ahmedabad. Some employees currently working as work from home. Office time is Monday to Saturday rather than Friday to gain more productivity.

CHAPTER 2: OVERVIEW OF DIFFERENT DEPARTMENT OF ORGANIZATION ANDLAYOUT OF THE PRODUCTION BEING CARRIED OUT IN COMPANY

2.1 DETAILS ABOUT THE WORK OF EACH DEPARTMENT

In our organization, there are different departments like web development, Digital marketing, App development, Project Training, etc.

Web development

UNNATI INFORMATICS provides different web development like custom web development, UX & prototyping designing progressive web development, software product development, customized API development, etc. We also do mobile app development. In which we will use ANDROID, IOS, FLUTTER, etc. It is Designed with Fully-Optimized Responsive & Modern Designs.

PHP, JAVA, PYTHON, .NET, ROR, NODE JS, etc. is the main key to web development through a highly dedicated development team.

Android development

UNNATI INFORMATICS offers good service in the area of app development. In the company, a cross-platform application is also being built but the major focus is on the android app. In this department also 12+ human resources are allocated including junior and senior developers and project managers. In-app development, the company mainly focuses on management apps like school and college, hospitals, traffic, restaurants ant, and other shopping shops.

Design

Attractive and powerful designs offer a unique experience to the customers and

UI/UX has been disrupting the digital market. The user interface generates a unique user experience that begins the customer journey. A specialized designing team does a design through HTML, CSS, and bootstrap.

The front-end developer is using REACT JS and ANGULAR JS for designing.

Deployment

DevOps & cloud developers come under deployment. Deployment usingkaizen implementation is good.

Some of the cloud platforms we use are the GOOGLE cloud platform, AMAZONweb services, MICROSOFT AZURE, etc.

Salesforce is the World's largest and most globally popular cloud CRM platform. It helps an organization manage customer data, processes, and operations thereby establishing and maintaining warm relationships with the customers.

2.2 LIST OF TECHNICAL SPECIFICATIONS OF MAJOR EQUIPMENT USED IN EACH DEPARTMENT

Web development

Android studio: Used in android app development Visual
studio: Used in web application developmentVisual studio
code: Used in front end development
SQL Server Management Studio: Used for database managementMy
SQL workbench: Used in database management
Anaconda: Used in python
MEAN and MERN: for complete web development
Xampp: This server is used to handle database

Android Development

Android studio: Used in android app developmentGoogle
Firebase: Storing database online
Sqlite: database storage
Android phone: for testing purposes
Mac: for developing apple store apps
Third-party designs: purchased subscription from online design site for app

Design

Figma: for creating an online logo and vector graphics

Adobe Photoshop: edit and modify the current design of the website or app and create a demo design the of the whole project for analysis or demo purpose

Adobe XD: edit graphics

Wireframe. cc: used early in the development process to establish the basic structure of a page before visual design and content is added.

Digital marketing

HubSpot marketing

2.3 PREPARE SCHEMATIC LAYOUT WHICH SHOWS THE SEQUENCE OF OPERATION FOR MANUFACTURING OF ENDPRODUCT

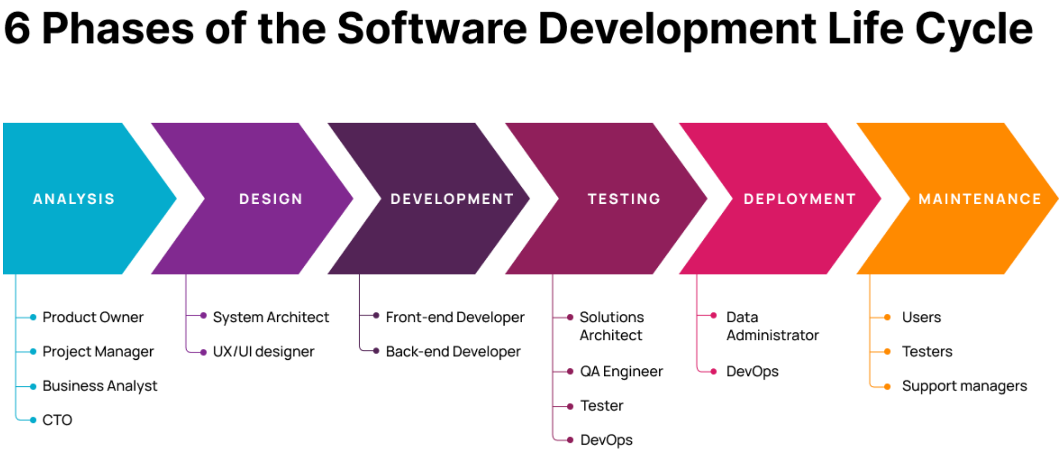


Figure 2.1 Phases of SDLC

2.4 EXPLAIN IN DETAIL ABOUT EACH STAGE OF PRODUCTION

1. Planning and Requirement Analysis

Requirement analysis is the most fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys, and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility studies in the economical, operational and technical, areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

2. Design

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually, more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third-party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

3. Development

In this stage of SDLC, the actual development starts, and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

4. Testing

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, at this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed, and retested, until the product reaches the quality standards defined in the SRS.

5. Deployment

Once the software is certified, and no bugs or errors are stated, then it is deployed. Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment. After the software is deployed, then its maintenance begins.

6. Maintenance

Once when the client starts using the developed systems, then the real issues comeup and requirements to be solved from time to time.

CHAPTER 3. INTRODUCTION TO INTERNSHIP AND INTERNSHIP MANAGEMENT

3.1 INTERNSHIP SUMMARY

I learned many things like how work is carried out in the company and how the complete process of IT solutions is carried out. It was a very nice experience and allowed me to deep dive into the real-time work environment and was a much more informative experience that opens a wide door for my career prospects.

UNNATI INFORMATICS is an IT service-based company in Ahmedabad. UNNATI INFORMATICS is sufficiently qualified to help clients to provide IT services.

Within these few months in my internship So much different things about software development is learned. How to work on a live project. Web development is my work. I can honestly say that my time spent interning with this Company resulted in one of the best summers of my life. With practical skills, I got the opportunity to meet industry experts.

The atmosphere at the office was always welcoming which made me feel right at home. Additionally, I felt like I was able to contribute to the company by assisting and working on projects throughout the summer. Working on a live project is a different thing. There are many opportunities and problems when you are working on a live project.

During this internship, I learned about Web development in the company with the working project of the company and the same title personal project called Bug tracking system.

