

VISVESVARAYA TECHNOLOGICAL UNIVERSITY BELAGAVI-590018



A DBMS Mini-Project Report
On

“NGO Database Management System”

Submitted in partial fulfilment of the requirements for the 5th semester of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi

Submitted by:

VIKHYAT VINOD KUMAR **1RN19CS180**

VARUNALINGAM P.M **1RN19CS176**

Under the Guidance of:

Mr. Karanam Sunil Kumar
Assistant Professor
Dept. of CSE

Dr. H R Shashidhara
Associate Professor
Dept. of CSE



Department of Computer Science and Engineering
RNS Institute of Technology Channasandra, Dr. Vishnuvardhan Road,
Bengaluru-560 098
2021-2022

RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road,

Bengaluru-560 098

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

Certified that the DBMS mini-project work entitled “**NGO Database Management System**” has been successfully carried out by **VIKHYAT VINOD KUMAR** bearing USN **1RN19CS180** and **VARUNALINGAM P.M** bearing USN **1RN19CS176**, bonafide students of **RNS Institute of Technology** in partial fulfilment of the requirements for the **5th semester Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the mini-project requirements of the DBMS lab of 5th semester BE in CSE.

Mr. Karanam Sunil Kumar
Assistant Professor
Dept. of CSE

Dr. H R Shashidhara
Associate Professor
Dept. of CSE

Dr. Kiran P
Prof. and Head
Dept of CSE

External Viva:

Name of the Examiners

Signature with Date

- 1.**
- 2.**

ACKNOWLEDGMENTS

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance , encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We are grateful to **Dr. M K Venkatesha**, Principal, RNSIT, Bangalore, for his support towards completing this mini project.

We would like to thank **Dr. Kiran P** Prof. & Head, Department of Computer Science & Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Mr. Karanam Sunil Kumar**, Assistant Professor, Dr. H R Shashidhara, Associate Professor, Department of CSE, RNSIT, Bangalore, for their able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of department of Computer Science & Engineering, RNSIT, Bengaluru for their constant support and encouragement.

Date : 24/01/2022

Vikhyat Vinod Kumar

1RN19CS180

Place : Bengaluru

Varunalingam P.M

1RN15CS176

CONTENTS

Chapter No.	Title	Page No
1.	Introduction	
1.1	Database technologies	1
1.2	Characteristics of database approach	2
1.3	Applications of DBMS	4
1.4	Problem description/ statement	4
2.	Requirements Analysis	
2.1	Hardware Requirements	5
2.2	Software Requirements	5
2.3	Functional Requirements	
2.3.1	Major Entities	5
2.3.2	End User Requirements	6
2.3.3	HTML/CSS	6
2.3.4	NodeJS	7
2.3.5	MySQL	8
2.3.6	ExpressJS	9
3.	Database Design	
3.1	Entities, Attributes and Relationships	11
3.2	Identify major entities/Attributes and relationships	11
3.3	ER Schema	14
3.4	Relational Schema	15
4.	Implementation	
4.1	Database connectivity	16
4.2	Pseudo code For Major Functionalities	17
5.	Results, snapshots and discussions	27
6.	Conclusion and Future Enhancements	31

CHAPTER 1

INTRODUCTION

1.1 DATABASE TECHNOLOGIES

The essential feature of database technology is that it provides an internal representation (model) of the external world of interest. Examples are, the representation of a particular date/time/flight/aircraft in an airline reservation or of the item code/item description/quantity on hand/reorder level/reorder quantity in a stock control system.

The technology involved is concerned primarily with maintaining the internal representation consistent with external reality; this involves the results of extensive R&D over the past 30 years in areas such as user requirements analysis, data modelling, process modelling, data integrity, concurrency, transactions, file organisation, indexing, rollback and recovery, persistent programming, object-orientation, logic programming, deductive database systems, active database systems... and in all these (and other) areas there remains much more to be done. The essential point is that database technology is a **CORE TECHNOLOGY** which has links to:

- Information management / processing
- Data analysis / statistics
- Data visualization / presentation
- Multimedia and hypermedia
- Office and document systems
- Business processes, workflow, CSCW (computer-supported cooperative work)

Relational DBMS is the modern base technology for many business applications. It offers flexibility and easy-to-use tools at the expense of ultimate performance. More recently relational systems have started extending their facilities in directions like information retrieval, object-orientation and deductive/active systems which lead to the so-called 'Extended Relational Systems'.

Information Retrieval Systems began with handling library catalogues and then

extended to full free-text by utilizing inverted index technology with a lexicon or thesaurus. Modern systems utilize some KBS (knowledge-based systems) techniques to improve the retrieval.

Object-Oriented DBMS started for engineering applications in which objects are complex, have versions and need to be treated as a complete entity. OODBMSs share many of the OOPL features such as identity, inheritance, late binding, overloading and overriding. OODBMSs have found favours in engineering and office systems but haven't been successful yet in traditional application areas.

Deductive / Active DBMS has evolved over the last 20 years and combines logic programming technology with database technology. This allows the database itself to react to the external events and also to maintain its integrity dynamically with respect to the real world.

1.2 CHARACTERISTICS OF DATABASE APPROACH

Traditional form included organising the data in file format. DBMS was a new concept then, and all kinds of research was done to make it overcome the deficiencies in traditional style of data management. A modern DBMS has the following characteristics

- Real-world entity – A modern DBMS is more realistic and uses real-world entities to design its architecture. It uses behaviour and attribute too. For example, a school database may use students as an entity and their age as an attribute.
- Relation-based tables – DBMS allows entities and relations to form tables.
A user can understand the architecture of a database by just looking at the table names.
- Isolation of data and application – A database system is entirely different than its data. A database is an active entity, whereas data is said to be passive, on which the database works and organizes. DBMS also stores metadata, which is data about data, to ease its own process.

- Less redundancy – DBMS follows the rules of normalization, which splits a relation when any of its attributes has redundancy in its values. Normalization is a mathematically rich and scientific process that will reduce the data redundancy.
- Consistency – Consistency is a state where every relation in a database remains consistent. There exist methods and techniques, that can detect an attempt of leaving database in an inconsistent state. DBMS can provide greater consistency as compared to earlier forms of data storing applications like file-processing systems.
- Query Language – DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and the filtering options as required to retrieve a set of data. Traditionally it was not possible where file-processing system was used.
- ACID Properties – DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (normally shortened as ACID). These concepts are applied on transactions, which manipulate data in a database. ACID properties help the database to stay healthy in multi-transactional environments and also in case of failure.
- Multiuser and Concurrent Access – DBMS supports multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when users attempt to handle the same data item, but users are always unaware of them.
- Multiple views – DBMS offers multiple views for different users. A user in the Sales department will have a different view of the database from the person working in the Production department. This feature enables the users to have a concentrate view of the database according to their requirements.
- Security – Features like multiple views offer security to certain extent when users are unable to access the data of other users and departments. DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage. DBMS offers many different levels of security features, which enables multiple users to have different views with different features. For example, a user in the Sales department cannot see the data that belongs to the Purchase department. It can also be helpful in deciding how much data of the Sales department should be displayed to the user. Since a DBMS is not saved on the disk as traditional file systems, it is very hard for miscreants to break the code.

1.3 APPLICATIONS OF DBMS

Applications of Database Management Systems :

- **Telecom:** There is a database to keeps track of the information regarding the calls made, network usage, customer details etc. Without the database system it is hard to maintain such huge amounts of data which gets updated every millisecond.
- **Industry:** Whether it is a manufacturing unit, a warehouse or a distribution centre, each one needs a database to keep the records of the ins and outs. For example, a distribution centre should keep a track of the product units that were supplied to the centre as well as the products that got delivered from the distribution centre on each day; this is where DBMS comes into picture.
- **Banking System:** For storing information regarding a customer, keeping a track of his/her day-to-day credit and debit transactions, generating bank statements etc is done with through Database management systems.
- **Education sector:** Database systems are frequently used in schools and colleges to store and retrieve the data regarding the student, staff details, course details, exam details, payroll data, attendance details, fees detail etc. There is lots of inter-related data that needs to be stored and retrieved in an efficient manner.
- **Online shopping:** You must be aware of the online shopping websites such as Amazon, Flip kart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

1.4 PROBLEM DESCRIPTION/STATEMENT

- Developing a platform for a Non-Government Organization (NGO) called “AIDEN”, which will facilitate multiple registered restaurants to provide food, that usually goes to waste, to the volunteers of the NGO, who will in turn perform a quality check, and provide this food to those who are less privileged.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 HARDWARE REQUIREMENTS

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor	:	Pentium4 processor
Processor Speed	:	2.4 GHz
RAM	:	1 GB
Storage Space	:	40 GB
Monitor Resolution	:	1024*768 or 336*768 or 1280*1024

2.2 SOFTWARE REQUIREMENTS

1. Operating System used: Windows 10
2. Brackets Text Editor: HTML, CSS, Java Script
3. Server: ExpressJS working on NodeJS framework
4. Database Used: MySQL 8.0
5. IDE used : Visual Studio Code
6. Browser that supports HTML, CSS and JS

2.3 FUNCTIONAL REQUIREMENTS

2.3.1 Major Entities

Restaurants: They are the main source for the supply of leftover food.

They login to our portal by creating an account by specifying restaurant name, join date, address, email and their phone number. They can choose the required hotspot where they want to deliver their food to.