Bug Tracking Web application: This is my project based on the same topic in this I have implemented functionality alone.

3.2 PURPOSE AND OBJECTIVE

I tried to implement all the major and basic functions of the company’s project. The main purpose of my app is to learn how to do work and develop a Web application without the help of others.

3.3 SCOPE

1.4.1 It provides an easy way for Bug solving.
1.4.2 Manager will add an employee.
1.4.3 Employee will log in with username and password.
1.4.4 This system can save time and effort for the employees.
1.4.5 Tester will add bus.
1.4.6 Developer will see the bugs and solve them and update as resolved.

3.4 TECHNOLOGY AND LITERATURE REVIEW

We are using a python-Django framework for back-end and other front-end languages like HTML, CSS, and JavaScript.

3.4.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and Scripting languages such as JavaScript.

3.4.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.

3.4.3 JavaScript

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high level, often just in time compiled, and multiparadigm. It has curly-bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions.

3.4.4 PHP

Django is a free and open-source web application framework written in Python. It is used for rapid web development and clean, pragmatic design. It is built by experienced developers to make repetitive tasks easier, so we can focus on writing apps instead of reinventing the wheel.

3.4.5 PostgreSQL

PostgreSQL is an advanced, enterprise-class, and open-source relational database system. PostgreSQL supports both SQL and JSON querying. PostgreSQL is used as a primary database for many web applications as well as mobile and analytics applications.

3.5 INTERNSHIP PLANNING

3.5.1 Project development approach and justification

To solve actual problems in an industry setting, a software engineer or a team of engineers must incorporate a development strategy that encompasses the process, methods, tools layers, and generic phrases. This strategy is often referred to as a process model or a software engineering paradigm or a project development approach.

A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required.

In this Bug Tracking project, we used a RAD model which is known as the **Rapid Application Development Model**

RAD (Rapid Application Development) is a concept that products can be developed fasterand of higher quality through:

- Gathering requirements using workshops or focus groups
- Prototyping and early, reiterative user testing of designs
- The re-use of software components
- A rigidly paced schedule that refers to design improvements to the next product version
- Less formality in reviews and other team communication

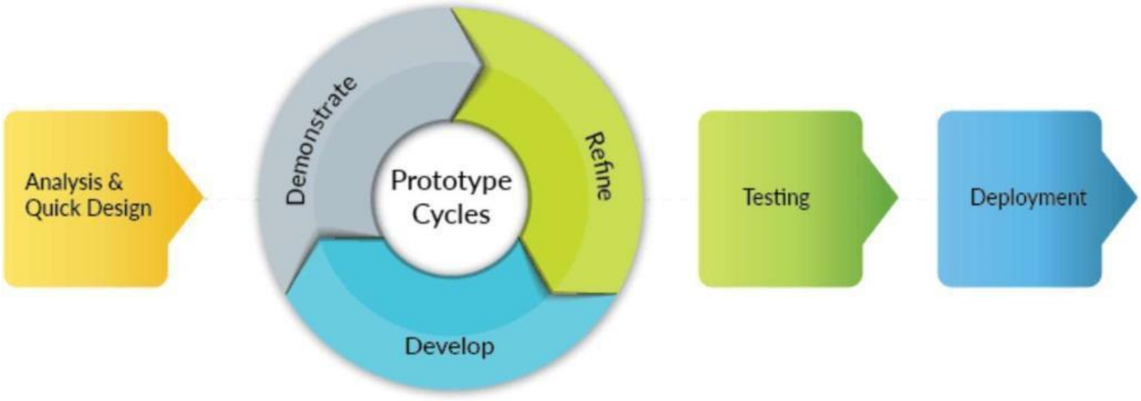


Figure 3.1 RAD model

Concerning the system development methodology. Since it allows fast development and delivery of a high-quality system, it was the perfect methodology for my project. I had permanent contact with the client throughout the different stages of the software process. I developed a major part of the real-time project alone and test it and give feedback about what should be changed or improved. This interaction saved mea lot of time because the feedback was instantaneous. Accordingly, the user was involved in the design. Also, since the project is the priority is to fulfill the business need of the company; which goes hand to hand with the main principles of the Rapid Application Development methodology. Moreover, since it is a project, we have deadlines and we need to meet them. This is among the basic principles of RAD. The presence of deadlines or "timeboxes".

3.5.2 Project Effort and Time, Cost Estimation

In this project, we need a lot of effort and time because all the things we have to do is first we have to understand the whole project. What is the main purpose behind this project?

How can we do this project? Always a good effort is needed to complete this project successfully. Good effort always required more time.

The effort is the number of units of work needed to complete a task. It is usually expressed in hours, days or weeks worked. The effort is therefore the number of hours of work needed to complete a task, ie the actual time spent working on the project. To estimate the duration of a project, first, we have to determine the effort.

Best cost estimation is essential for project management success. Many costs can appear over the project management life cycle, and an accurate cost estimation method can be the difference between a successful plan and a failed one. Cost estimation, however, is easier said than done. Projects bring risks, and risks bring unexpected costs and cost management issues.

Cost estimation is the process that takes direct costs, indirect costs, and other factors into account, and calculates a budget that meets the financial commitment necessary for a successful project. Project cost estimation techniques apply to any project, from building a bridge to developing that new killer Web application. Everything costs money, so the clearer you are on the amount required, the more likely you'll achieve your objective.

3.5.3 Roles and Responsibilities

A successful project requires the project team to participate (at some level) in the planning process, buy into the project plan, and be responsible for the completion of assignments. It is important to have a defined formal structure for the project and the project team. This provides each individual with a clear understanding of the authority given and the responsibility necessary for the successful accomplishment of project activities. This section describes the typical roles and responsibilities for projects. Roles may be assigned to one or more individuals. Conversely, individuals may have one or more roles in a project.

Each person plays a different role to complete a project. In this team, there are project managers, project coordinators, senior developers, and junior developers who play an important role. My work in this project is to help developers learn the development also some common problem-solving in the different modules of the web application and find possible solutions.

Project manager responsibilities may include:

- Developing a project plan
- Managing deliverables according to the plan
- Recruiting project staff
- Contributing to overall project objectives
- Completing individual deliverables
- Providing expertise
- Working with users to establish and meet business needs
- Documenting the process

All the work I have to do alone so from scratch I have to build a web applicationthrough the testing. All the bugs that occurred during the development, I had solved from the internet.