Admins: They are the organisers for each hotspot and they make sure that the food is delivered to the underprivileged. They also do a quality check on the food to make sure that the food edible. They also track the orders and making sure of timely delivery of food.

2.3.2 End User Requirements

The technical requirements for the end user include a browser which supports HTML, CSS and basic JavaScript code.

2.3.3.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from a local storage and render them to multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects like interactive forms can be embedded into the rendered page. It provides a way to create structured documents by denoting structural semantics for the text like headings, paragraphs, lists, links, quotes and other items. HTML elements are delimited by tags that are written within angle brackets. Tags such as `` and `<input />` introduce content into the page directly. Other tags such as `<p>...</p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can also embed programs written in a scripting language such as JavaScript which affect the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content.

2.3.3.2 CSS

Cascading Style Sheets (CSS) is a style sheet language which is used for describing the presentation of a document written in a markup language. Although most often it used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is also applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a

cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share the formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

2.3.4 NODE JS

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project!

Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.

A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behaviour the exception rather than the norm.

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to write the server-side code in addition to the client-side code without the need to learn a completely different language.

In Node.js the new ECMAScript standards can be used without problems, as you don't have to wait for all your users to update their browsers - you are in charge of deciding which ECMAScript version to use by changing the Node.js version, and you can also enable specific experimental features by running Node.js with flags.

NodePackageManager(npm) with its simple structure helped the ecosystem of Node.js proliferate, and now the npm registry hosts over 1,000,000 open-source packages you can

freely use.

2.3.5 MySQL

MySQL is a Relational Database Management System (RDBMS). MySQL server can manage many databases at the same time. In fact, many people might have different databases managed by a single MySQL server. Each database consists of a structure to hold onto the data itself. A data-base can exist without data, only a structure, be totally empty, twiddling its thumbs and waiting for data to be stored in it.

Data in a database is stored in one or more tables. You must create the data-base and the tables before you can add any data to the database. First you create the empty database. Then you add empty tables to the database. Database tables are organized in rows and columns. Each row represents an entity in the database, such as a customer, a book, or a project. Each column contains an item of information about the entity, such as a customer name, a book name, or a project start date. The place where a particular row and column intersect, the individual cell of the table, is called a field. Tables in databases can be related. Often a row in one table is related to several rows in another table. For instance, you might have a database containing data about books you own. You would have a book table and an author table. One row in the author table might contain information about the author of several books in the book table. When tables are related, you include a column in one table to hold data that matches data in the column of another table.

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers. It is a second-generation Open-Source company that unites Open Source values and methodology with a successful business model.

- MySQL is a database management system. A database is a structured collection of data. It can be anything from a simple shopping list to a picture gallery or the vast amount of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQLstandard” to refer to the current version of the SQL Standard.

- MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations.

The MySQL Database Server is very fast, reliable, and easy to use.

MySQL Server was originally developed to handle large databases and has been successfully used in highly demanding production environments for several years. MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- MySQL Server works in a client/server or embedded systems. The MySQL Database Software is a client/server system which consists of a multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

2.3.6 EXPRESS JS

Express.js, or simply Express, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

The original author, TJ Holowaychuk, described it as a Sinatra-inspired server, meaning that it is relatively minimal with many features available as plugins. Express is the back-end

component of popular development stacks like the MEAN, MERN or MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library.

CHAPTER 3

DATABASE DESIGN

3.1 Entities, attributes and relationships

Admin: Staion_ID,password,location,emp_id

Users: Username,password,fname,lname,DOE,join_date,address,email,phone

Employee: Emp_id,emp_name,addhar_no;

Orders: order_id,suffices,date_of_order,food_type,status,user_name,location,date_of_issue

Ratings: order_id,rating

3.2 Major entities description

- USERS:**

```
mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| username   | varchar(25)   | NO   | PRI | NULL    |       |
| password   | varchar(256)  | NO   |     | NULL    |       |
| fname      | varchar(25)   | NO   |     | NULL    |       |
| lname      | varchar(25)   | YES  |     | NULL    |       |
| DOE        | date          | YES  |     | NULL    |       |
| join_date  | date          | YES  |     | NULL    |       |
| Address     | varchar(50)   | YES  |     | NULL    |       |
| email      | varchar(30)   | YES  |     | NULL    |       |
| phone      | varchar(10)   | NO   | UNI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.07 sec)
```

- **ADMIN:**

```
mysql> desc admin;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| station_ID | varchar(8) | NO   | PRI | NULL    |       |
| password   | varchar(256) | YES  |     | NULL    |       |
| location   | varchar(6)  | YES  | UNI | NULL    |       |
| emp_ID     | int(11)    | YES  | UNI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

- **ORDERS:**

```
mysql> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| order_ID   | int(11)    | NO   | PRI | NULL    |       |
| suffices   | int(11)    | YES  |     | NULL    |       |
| date_of_order | date      | YES  |     | NULL    |       |
| food_type  | varchar(25) | YES  |     | NULL    |       |
| status     | varchar(4)  | YES  |     | NULL    |       |
| user_name  | varchar(25) | YES  | MUL | NULL    |       |
| location   | varchar(6)  | YES  | MUL | NULL    |       |
| date_of_issue | date      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```


- **RATING:**

```
mysql> desc rating;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| order_ID   | int(11)   | YES  | UNI | NULL    |       |
| rating     | int(11)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.04 sec)
```

- **EMPLOYEE:**

```
mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_ID     | int(11)   | NO   | PRI | NULL    |       |
| emp_name   | varchar(20) | YES  |     | NULL    |       |
| aadhar_no  | varchar(12) | YES  | UNI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

3.3 ER Diagram

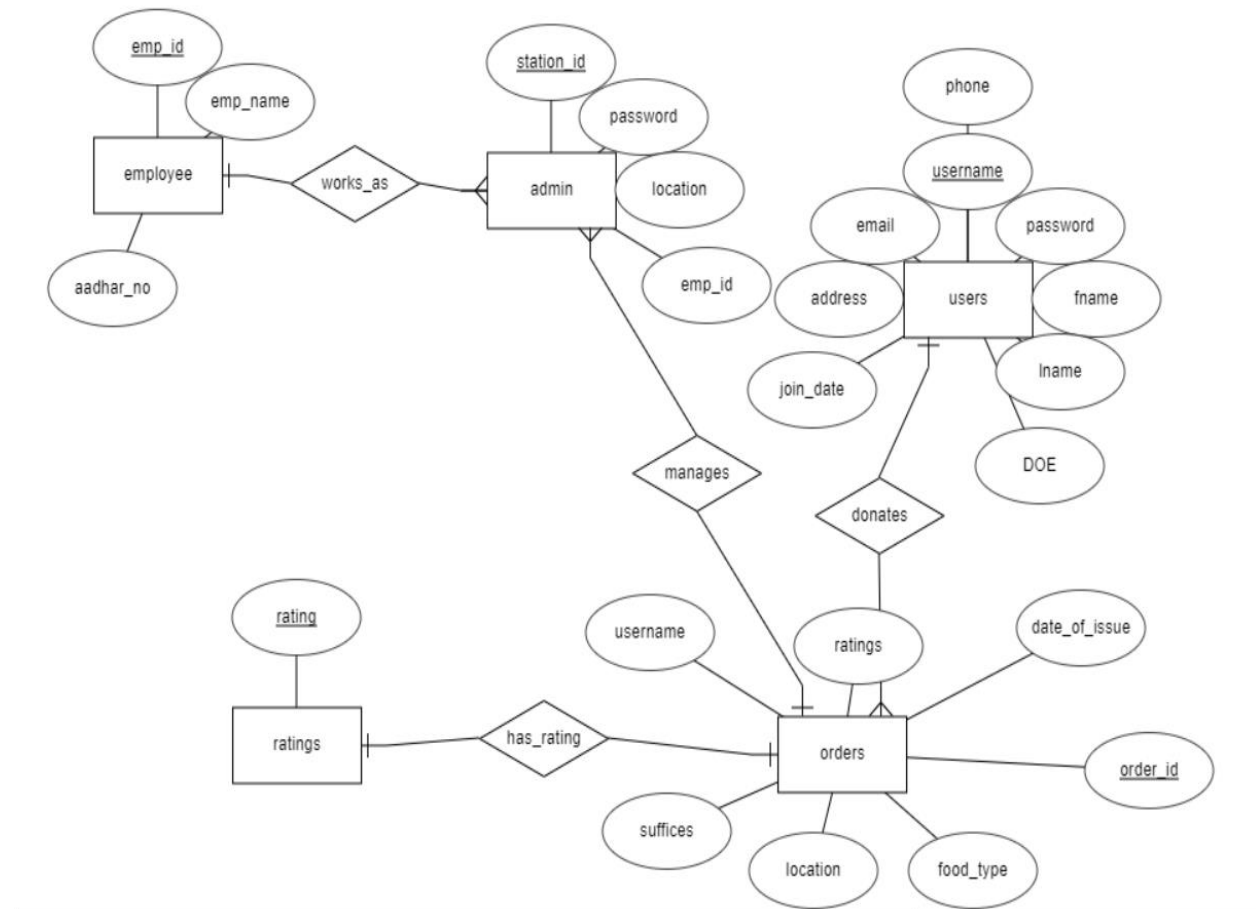


Fig.3.1. ER Diagram for AIDEN database management system

3.3 Relational Schema

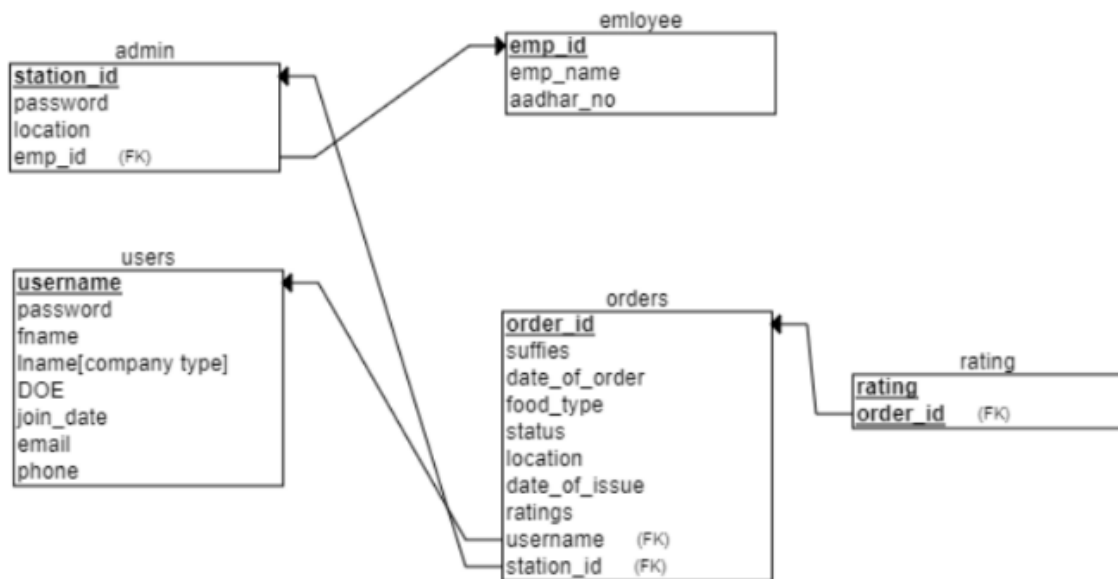


Fig.3.2. Schema for AIDEN database management system

CHAPTER 4

IMPLEMENTATION

4.1 Creating Database Connection

1. ExpressJS provides built-in database connectivity for a wide range of databases – MySQL, MongoDB, Scala, PostgreSQL etc.
2. We create a variable that holds the mysql connection.
3. The connection is created using the following code:

```
var connection = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: 'password',
  database: 'aiden',
  timezone: 'utc'
})
```

4. We use the variable holding the connection and establish the connection using .connect

```
connection.connect(function(error){
  if(!error){
    console.log("Error")
  }
  else{
    console.log("Successfully connected");
  }
})
```

5. SQL Operations

5.1. Selection query

```
connection.query('Select * from admin where station_ID="${req.body.usr_name}" and password="${pwd}"',function(err,result,fields){
  if(err){
    console.log("ERROR");
    res.send(err);
  }
})
```

5.2 Updation Query

```
connection.query(`update orders set status="${stat}" where order_ID=${ord_id}`,function(err,result,fields){
  if(err)
    console.log("ERROR");
});
```

5.3 Insertion Query

```
connection.query(`insert into rating values(${ord_id},${rating});`,function(err,result,fields){
  if(err)
    console.log("ERROR");
});
```

6. Closing the server:

If the server is running on a terminal, press CTRL+C. If the server is running in the background as a process, determine the process ID, and send a SIGINT command to that process.

4.2 Pseudo Code For Major Functionalities

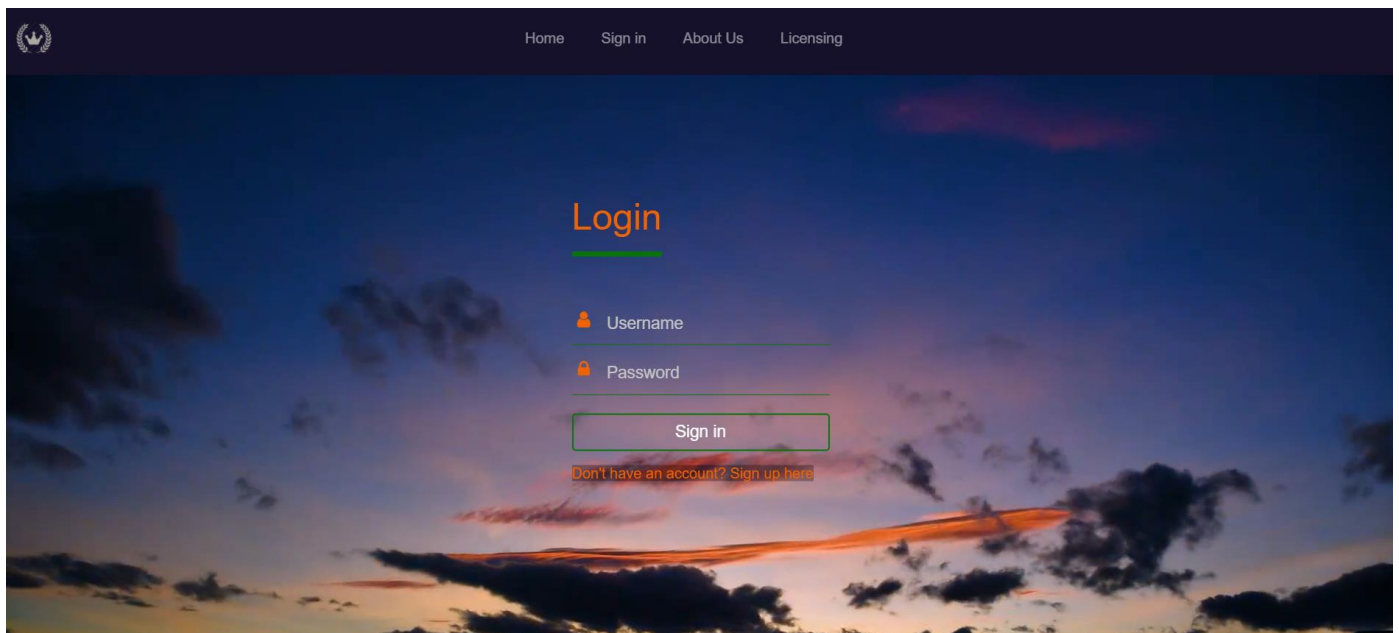
4.2.1 Login Page:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
9    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
10      integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
11    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIikvYIK3UEHnZm7KCKR/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5Kkkl" crossorigin="anonymous"></script>
12    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
13      integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
14    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
15      integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/3QGiRRSQQxSfFFWpi1MQuVdAyjUar5+76PVCmY1" crossorigin="anonymous"></script>
16    <link rel="stylesheet" href="..css/style_sign.css">
17  </head>
18  <body>
19    <nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark" id="navigation">
20      <a class="navbar-brand" href="#"></a>
21      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle">
22        <span class="navbar-toggler-icon"></span>
23      </button>
24      <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
25        <div class="navbar-nav" align="right">
26          <a class="nav-item nav-link" href=".." id="home">Home
27          <a class="nav-item nav-link" href="/signin" id="SignIn">Sign in</a>
28          <a class="nav-item nav-link" href=".." id="About">About Us</a>
29          <a class="nav-item nav-link" href="..#AboutUs" id="Licensing">Licensing</a>
30        </div>
20
```

```
31 </nav>
32 <container class="mainvid">
33   <video loop height="75%" width="100%" muted autoplay>
34     <source id="vid" src="/video/pexels-carlos-10161904.mp4" type="video/webm">
35   </video>
36 </container>
37 <div class="login-box">
38   <h1>Login</h1>
39   <form method="post" action="/signIn/submit">
40     <div class="textbox">
41       <i class="fa fa-user"></i>
42       <input type="text" placeholder="Username" id="usr_name" name="usr_name">
43     </div>
44
45     <div class="textbox">
46       <i class="fa fa-lock"></i>
47       <input type="password" placeholder="Password" id="pwd" name="pwd">
48     </div>
49     <input type="submit" class="btn" value="Sign in" id="SignInButton" >
50   </form>
51   <div><a href="/create" id="create" style="color: rgb(241, 100, 6);">Don't have an account? Sign up here</a></div>
52 </div>
53 <script src="/js/signIn.js"></script>
54 </body>
55 </html>
```

Server-Side JavaScript Code:

```
365 app.get('/signIn',function(req,res){
366   if(CURRENT_USER!=undefined&&(CURRENT_USER!='BLRIDN01' || CURRENT_USER!='BLRVJN01' || CURRENT_USER!='BLRRRN01' || CURRENT_USER!='BLRKOR01' || CURRENT_USER!='BLRJAY01' ||
367     CURRENT_USER!='BLRJPN01' || CURRENT_USER!='BLRMAJ01' || CURRENT_USER!='BLRYSH01'))
368     res.redirect('/signedIn_Home')
369   else
370   {
371     console.log("IN Sign IN!!"+"SOS");
372     res.sendFile(path.join(__dirname + "/public/Sign/signin.html"));
373   }
374 })
```