3.5.4 Group Dependencies

In a big project, all members are working in a group so dependency on others comes. But dependency is not a big problem because when you are working in a team thenautomatically all the load of the project is divided. Dependencies in project management deal with managing and scheduling project tasks while keeping their sequences and requirements in mind. If task B requires the completion of task A, then we'll say that task B is dependent on task A. This may sound simple right nowbut in complex projects with several interdependent tasks, things can get messy.

Similar to many other challenges in project management, there is no textbook way of dealing with dependencies. How you'll deal with them depends on the requirements and the conditions of the project at hand. It's important to identify and record all the dependencies during the project plan. Otherwise, you'll risk

3.6 INTERNSHIP SCHEDULING

The below chart clearly shows how days are divided throughout the internship. I had spent most of the time designing and developing of the application Hospital management system.

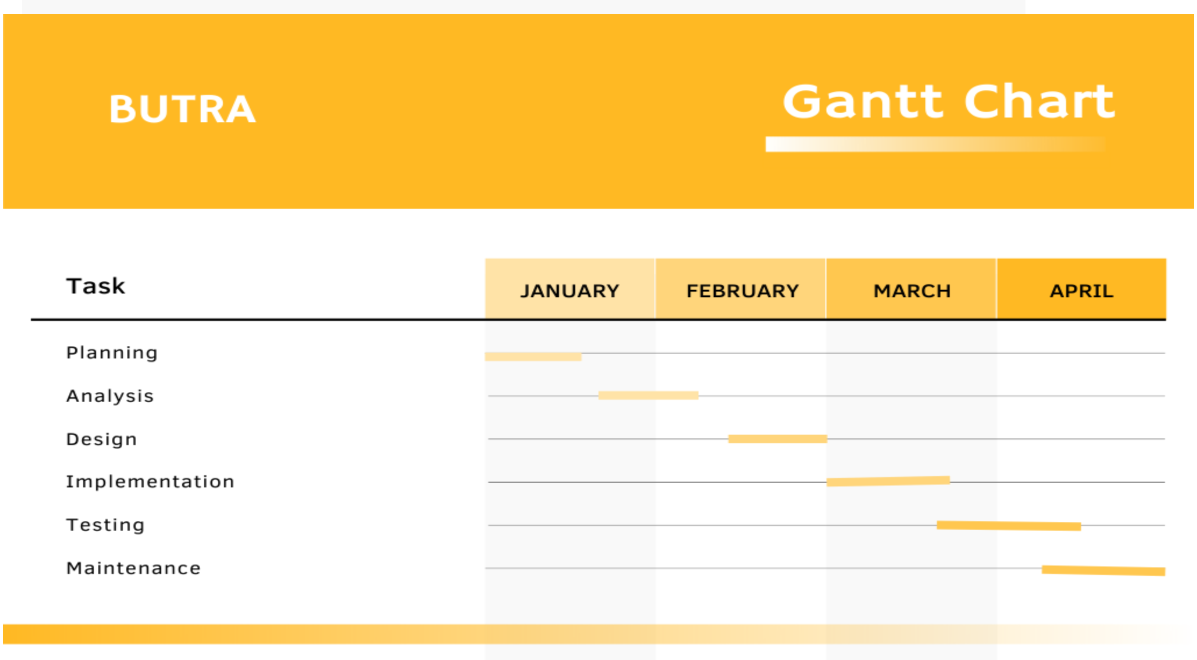


Figure 3.2 Gantt Chart

CHAPTER 4: SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

Recently I had studied many Bug tracking projects including websites and applications which is based on either mobile technology or web technology. It manages all project issues (bugs). Sometimes the project has any problems regarding any particular bug and can be identified using their types and category.

4.2 PROBLEM AND WEAKNESSES OF THE CURRENT SYSTEM

The basic problems of the existing system are the non-interacting environment they provide to the users. User is not free to use an application because of non-user-friendly interfaces. Due to the current pandemic, our study has only been conducted online website. And there are many other weaknesses in the current system.

4.3 REQUIREMENTS OF NEW SYSTEM

The question of the new requirement begins with managing large numbers of bugs and much more functionality. In larger companies, they provide a well quality application so they need to be free of any bugs so they need accurate well-known software to handle it.

The old available application is not enough to provide all the functionality to the developer for problem-solving.

4.4 SYSTEM FEASIBILITY

The feasibility analysis phase is very important, because it is the one that helps the software engineer to see the negative sides of his project and it helps him to refine the software process. The feasibility study can make any project a successful one if it is done correctly. This deliverable is the feasibility study for the project "Bug tracking system"

4.4.1Product Feasibility

The system that will be developed will help the client achieve their objectives This will be achieved thanks to the Web application that will make access to the Manager, developer, and their services, and other employees easily.

4.4.2Technical and Operational Feasibility

The website is based on JavaScript and Django it is very robust for completion for such kinds of projects even large-scale projects can be easily managed in this. Apart from this, the database used in this is PostgreSQL which is reliable and not to worry about that side. Our company has enough Web developers and senior developers, Test experts, and project managers to accomplish given tasks easily in the given time frame. One issue may occur about design like our design may not be liked by company manager but can be solved any time with the changes in it.

4.5 FEATURES OF THE SYSTEM

4.5.1 Authentication

- Login – the employee can log in to the system with his/her credentials.
- Logout – the employee can log out from the system.
- Login failure – if the employee does not exist in the database or the employee has not yet been authorized by the admin of the system.

4.5.2 Authorization

- Employee role check – after logging in to the system, the employee role will be checked from the database employee interface and will be displayed according to their role.

4.6 LIST OF MAIN MODULES

1. Create an account
2. Log in to the system
3. Add new employee
4. Add project
5. Create a team for projects
6. Place bugs
7. Update bug

4.7REQUIREMENT SPECIFICATION

4.7.1 Software Requirements

1. Operating System
2. Coding language
3. Front End & Back End

4.7.2 Hardware Requirements

1. System: Intel Core i5
2. Hard Disk: 1TB
3. RAM: 16GB

CHAPTER 5: SYSTEM DESIGN

5.1 SYSTEM DESIGN & METHODOLOGY

System design is the process of defining elements of a system like modules, architecture, components, and their interfaces for a system based on the specified requirements. It is the process of defining, developing, and designing a project which satisfies the specific needs and requirements of a business or company.

Design methods:

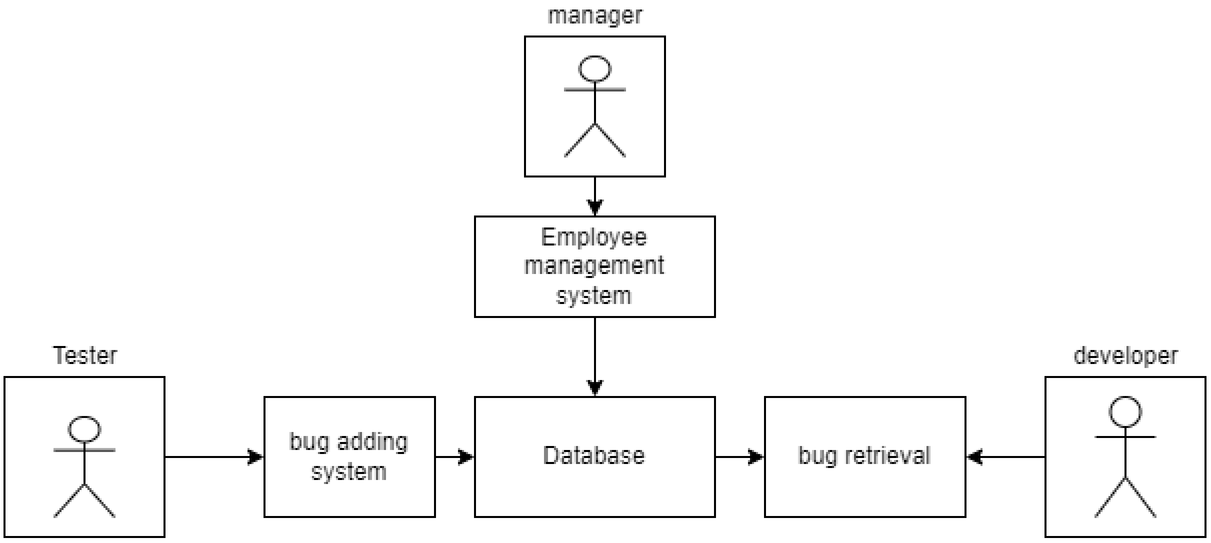
1) Architectural design: To describe the views, models, behavior, and structure of the system.

2) Logical design: To represent the data flow, inputs, and outputs of the system.
Example:ER Diagrams (Entity Relationship Diagrams).

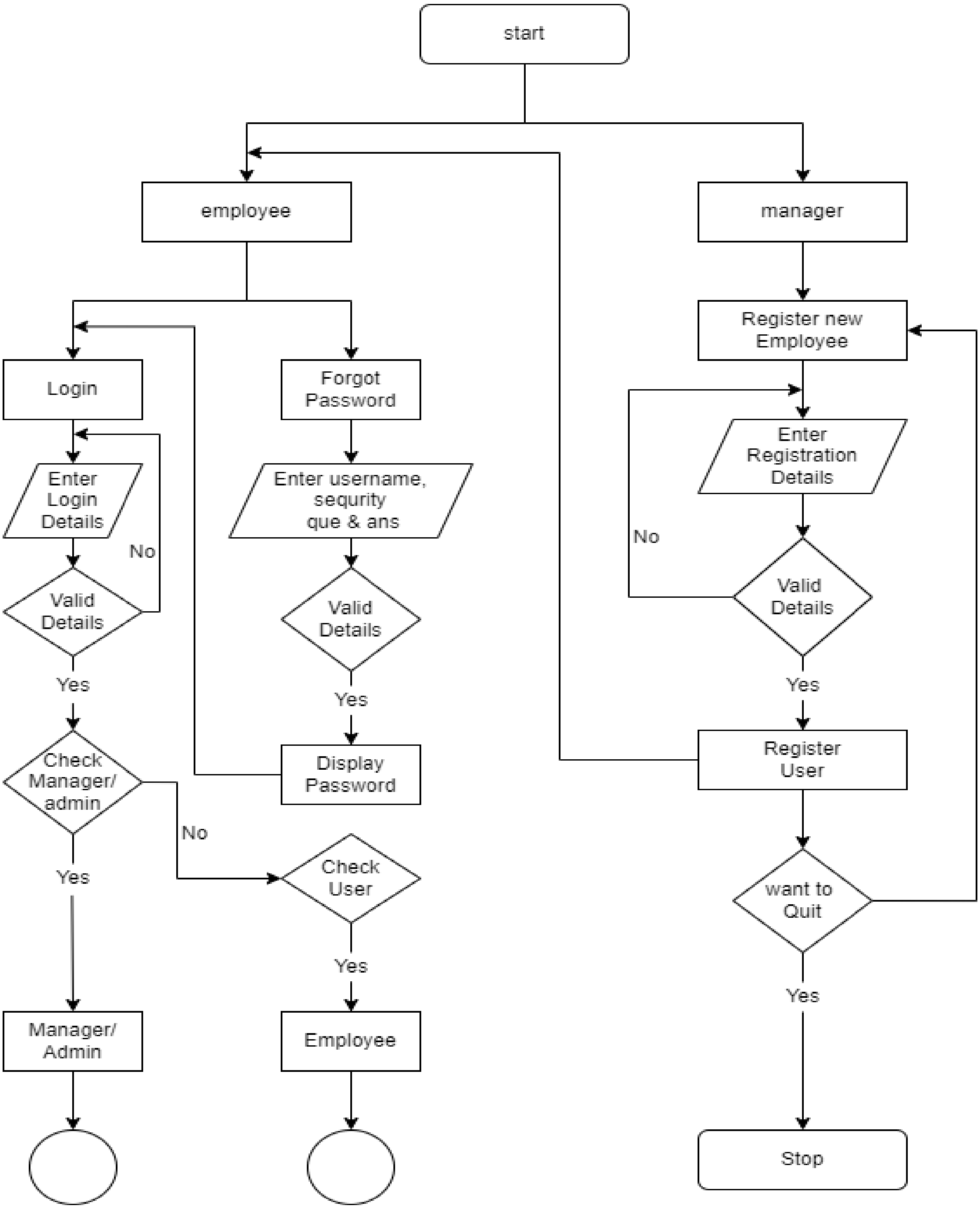
3) Physical design: Defined as

a) How users add information to the system and how the system represents information back to the user. b) How the data is modeled and stored within the system. c) How data moves through the system, how data is validated, secured, and or transformed as it flows through and out of the system.