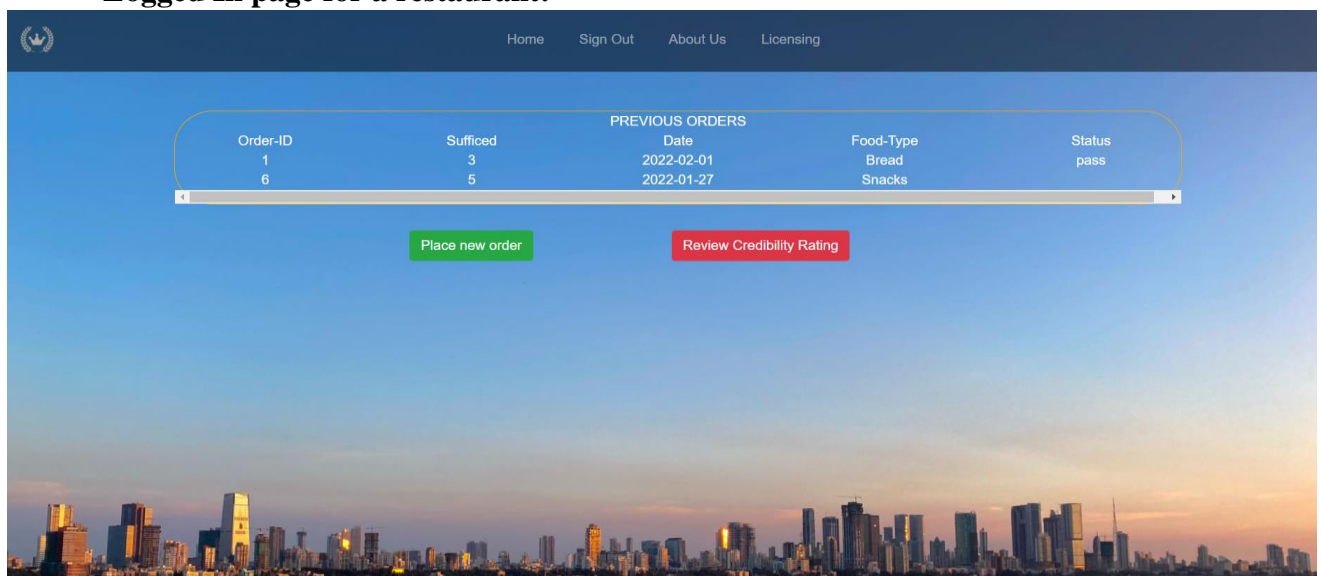
The Login Page as it appears for the user.


```

406     <a class="navbar-brand" href="#"></img></a>
407     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
408       <span class="navbar-toggler-icon"></span>
409     </button>
410     <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
411       <div class="navbar-nav" align="right">
412         <a class="nav-item nav-link" href=".." id="home">Home</a>
413         <a class="nav-item nav-link" href="/signOut" id="SignIn">Sign Out</a>
414         <a class="nav-item nav-link" href=".." id="About">About Us</a>
415         <a class="nav-item nav-link" href=".." id="AboutUs" id="Licensing">Licensing</a>
416       </div>
417     </div>
418   </nav>
419   <div id="result_orders" style="color:white" align="center">${JSON.stringify(result)}</div>
420   <div class="table" id="order_table" style="color:white" align="center">
421     <div class="row" align="center">
422       <div class="col" >PREVIOUS ORDERS</div>
423     </div>
424     <div class="row" align="Center">
425       <div class="col" id="Order_ID">Order-ID</div>
426       <div class="col" id="Sufficed">Sufficed</div>
427       <div class="col" id="Date">Date</div>
428       <div class="col" id="Type">Food-Type</div>
429       <div class="col" id="Status">Status</div>
430     </div>
431   </div>
432   <div>
433     <a type="button" href=".." id="nol"><button type="button" id="New_Order" name="New_Orders" class="btn btn-success" value="NewOrders" style="color:white">
434       Place new order</button></a>
435     <a type="button" href=".." id="rev"><button type="button" id="Rating" name="Rating" class="btn btn-danger" value="Rating" style="color:white" >
436       Review Credibility Rating</button></a>
437   </div>
438
439
440   <script src="/js/signedIn.js" type="module"></script>
441 </body>
442 </html>
443 }
444 });
445 //res.sendFile(__dirname+"/public/Sign/signedIn.html")
446
447 })

```

Logged In page for a restaurant:



4.2.3 Login process for Station Administrators:

Checking if Admin credentials are valid:

```

55 app.post('/signIn_Admin/submit',function(req,res){
56     var pwd=SHA256(req.body.pwd)
57     console.log(pwd)
58     console.log("IN ADMIN SUBMIT!")
59     connection.query('Select * from admin where station_ID="${req.body.usr_name}" and password="${pwd}" ;',function(err,result,fields){
60         if(err){
61             console.log("ERROR");
62             res.send(err);
63         }
64         else if((result[0])==undefined){
65             console.log("IN UNDEF");
66             res.redirect("/signIn_Admin/invalid_pwd");
67         }
68         else{
69             CURRENT_USER=req.body.usr_name;
70             console.log("IN ELSE SUBMIT ADMIN"+CURRENT_USER)
71             res.redirect("/signedIn_Admin");
72         }
73     });
74 })

```

Server-side Dynamically Created HTML Page:

```

75 app.get('/signedIn_Admin',function(req,res){
76     if(CURRENT_USER==undefined){
77         if(CURRENT_USER!='BLRIDN01'&&CURRENT_USER!='BLRVJN01'&&CURRENT_USER!='BLRRRN01'&&CURRENT_USER!='BLRKOR01'&&CURRENT_USER!='BLRJAY01'&&CURRENT_USER!='BLRJPN01'&&CURRENT_USER!='BLRMAJ01'&&CURRENT_USER!='BLRYSH01')
78             res.redirect('/signIn_Admin')
79         else{
80             connection.query('select o.*,emp_name from orders o,employee e,admin a where status is null and a.station_ID='${CURRENT_USER}' and o.location=a.location and e.emp_ID=a.emp_ID;',
81             function(err,result,fields){
82                 if(err){
83                     console.log("ERROR");
84                     res.send(err);
85                 }
86                 else if(result[0]==undefined){
87                     connection.query('select emp_name from employee e,admin a where a.station_ID='${CURRENT_USER}' and a.emp_ID=e.emp_ID',function(err,result1,fields){
88                         console.log(result1);
89                         orders=result;
90                         module.exports = orders;
91                         res.send('<!DOCTYPE html>
92                             <html lang="en">
93                             <head>
94                                 <meta charset="UTF-8">
95                                 <meta http-equiv="X-UA-Compatible" content="IE=edge">
96                                 <meta name="viewport" content="width=device-width, initial-scale=1.0">
97                                 <title>Document</title>
98                                 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
99                                 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoKWA+058RXPxPg6fy4IwvTlnhQE263XmFc31S/
100 crossorigin="anonymous">
101                                 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-K33o2DKtIkvYIK3UENzmM7KCKR/rnE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonym
102                                 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfi
103 crossorigin="anonymous"></script>

```

```

104 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFFwpi1MquVdAyjUar5+76PVCmY1"
105 crossorigin="anonymous"></script>
106 <link rel="stylesheet" href="css/style_sign.css"></link>
107 </head>
108 <body>
109 <nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark " id="navigation">
110 <a class="navbar-brand" href="#"></img></a>
111 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false"
112 aria-label="Toggle navigation">
113 <span class="navbar-toggler-icon"></span>
114 </button>
115 <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
116 <div class="navbar-nav align="right">
117 <a class="nav-item nav-link" href=".." id="home">Home</a>
118 <a class="nav-item nav-link" href="/signIn" id="SignIn">Sign Out</a>
119 <a class="nav-item nav-link" href=".." id="About">About Us</a>
120 <a class="nav-item nav-link" href=".." id="AboutUs" id="Licensing">Licensing</a>
121 </div>
122 </div>
123 <div class="social text-gray" id="greeting">$
124 </div>
125 </nav>
126 <div id="result_orders" style="color:white" align="center">${JSON.stringify(result1)}</div>
127 <form method="post" action="/signIn_Admin/ratings_submission">
128 <div class="table" id="order_table" style="color:white" align="center">
129 <div class="row" align="center">
130 <div class="col" >PENDING ORDERS</div>
131 </div>
132 <div class="row" align="Center">
133 <div class="col" id="Order_ID">Order-ID</div>
134 <div class="col" id="Sufficed">Sufficed</div>
135 <div class="col" id="Date">Date</div>
136 <div class="col" id="Type">Food-Type</div>
137 <div class="col" id="Status">Provide Rating</div>
138 <div class="col" id="Lock">Lock Rating</div>
139 </div>
140 <br><br>