5.2 DIAGRAM



(Figure 5.1 System Design & Methodology)



(Figure 5.2 System Flow diagram)

5.3 Database Design

Sr.NO	Field Name	Data Type (Size)	Constrain	Description
1	id	Int ()	Primary key	User id
2	role	Int ()	Foreign key	Role id
3	username	Varchar (99)	Not null	Employee name
4	email	Varchar (150)	Unique	Email id
5	Phone_number	Int (10)	Unique	Mobile number
6	password	Varchar (99)	Not null	login password
7	picture	Varchar (50)	Null true	Emp image
8	address	Varchar (100)	Null true	Address
9	city	Varchar (50)	Null true	City
10	State	Varchar (50)	Null true	Sate
11	Country	Varchar (50)	Null true	Country
12	Pincode	Varchar (10)	Null true	Pincode

Table 5.1 Employee details

Sr.NO	Field Name	Data Type (Size)	Constrain	Description
1	id	Int ()	Primary key	Role id
2	rolename	Varchar (99)	Not null	Role name

Table 5.2 role details

Sr.NO	Field Name	Data Type (Size)	Constrain	Description
1	id	Int ()	Primary key	Role id
2	projectname	Varchar (30)	Not null	Project name
3	description	Varchar (500)	Not null	Project description
4	Technology	Varchar (100)	Not null	Project technology
5	startDate	Int ()	Not null	Project start date
6	completionDate	Int ()	Not null	Project end date

Table 5.3 Project details

Sr.NO	Field Name	Data Type (Size)	Constrain	Description
1	id	Int ()	Primary key	team id
2	team_name	Varchar (30)	Not null	team name
3	project	Int ()	Foreign key	Project name
4	Employee	int ()	Foreign key	Team employee

Table 6.4 Project team details

Sr.NO	Field Name	Data Type (Size)	Constrain	Description
1	id	Int ()	Primary key	team id
2	statusName	Varchar (50)	Not null	status name

Table 6.5 status details

Sr.NO	Field Name	Data Type (Size)	Constrain	Description
1	id	Int ()	Primary key	team id
2	bugname	Varchar (100)	Not null	bug name
3	description	Varchar (100)	Not null	Bug discription
4	bugtype	Varchar (50)	Not null	Bug type
5	Status	Int ()	Foreign key	Status of bug
6	Project	Int ()	Foreign key	Name of project
7	startDate	Int ()	Not null	Start date of bug solve
8	completionDate	Int ()	Not null	Expected date of solution
9	employee	Int ()	Foreign key	Name of employee

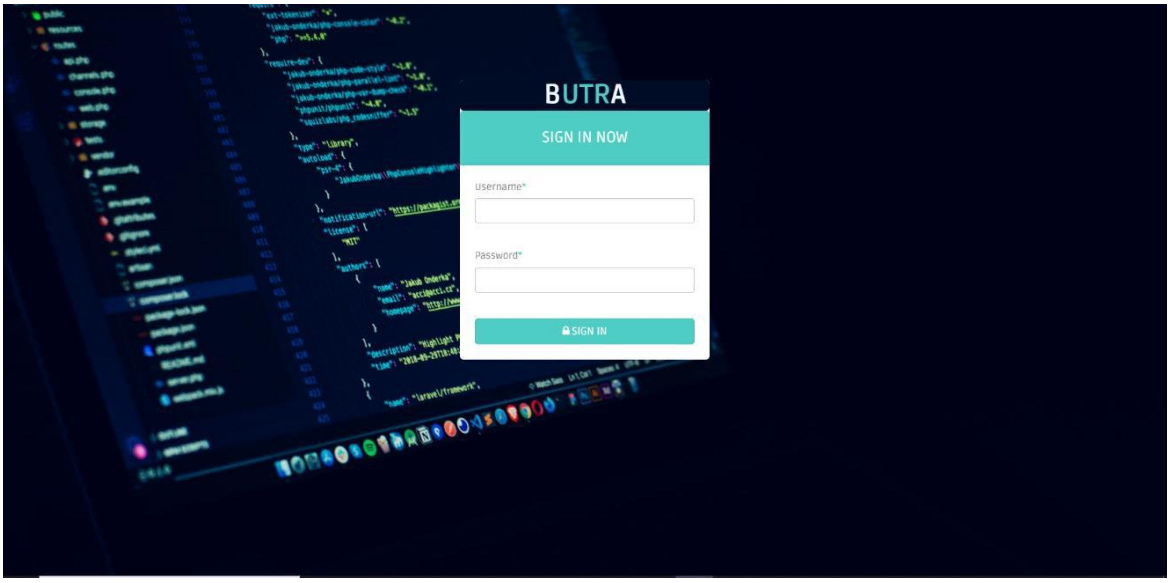
Table 6.6 bug details

Sr.NO	Field Name	Data Type (Size)	Constrain	Description
1	id	Int ()	Primary key	team id
2	name	Varchar (100)	Not null	name of employee
3	Email	Varchar (100)	Not null	Email address
4	Message	Varchar (500)	Not null	Feedback message