```

```

141 </div>
142 <div class="row" style="margin-top: 3em" id="func_div">
143 <div class="col" style="text-align: center"><button type="submit" id="Ratings_Sub" name="Ratings_Sub" align="center" class="btn btn-success"
144 value="Ratings_Sub" style="color:white">Submit Locked Ratings</button></div>
145 </div>
146 </form>
147
148 <script src="/js/signInAdmin.js" type="module"></script>
149 </script>
150 </body>
151 </html>` )
152 ))
153 }
154 else
155 {
156 console.log(result);
157 orders=result;
158 module.exports = orders;
159 res.send(`<!DOCTYPE html>
160 <html lang="en">
161 <head>
162 <meta charset="UTF-8">
163 <meta http-equiv="X-UA-Compatible" content="IE=edge">
164 <meta name="viewport" content="width=device-width, initial-scale=1.0">
165 <title>Document</title>
166 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
167 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWA+958RXPxPg6fy4IWvTNh0E263XmfcJ1S
168 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UEJNzmM7KCkRn/rE9/Qp6GaAZGjwFDMVNA/GpGFF93hXpG5KkN"
169 crossorigin="anonymous"></script>
170 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
171 integrity="sha384-ApNbgh9B+Y1QKt3Rn7W3mgPxmU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
172 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
173 integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFFwpi1MquVdAyjUar5+76PVCmY1" crossorigin="anonymous"></script>
174 <link rel="stylesheet" href="css/style_sign.css"></link>
175 </head>

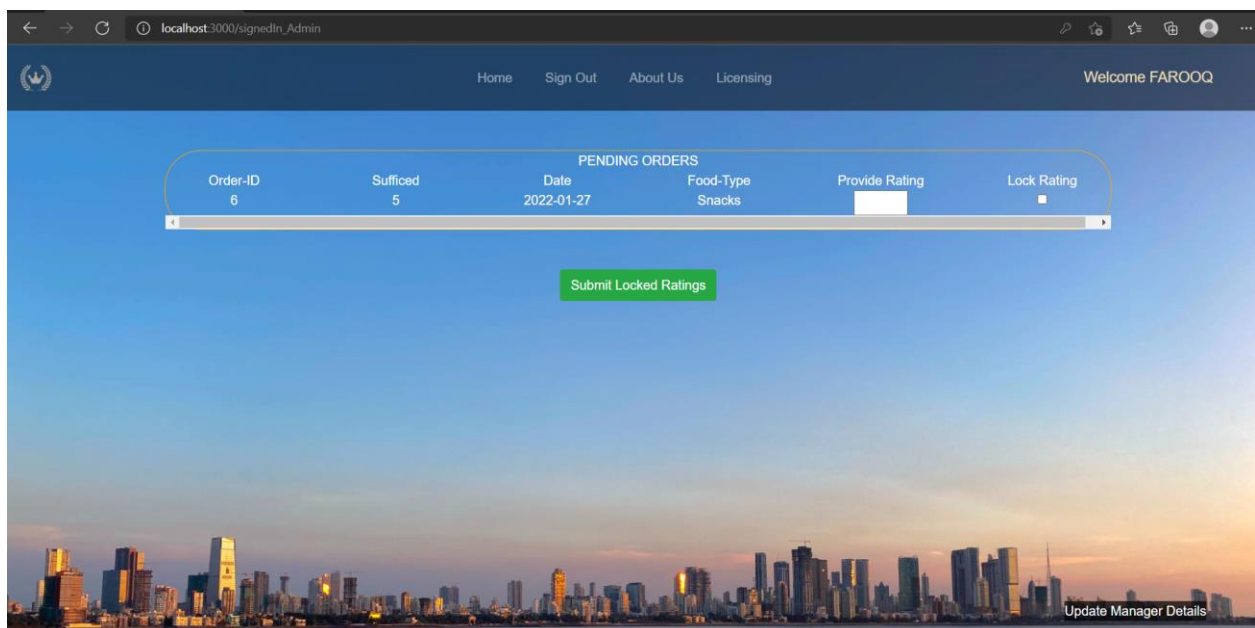
```

```

176 <body>
177 <nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark " id="navigation">
178 <a class="navbar-brand" href="#"></img></a>
179 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-expanded="false"
180 aria-label="Toggle navigation">
181 <span class="navbar-toggler-icon"></span>
182 </button>
183 <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
184 <div class="navbar-nav" align="right">
185 <a class="nav-item nav-link" href="#" id="home">Home</a>
186 <a class="nav-item nav-link" href="/signIn" id="SignIn">Sign Out</a>
187 <a class="nav-item nav-link" href="#" id="About">About Us</a>
188 <a class="nav-item nav-link" href="#" id="AboutUs" id="Licensing">Licensing</a>
189 </div>
190 </div>
191 <div class="social text-gray" id="greeting">
192 </div>
193 </nav>
194 <div id="result_orders" style="color:white" align="center">${JSON.stringify(result)}</div>
195 <form method="post" action="/signIn_Admin/ratings_submission">
196 <div class="table" id="order_table" style="color:white" align="center">
197 <div class="row" align="center">
198 <div class="col" >PENDING ORDERS</div>
199 </div>
200 <div class="row" align="Center">
201 <div class="col" id="Order_ID">Order-ID</div>
202 <div class="col" id="Sufficed">Sufficed</div>
203 <div class="col" id="Date">Date</div>
204 <div class="col" id="Type">Food-Type</div>
205 <div class="col" id="Status">Provide Rating</div>
206 <div class="col" id="Lock">Lock Rating</div>
207 </div>
208 </div>
209 <div class="row" id="adm_func">
210 <div class="col" style="color:white;text-align:center;margin-top:2em;padding:1em;" >
211 <button type="submit" id="Ratings_Sub" name="Ratings_Sub" class="btn btn-success" value="Ratings_Sub" style="color:white">Submit Locked Ratings</button>
212 </div>
213 </div>
214 </form>
215
216 <script src="/js/signInAdmin.js" type="module"></script>
217 </script>
218 </body>
219 </html>]]
220 }
221 });
222 }
223 })

```

Logged in page for station admin



4.2.3 Process for placing order by restaurant:

HTML Code for Order Submission Page:

```

public > Sign > orders.html > html > head > title
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Orders</title>
8   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
9   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWA+058RXPxPg6fy4IwvTnH0E263Xmf cJ1SAwiGgFAM/dAiS6JXk"
10  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UEIznm7CKCR/rE9/Qp6aaZGjwFDMVNA/gpGFF93hXpG5Kd" crossorigin="anonymous"></script>
11  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="
12  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZr66Pejzh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFFWpi1MuqVdAyjUar5+76PVCmYl" crossorigin="an
13  <!-- <link rel="stylesheet" href="style.css"> -->
14  <link rel="stylesheet" href="../css/style_sign.css">
15 </head>
16 <body>
17 <nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark" id="navigation">
18   <a class="navbar-brand" href="#"></a>
19   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle nav
20     <span class="navbar-toggler-icon"></span>
21   </button>
22   <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
23     <div class="navbar-nav" align="right">
24       <a class="nav-item nav-link" href="#" id="home">Home
25       <a class="nav-item nav-link" href="/signOut" id="SignIn">Sign Out</a>
26       <a class="nav-item nav-link" href="#" id="About">About Us</a>
27       <a class="nav-item nav-link" href="#" id="AboutUs">Licensing</a>
28     </div>
29   </nav>
30   <div id="form">
31     <form action="/order_submission" method="post">
32
33     </form>
34     <div class="order-box">
35       Enter details to place order
36       <br><br>
37       <form method="post" action="/order_submission">
38         <div>
39           <label for="suf">Surfaces<br><p style="font-size:x-small;">(No. of people it will feed)</p></label>
40           <input id="suf" type="number" name="num" min="3" max="100">
41           <label for="order_date">Date of order<p style="font-size:x-small;">(Date of donation)</p></label>
42           <input id="order_date" type="date" name="date">
43           <label for="food">Food Type</label>
44           <select id="food" name="food">
45             <option value="Bread">Bread</option>
46             <option value="Curry">Curry</option>
47             <option value="Rice">Rice</option>
48             <option value="Rasam">Rasam</option>
49             <option value="Snacks">Snacks</option>
50           </select>
51           <label for="location">Drop off station</label>
52           <select id="location" name="location">
53             <option value="BLRIND">Bangalore- Indiranagar</option>
54             <option value="BLRVJN">Bangalore- Vijayanagar</option>
55             <option value="BLRRRN">Bangalore- RR Nagar</option>
56             <option value="BLRJAY">Bangalore- Jayanagar</option>
57             <option value="BLRMJ">Bangalore- Majestic</option>
58             <option value="BLRYSH">Bangalore- Yeshwanthpur</option>
59             <option value="BLRJPN">Bangalore- JP Nagar</option>
60             <option value="BLRKOR">Bangalore- Koramangala</option>
61           </select>
62         </div>
63         <input type="submit" class="btn" value="Place Order" id="SignInButton">
64       </form>
65     </div>
66   </div>
67   <script src="../js/orders.js"></script>
68 </body>
69 </html>

```

Order Submission Page for the restaurant:

Enter details to place order

Suffixes
(Min. no. people to add food) 5

Date of order
(Date of delivery) 29-01-2022

Food Type Rice

Drop off station
Bangalore- Vijayanagar

Place Order

Server-Side Dynamically Created HTML Page for processing order submission request.

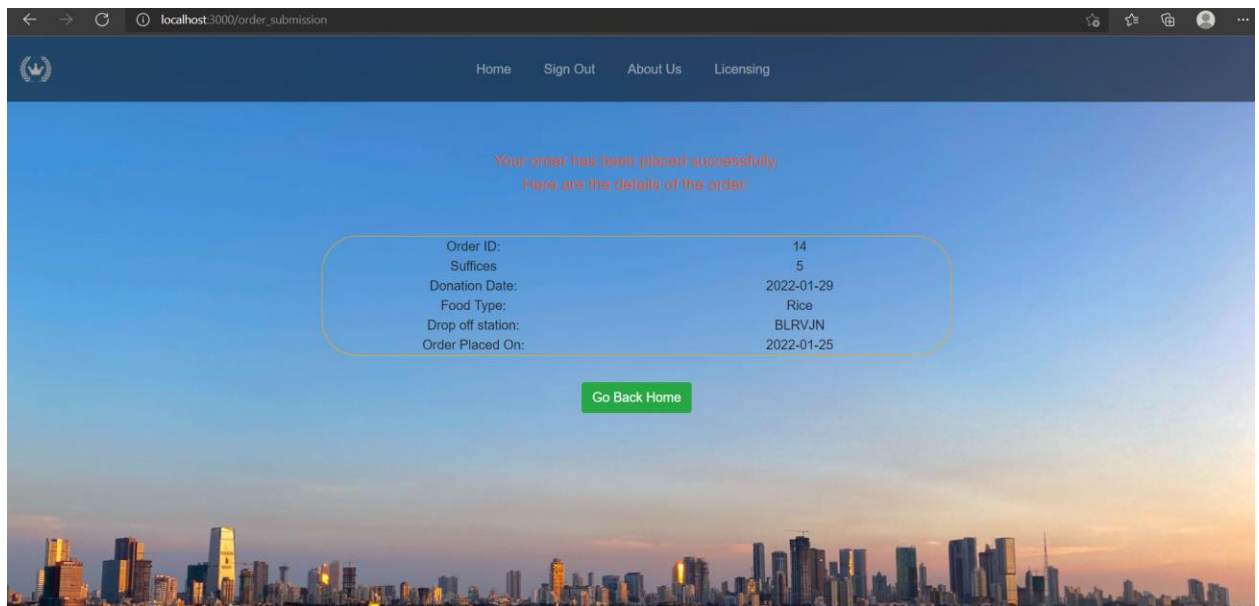
```

527 app.post("/order_submission",function(req,res){
528     var today = new Date();
529     var dd = String(today.getDate()).padStart(2, '0');
530     var mm = String(today.getMonth() + 1).padStart(2, '0');
531     var yyyy = (today.getFullYear()).toString();
532
533     today = yyyy + '-' + mm + '-' + dd;
534     var q='insert into orders values(${o_count+1},${req.body.num},${req.body.date},${req.body.food},null,${CURRENT_USER},${req.body.location},${today});'
535     console.log(q)
536     var send={};
537     send['oid']=o_count+1;
538     send['suff']=req.body.num;
539     send['order_date']=req.body.date;
540     send['food']=req.body.food;
541     send['location']=req.body.location;
542     send['origin_date']=today;
543     console.log(send)
544     connection.query(q,function(err,result,fields){
545         if(err){
546             console.log("ERROR dsfgdfg");
547             res.send(err);
548         }
549         else{
550             o_count++;
551             res.send('
552             <!DOCTYPE html>
553             <html lang="en">
554             <head>
555                 <meta charset="UTF-8">
556                 <meta http-equiv="X-UA-Compatible" content="IE=edge">
557                 <meta name="viewport" content="width=device-width, initial-scale=1.0">
558                 <title>Document</title>
559                 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
560                 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
561                 integrity="sha384-Gn5384xqQla0Nx+058RXPxPg6fy4IWvTNh0E263XmFcJISAwIGGfAM/dA1S6JXm" crossorigin="anonymous">
562                 <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCRN/rE9/QpG6aZG3wFDMVIA/GpGFF93hXpG5Kk4I"
563                 crossorigin="anonymous"></script>
564                 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
565                 integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
566                 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
567                 crossorigin="anonymous"></script>
568                 <link rel="stylesheet" href="/css/style_sign.css"></link>
569             </head>
570             <body>
571                 <nav class="navbar sticky-top navbar-expand-lg navbar-dark bg-dark " id="navigation">
572                     <a class="navbar-brand" href="#"></a>
573                     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false"
574                     aria-label="Toggle navigation">
575                         <span class="navbar-toggler-icon"></span>
576                     </button>
577                     <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
578                         <div class="navbar-nav" align="right">
579                             <a class="nav-item nav-link" href="#" id="home">Home
580                             <a class="nav-item nav-link" href="/signOut" id="SignOut">Sign Out</a>
581                             <a class="nav-item nav-link" href="#" id="About">About Us</a>
582                             <a class="nav-item nav-link" href="#" id="AboutUs">Licensing</a>
583                         </div>

```

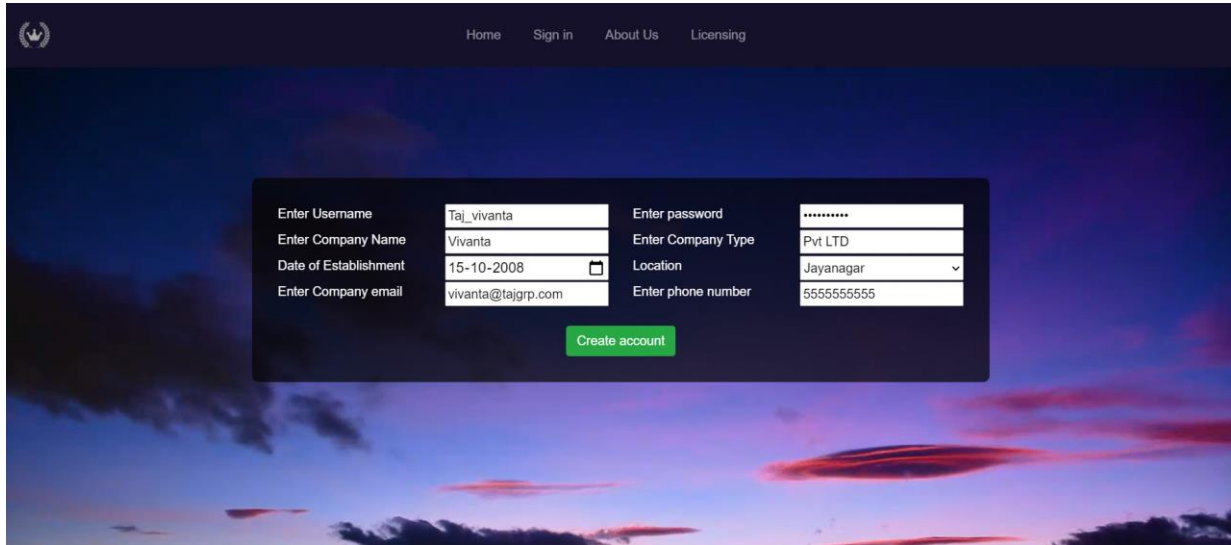
```
584     </nav>
585     <div id="message">Your order has been placed successfully.<br>Here are the details of the order:</div>
586     <div id="success" >${JSON.stringify(send)}</div>
587     <script src="/js/success.js">
588     </script>
589 </body>
590 </html>
591 </script>
592 </script>
593 </script>
594 </script>
```

Order Submission Confirmation Page:



CHAPTER 5

RESULTS , SNAPSHOTS AND DISCUSSIONS

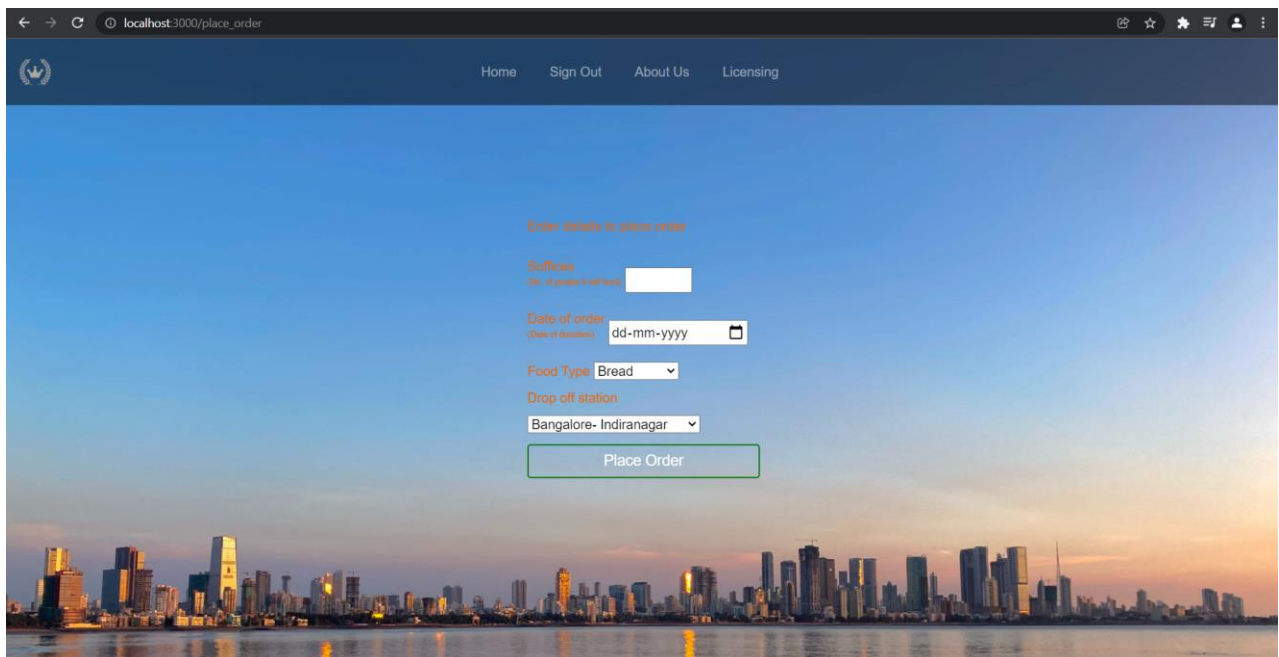


Enter Username	Taj_vivanta	Enter password	*****
Enter Company Name	Vivanta	Enter Company Type	Pvt LTD
Date of Establishment	15-10-2008	Location	Jayanagar
Enter Company email	vivanta@tajgrp.com	Enter phone number	5555555555

Create account

The Create Account Page for Restaurants

Once the account is created, the restaurants can login and proceed with placing donation orders.