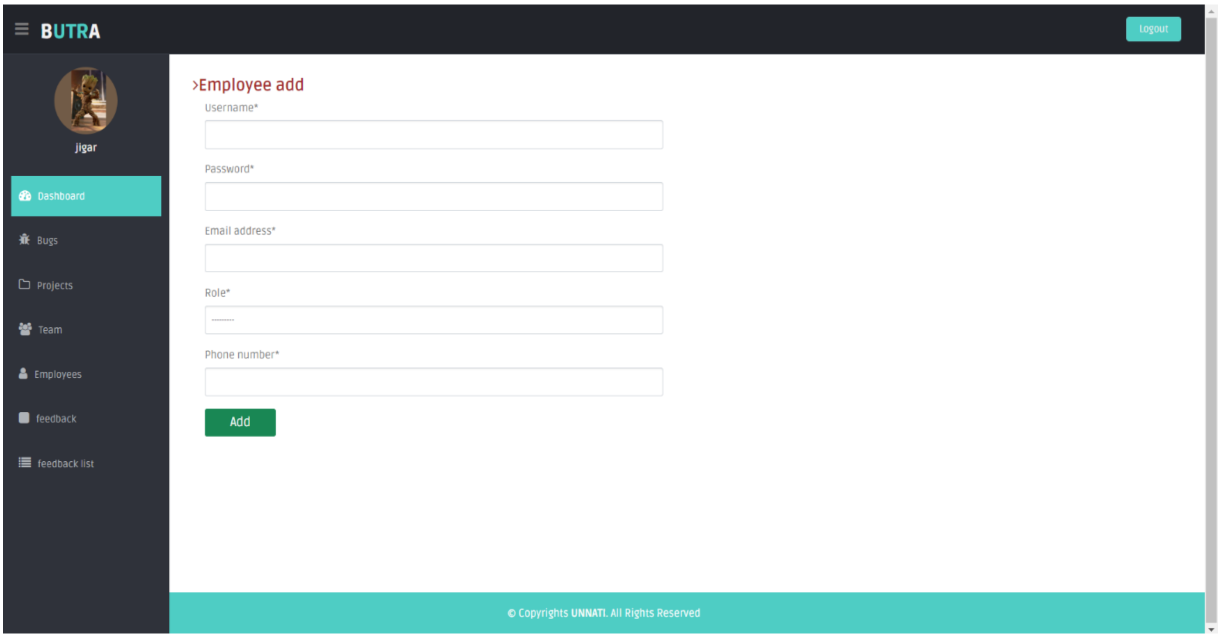
Table 6.7 feedback table

5.4 INPUT/ OUTPUT INTERFACE DESIGN

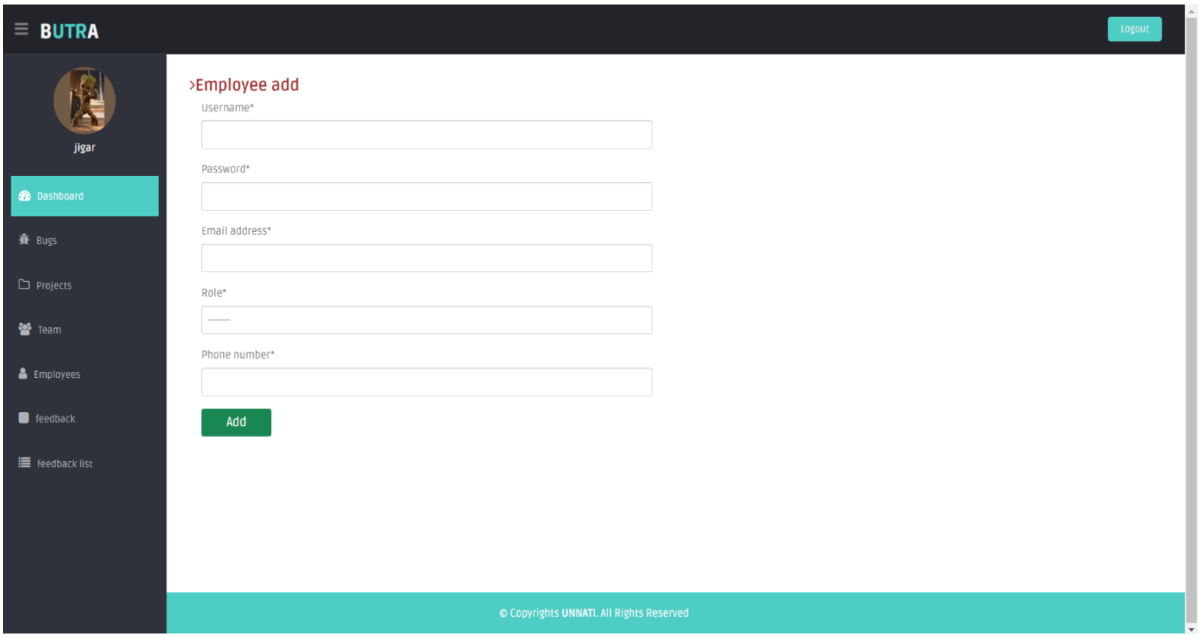
In this section, I had put some design screenshots of the Web application.