Order details to place order

Suffixes
(You can provide a suffix)

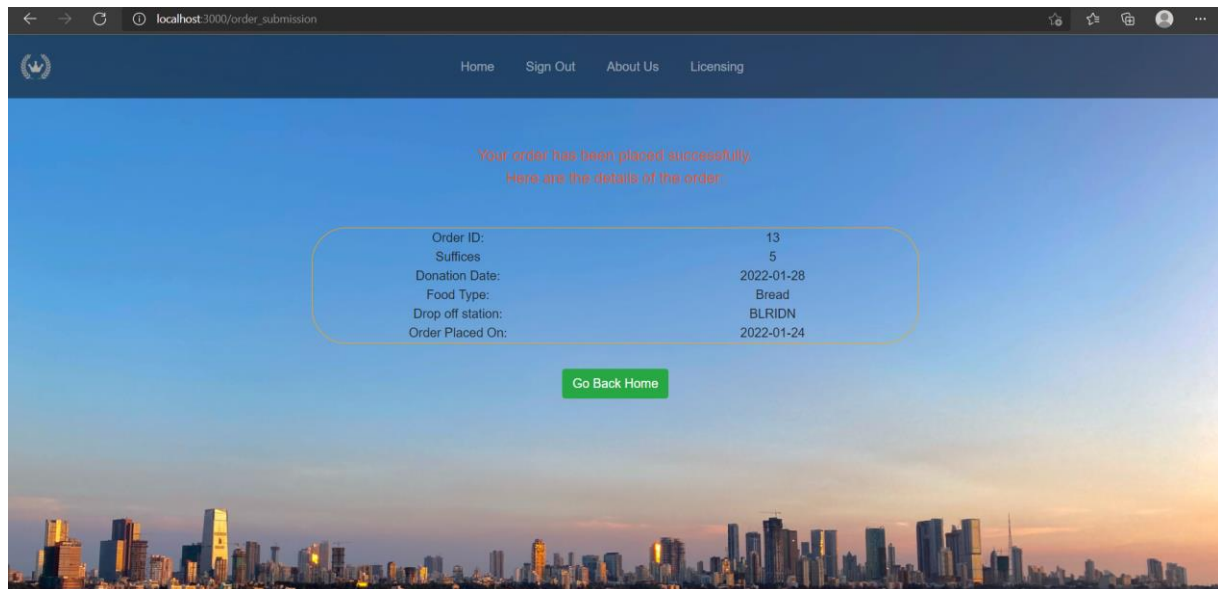
Date of order
(Date of donation)

Food Type Bread

Drop off station
Bangalore- Indiranagar

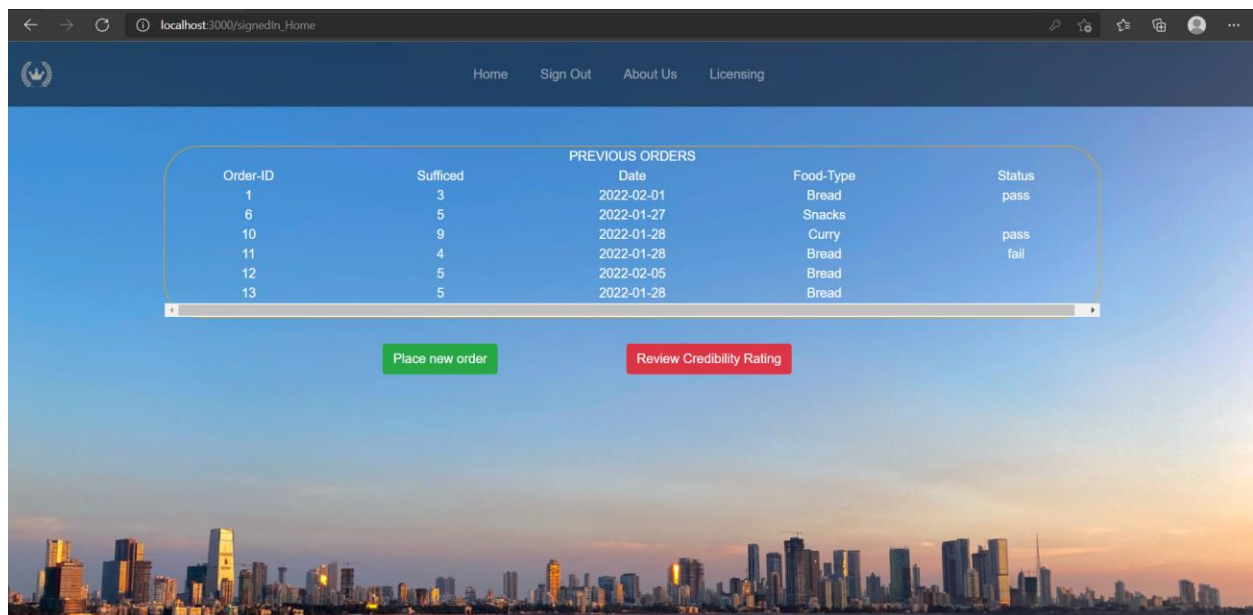
Place Order

The 'Place Orders' Page for a restaurant to place a donation order

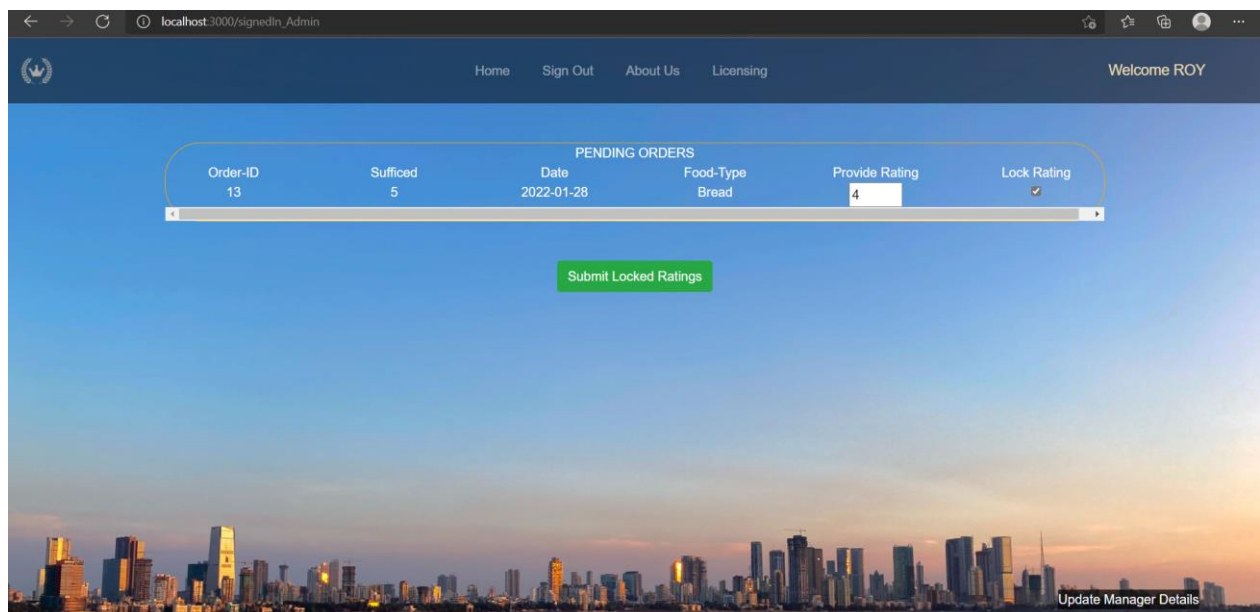


Order Submission confirmation

Once the order has been placed, the restaurants must await the review rating result from the manager of the station at which they dropped off the food donation. Once the admin of the station gives that particular order a rating, it gets reflected under 'status' field pertaining to that particular order.