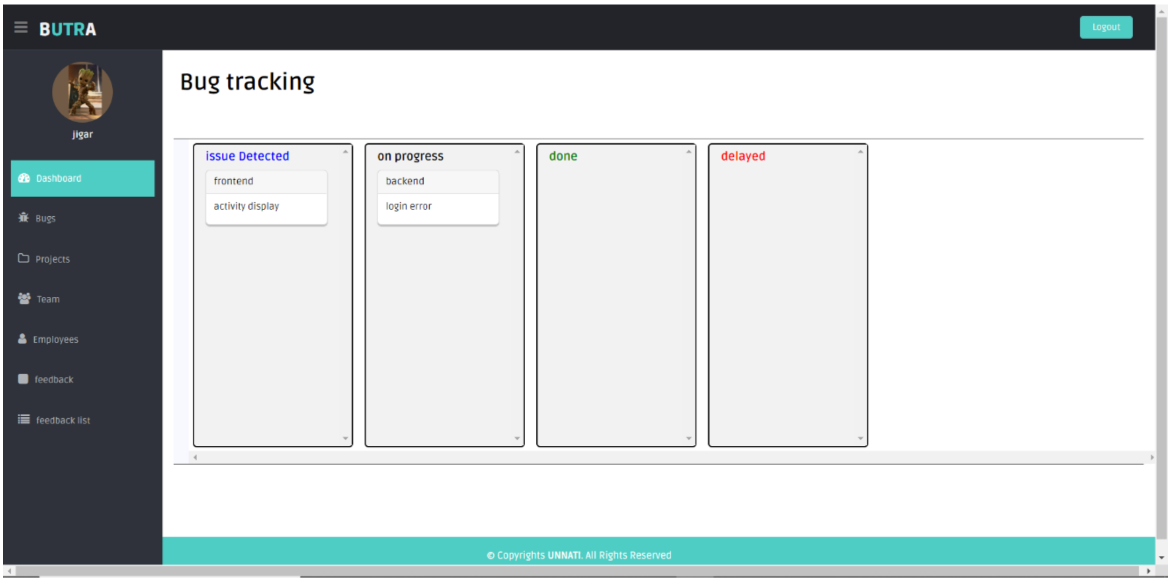
5.4.1 Login Screenshot



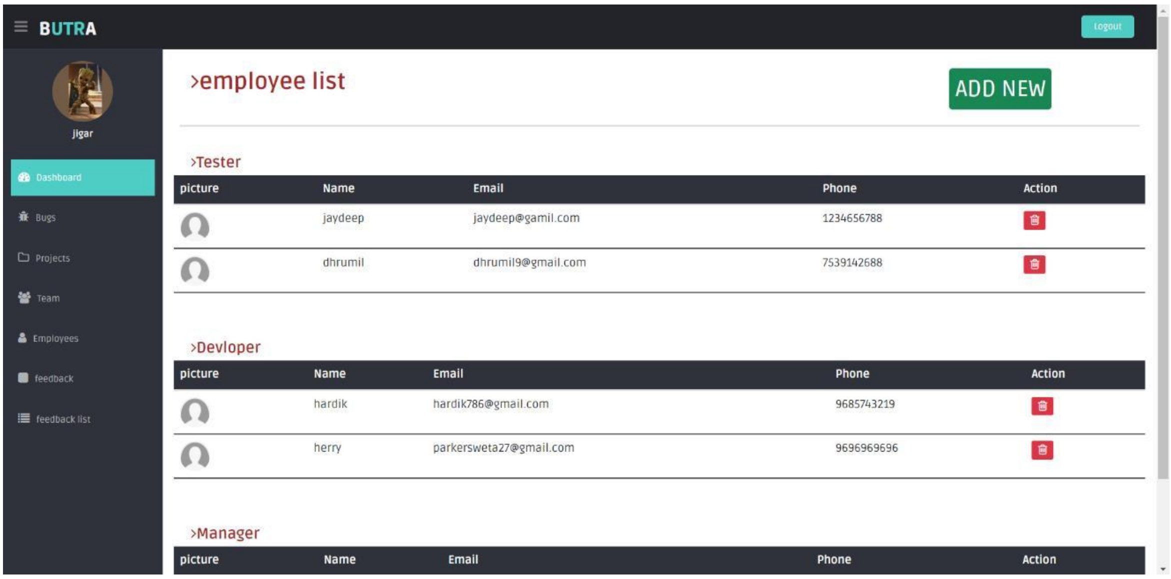
5.4.2 Register employee



5.4.3 Bug form



5.4.4 Dashboard



>employee list

>Tester

picture	Name	Email	Phone	Action
	jaydeep	jaydeep@gamil.com	1234656788	🗑
	dhrumil	dhrumil9@gmail.com	7539142688	🗑

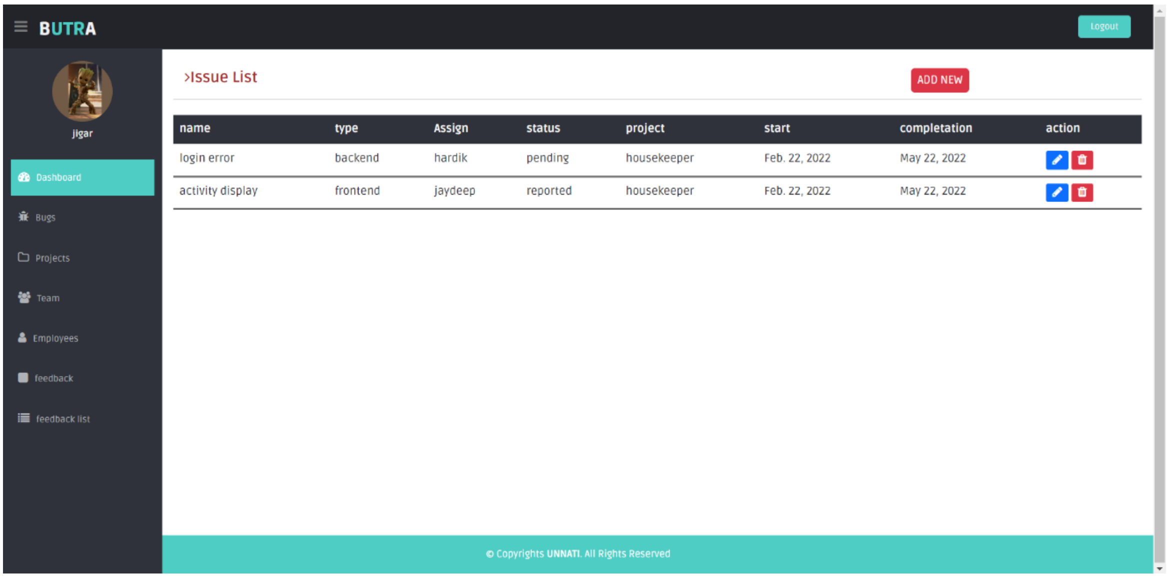
>Devloper

picture	Name	Email	Phone	Action
	hardik	hardik786@gmail.com	9685743219	🗑
	herry	parkersweta27@gmail.com	9696969696	🗑

>Manager

picture	Name	Email	Phone	Action

5.5.5 Employee list page



>Issue List

name	type	Assign	status	project	start	completation	action
login error	backend	hardik	pending	housekeeper	Feb. 22, 2022	May 22, 2022	✎ 🗑
activity display	frontend	jaydeep	reported	housekeeper	Feb. 22, 2022	May 22, 2022	✎ 🗑

© Copyrights UNNATI. All Rights Reserved

5.5.6 Bug detail page

CHAPTER 6 IMPLEMENTATION

6.1 IMPLEMENTATION PLATFORM

The project is completely implemented on vs code. Vs code uses development platforms such as Windows or macOS. The registration and request management are done on the Web Application provided by the server. All the process of Login, Registration, and Sending Requests is done on the Web Application.

6.2 PROCESS / PROGRAM / TECHNOLOGY / MODULES SPECIFICATION(S)

Specifications Of This Program and Model are as follows:

- Store And View the Data Through Employee Unique Id
- Data Security and Privacy
- Centralized Database Used for Employee Records
- Use Login & Registration
- All Employees Can Access Information

6.3 RESULT ANALYSIS

The main goal of this project is to learn to implement and develop experience. By implementing functionality alone I had learned many things like how to solve bugs.

6.4 SAMPLE CODING

6.4.1 login page

```
{% load static %}
{% load crispy_forms_tags %}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="keyword" content="bugs, todo, list, butra">
  <title>login</title>
  <link href="{% static 'img/favicon.png' %}" rel="icon">
  <link href="{% static 'img/apple-touch-icon.png' %}" rel="apple-touch-icon">
  <link href="{% static 'lib/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
  <link href="{% static 'lib/font-awesome/css/font-awesome.css' %}" rel="stylesheet" />
  <link href="{% static 'css/style.css' %}" rel="stylesheet">
  <link href="{% static 'css/style-responsive.css' %}" rel="stylesheet">
  <style>
    .logo{
      background-color: #000b26;
      margin: 0px;
      border-radius: 5px;
      justify-content: left;

    }
    b{
      color: aliceblue;
    }
    span{
      color: rgb(78,205,196);
    }
  </style>

</head>

<body>
  <div id="login-page">
    <div class="container" >
      <form class="form-login" method="POST" >
        {% csrf_token %}
        <center>
          <h1 class="logo"><b>B<span>UTR</span>A</b></h1>
        </center>
        <h2 class="form-login-heading">sign in now</h2>
        <div class="login-wrap">
          {{ form.username | as_crispy_field }}
          <br>
          {{ form.password | as_crispy_field }}
          <br>
          <button class="btn btn-theme btn-block"  type="submit"><i class="fa fa-lock"></i>
```

```
SIGN IN</button>
        </div>
      </form>
    </div>
  </div>
  <!-- js placed at the end of the document so the pages load faster -->
  <script src="{% static 'lib/jquery/jquery.min.js' %}"></script>
  <script src="{% static 'lib/bootstrap/js/bootstrap.min.js' %}"></script>
  <script type="text/javascript" src="{% static 'lib/jquery.backstretch.min.js' %}"></script>
  <script>
    $.backstretch("{% static 'img/bg.jpg' %}", {
      speed: 500
    });
  </script>
</body>
</html>
```

6.4.2 employee views.py

```
from re import template
from django.conf import settings
from django.http import HttpResponseRedirect, JsonResponse
from django.shortcuts import render,redirect
from django.urls import reverse
from django.views.generic import  ListView,UpdateView, DeleteView, FormView,
CreateView
from django.contrib.messages.views import SuccessMessageMixin
from django.contrib.auth.views import redirect_to_login
from django.contrib import messages
from .forms import *
from django.views import View, generic
from django.contrib.auth.views import LoginView, LogoutView
from .models import User
from django.core.mail import EmailMultiAlternatives
from django.template.loader import render_to_string
from django.utils.html import strip_tags

class BaseRegisterView(SuccessMessageMixin, FormView):

    form_class = UserForm
    template_name = 'employee/register.html'
    success_url ="/view"
```

```
  def form_valid(self, form):
        user = form.save()
        user.set_password(user.password)


        user.save()
        username= form.cleaned_data.get("username")
        password = form.cleaned_data.get("password")
        dict = {'username': username,'password': password}
        subject, from_email, to = 'login detail', settings.EMAIL_HOST_USER,
form.cleaned_data.get('email')
        html_content = render_to_string('employee_mail.html', dict) # render with dynamic
value
        text_content = strip_tags(html_content)
        msg = EmailMultiAlternatives(subject, text_content, from_email, [to])
        msg.attach_alternative(html_content, "text/html")
        msg.send()
        return super().form_valid(form)

    def get_success_message(self, cleaned_data):
        username = cleaned_data["username"]
        return username + " - User Created Successfully..!!"

class Userlogin(LoginView):
    model = User
    template_name = 'employee/login.html'
    success_url ="/home"

class logoutview(LogoutView):
    success_url ="/login"

class profileview(UpdateView):
    form_class = updateForm
    model = User
    template_name = "employee/profile.html"
    success_url ="/view"

class Listemployee(ListView):
    model = User
    users = model.objects.all()
    context_object_name = 'users'
    template_name = "employee/list_employee.html"
```

```
def Deleteemployee(request, pk):
    employee = User.objects.get(pk=pk)
    employee.delete()
    messages.success(request, "employee deleted successfully!")


return redirect('/view')

class employeeDetail(generic.DetailView):
    model = User
    context_object_name = 'employee'
    template_name = "employee/detail_employee.html"

def home(request):
    return render(request, 'home.html')


class Feedback(CreateView):
    model = Feedback
    form_class = FeedbackForm
    template_name = 'employee/feadback.html'
    success_url ="/view"


class Feedbacklist(ListView):
    model = Feedback
    context_object_name = 'feeds'
    template = 'employee/list_feedback.html'
```

6.4.3 home page

```
{% extends "base.html" %}
{% load static %}
{% block content %}
<style>
  .card{
    background-color: white;
    height: max-content;
    border-radius: 5px;
  }
</style>
```

```
<script>
  function allowDrop(ev) { ev.preventDefault(); }
  function drag(ev) { ev.dataTransfer.setData("text", ev.target.id);}
  function drop(ev) {
    ev.preventDefault();


var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));}
  </script>



<section id="main-content" style="background-color:white; color:black;">
  <section class="wrapper">
    <div class="container" style="margin: 25px 0px; ">
      <h1 style="font-weight: bolder; font-size: 30px;">Bug tracking</h1>
    </div>
    <div class="row mt inbox" style="margin: 50px 0px;">
      <div class="col-lg-18" id="box" style="background-color:white;">
        <div class="col-lg-2" id="scroll" ondrop="drop(event)"
ondragover="allowDrop(event)">
          <header class="head" style="color: blue;">issue Detected</header>

          {% for bug in bugs %}
          {% if bug.status.statusName == 'reported' %}
          <div class="card " draggable="true" ondragstart="drag(event)" id="drag1">
            <div class="card-header">
              {{bug.bugtype}}
            </div>
            <div class="card-body">
              <h5 class="card-title">{{bug.bugname}}</h5>
            </div>
          </div>
          {% endif %}
          {% endfor %}

        </div>
        <div class="col-lg-2" id="scroll" ondrop="drop(event)"
ondragover="allowDrop(event)">
          <header class="head" style="color: black;">on progress</header>

{% for bug in bugs %}
            {% if bug.status.statusName == 'pending'%}
    <div class="card" draggable="true" ondragstart="drag(event)" id="drag1">
            <div class="card-header" >
              {{bug.bugtype}}
            </div>
            <div class="card-body">
              <h5 class="card-title">{{bug.bugname}}</h>

    </div>
          </div>
          {% endif %}
          {% endfor %}

    </div>
      <div class="col-lg-2" id="scroll" ondrop="drop(event)"
ondragover="allowDrop(event)">
        <header class="head" style="color: green;">done</header>
        {% for bug in bugs %}
          {% if bug.status.statusName == 'done'%}
          <div class="card " draggable="true" ondragstart="drag(event)" id="drag1">
            <div class="card-header">
              {{bug.bugtype}}

            </div>
            <div class="card-body">
              <h5 class="card-title">{{bug.bugname}}</h5>

            </div>
          </div>
          {% endif %}
          {% endfor %}

        </div>
      <div class="col-lg-2" id="scroll" ondrop="drop(event)"
ondragover="allowDrop(event)">
        <header class="head" style="color: red;">delayed</header>

        {% for bug in bugs %}
          {% if bug.status.statusName == 'deley'%}
          <div class="card " draggable="true" ondragstart="drag(event)" id="drag1">
            <div class="card-header">
              {{bug.bugtype}}
            </div>
            <div class="card-body">
```

```
  <h5 class="card-title">{{bug.bugname}}</h5>
            </div>
          </div>
          {% endif %}
          {% endfor %}

</div>
      </div>
    </div>
</section>
</section>
{% endblock content %}
```