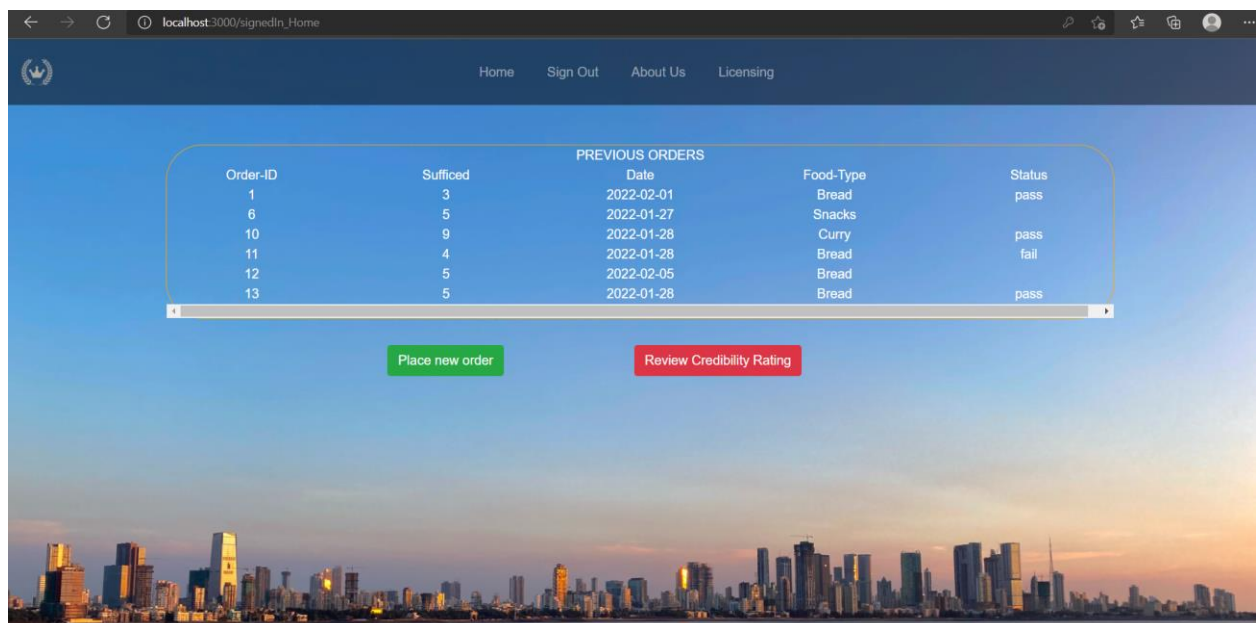
Observe that some orders do not have a Status. This means that it is pending, and the station admin has yet to rate those orders.



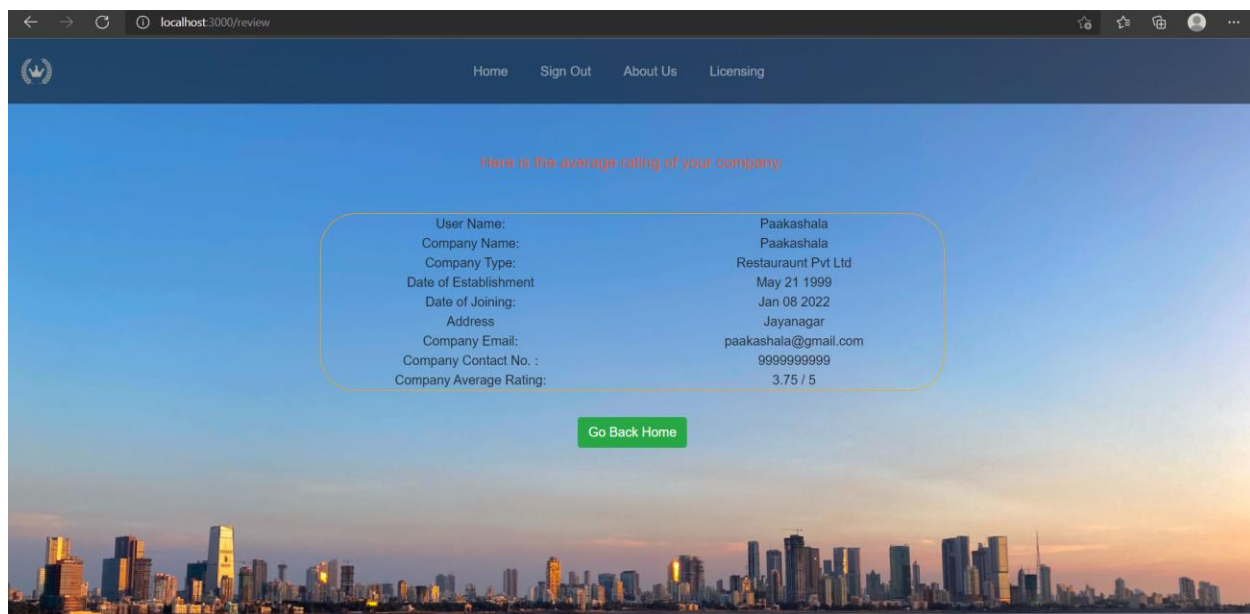
Admin of BLRIDN station providing rating for order_ID:13

The admin of BLRIDN(Bangalore-Indiranagar) station issues the rating for order ID:13 as 4/5.

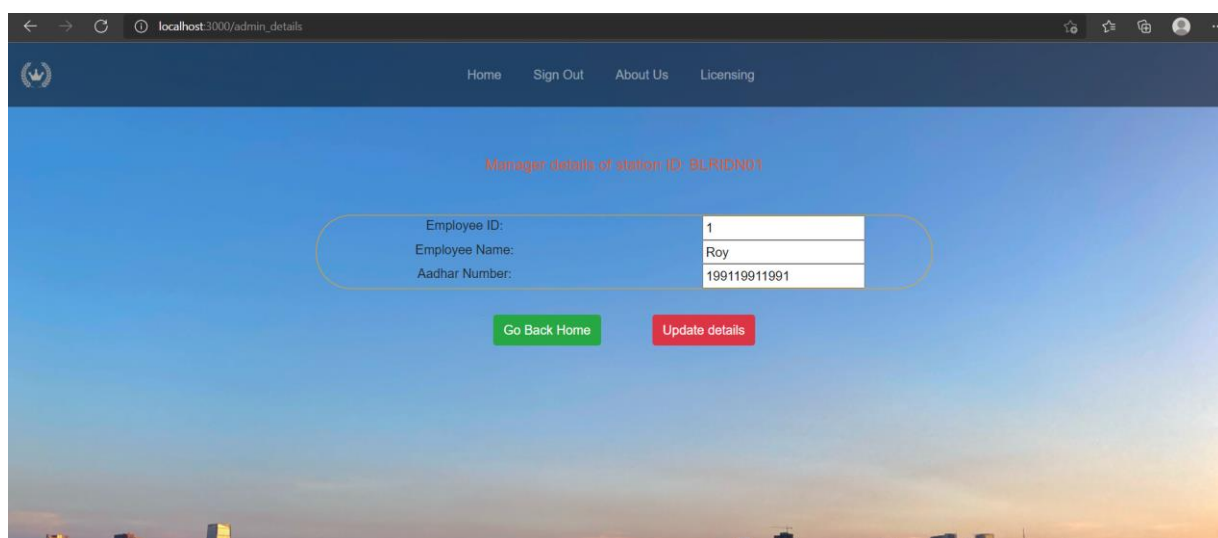
If the rating given by admin is said to be greater than 2, it is shown as 'pass'. If the admin issues a rating of 2 or less, it is shown as 'fail'



Once the restaurant logs in again, they can see that the status of order with order ID 13 is updated as pass.



The restaurant can also check their average rating, which is the average rating of all the donation orders they have placed. Here, we see that Paakashala Restaurant Pvt Ltd has a credibility rating of 3.75/5.



Facility to update the station manager's information

NOTE: This facility is available when the admin signs into his/her station account. The Employee ID field is disabled and cannot be edited by the manager. If any update action is performed on the Aadhar Number of an employee, and the entered number is not a 12-digit unique number, then it will automatically be set as null. This was implemented with the use of triggers in MySQL.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 Conclusion

It is estimated that 40 % of the food produced in India is wasted. Despite adequate food production, the UN has reported that about 190 million Indians remain undernourished. It is further estimated that the value of food wastage in India is around ₹92,000 crores per annum. Also in India, 21.9% of the population live below the national poverty line. Especially in the current scenario where the COVID-19 pandemic is creating unexpected challenges in our country, there are people that are going hungry. In order to bridge the gap between the ones in need and the restaurants that can help, we have come up with a platform called AIDEN to connect them.

AIDEN is a zero-fund organisation that aims to help the less fortunate by getting surplus food from restaurants and communities. The purpose of this NGO Management system is to provide a platform for various restaurants in the city of Bengaluru to provide the food that usually goes to waste at the end of their business hours, to our NGO so that we can serve the under privileged by providing them with one of the basics needs in life – Food.

This system will also ensure that no complications arise from the restaurant side, as the process is fairly simple for them. The restaurants will specify the donation date, the number of people for whom that food will suffice, the food type, and will select one out of the 8 drop off stations where they will deliver the food to. The volunteers in that station will later verify the quality of the food and then distribute it the needy. This platform will not only ensure that food is not getting wasted, but it will also make sure that no one goes to sleep hungry.

6.2 Future Enhancements

The AIDEN database could be improved by adding some more attributes and tables.

The entities could be normalized even further in the future.

The tables could contain more attributes to further give clarity on the real-world entities they point to.

We could also incorporate the concept of GST ID for the restaurants which would be another key factor not only for uniquely identifying the restaurants, but also for verifying their legitimacy.

CHAPTER 7

BIBLIOGRAPHY

1. Ramez Elmarsi and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson, 7th Edition.
2. <https://stackoverflow.com/>
3. <https://www.w3schools.com/>
4. <http://www.udemy.com>