

ABSTRACT

The Sparsh Online Plant Store application is a comprehensive e-commerce platform designed to facilitate the seamless buying and selling of plants. This project report provides an overview of the application's purpose, scope, objectives, functional requirements, and non-functional requirements.

The primary objective of Sparsh is to provide users with a user-friendly online platform to purchase a wide range of plants. This digital store aims to streamline the plant shopping experience, thereby enhancing the organization's efficiency and future growth prospects. Sparsh incorporates role-based authorization with two primary roles: "admin" and "user." Admin users hold administrative privileges, enabling them to perform CRUD operations on products.

The project abstract underscores the critical aspects of the Sparsh application's development and implementation. By addressing these functional and non-functional requirements, Sparsh aims to provide users with a secure, reliable, and user-friendly platform for purchasing plants online. With a strong focus on data security, performance optimization, and user experience, Sparsh is poised to become a prominent player in the online plant retail space.

ACKNOWLEDGEMENT

Apart from the efforts of the team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

The completion of any inter-disciplinary project depends upon cooperation, coordination and combined efforts of several sources of knowledge.

We are eternally grateful to our guide Mrs. Manjiri Deshpande, for her even willingness to give us valuable advice and direction under which we executed this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks. I extend my sincere thanks to our respected Centre Co-ordinator Mr. Rohit Puranik for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

Gaikwad Suraj Mahadev (233152)

Rohit Kumar (233194)

Table of Contents

INTRODUCTION...	6
1.1 Introduction	6
PRODUCT OVERVIEW AND SUMMARY...	8
2.1 Purpose	8
2.2 Scope	8
2.3 User Classes and Characteristics	8
2.4 Design and Implementation Constraints	10
REQUIREMENTS	11
3.1 Functional Requirements.....	11
3.1.1 Use case for Administrator.	11
3.1.2 Use case for Customer	11
3.1.3 Use case for Supplier	12
3.2 Non - Functional Requirements	13
3.2.1 Usability Requirement.....	13
3.2.2 Performance Requirement.....	13
3.2.3 Reliability Requirement	14
3.2.4 Portability Requirement	14
PROJECT DESIGN	15
4.1 Data Model	15
4.1.1 Database Design	17
4.2 Process Model	20
4.2.1 Functional Decomposition Diagram	20
4.2.2 Data Flow Diagram (DFD)	22
4.2.3 Activity Diagram	23
4.2.4 Sequence Diagram	28
4.2.5 Deployment Diagram	29
SYSTEM AND HARDWARE REQUIREMENTS.....	33
PROJECT SCREEN SHOTS.....	34

FUTURE SCOPE...42

CONCLUSION..... 43

REFERENCES 44

LIST OF FIGURES

FIGURE 1: Zero Level DFD	21
FIGURE 2: Level 1 DFD Customer And User	21
FIGURE 3: Level 2 DFD For Admin.....	23
FIGURE 4: Sequence Diagram.....	24
FIGURE 5: E-R Diagram	25

INTRODUCTION

In the age of digital transformation, where convenience and accessibility are paramount, the Sparsh Online Plant Store application emerges as a solution tailored to meet the growing demands of plant enthusiasts. This project report serves as an introductory guide to the development and specifications of the Sparsh application, an online platform that facilitates the seamless purchase of plants.

Purpose: The primary objective of this document is to present a comprehensive overview of the Sparsh Online Plant Store application. By delineating its purpose, scope, objectives, and requirements, this report sets the stage for understanding the intricacies of a modern e-commerce platform designed to redefine the plant shopping experience.

Scope: The Sparsh application is conceived as an online haven for plant enthusiasts, offering a variety of functionalities to cater to users' diverse needs. From user registration and authentication to sophisticated admin operations, Sparsh encompasses an array of features such as role-based authorization, product browsing, shopping cart management, checkout, and more.

Objective: Sparsh is developed with a clear purpose: to provide a virtual marketplace for the purchase of plants. This project aims to empower the organization by facilitating efficient resource management and aligning with its future goals.

These encompass various aspects such as stringent security measures to safeguard user data, system reliability through regular data backups and administration, uninterrupted system availability, maintainability through strategic technological choices, portability across devices and environments, and more.

As we delve further into this project report, we will explore each facet of Sparsh's development journey, gaining insights into how the application's features align with its objectives and how its design prioritizes user experience, security,

and sustainability. By meticulously addressing both functional and non-functional requirements, Sparsh endeavors to establish itself as a pioneer in the online plant retail landscape, catering to the needs of plant enthusiasts and fostering a harmonious ecosystem for buying and selling plants online.

1.1 PROJECT OBJECTIVE

The objective of this project report is to provide a comprehensive overview of the development, specifications, and functionalities of the Sparsh Online Plant Store application. This report aims to clearly outline the purpose, scope, requirements, and features of the application, shedding light on its significance, user benefits, and technological aspects. By achieving this objective, the report seeks to facilitate a deeper understanding of Sparsh as a modern e-commerce platform dedicated to simplifying the process of purchasing plants online.

1.2 PROJECT OVERVIEW

The Sparsh Online Plant Store application is an innovative e-commerce solution designed to redefine the plant shopping experience. This project report provides a comprehensive overview of the development, purpose, features, and functionalities of the Sparsh application. By detailing its objectives, scope, requirements, and non-functional attributes, this report aims to offer a holistic understanding of Sparsh as a modern online plant retail platform. The application contains two modules – the Admin module and the User module. Our application will not only help the admin to preview the data but it will also allow them to edit, add or update records.

1.3 PROJECT SCOPE

The scope of this project report is to provide a comprehensive understanding of the Sparsh Online Plant Store application, outlining its purpose, functionalities, and attributes. By defining the boundaries and extent of the application's capabilities, this report aims to elucidate the project's focus and the value it brings to users.

1.4 STUDY OF THE SYSTEM

1.4.1 MODULES:

The system after careful analysis has been identified to be presented with the following modules and roles. The modules involved are:

- Administrator
- Users

1.4.1.1 Administrator:

DESCRIPTION - The admin can add users, update user identity verification , update appointment status, create/update/delete events.

MAIN FLOW OF EVENTS

1. Admin logs in the system.
2. Admin can add users.
 - 2.1 admin can fill the user's details.
3. Admin maintains user's record.
 - 3.1 List of registered user's details is displayed.
 - 3.2 Admin can see/update the user's ID proofs.
4. Admin View the blood stocks available and also can manage them.
5. Admin can approve or reject the blood consumption requests made by the users.
6. Admin can create events and manage and see them in a list form.

➤ **Add user**

Admin can add new users and register him with details like name, Email, age, gender, Idcard and its number etc.

➤ **Edit users**

Admin will have a list view of all the existing users. He can also edit for a particular user.

➤ **Add/Remove blood stock**

Admin can add /remove blood stocks from the inventory.

➤ **View blood donation history**

Admin will have a dashboard where he/she can view all the blood donation history.

➤ **Manage Events**

Admin can add new events and also can remove from them.

➤ **View all events**

Admin will have a list view of all the existing events. He/she can also edit for a particular event.

1.4.1.2 Users:

MAIN FLOW OF EVENTS

1. Users can sign up in the system.
2. Users can sign up by filling their details like name, Email, age, phoneNumber , Id proof, Id number.
3. Users can view a list of all appointments.
4. Users view a list of Blood donations done by him/her.

PRE –CONDITION - Users must first log in with his/her credentials.

POST CONDITION - The dbms database is updated.

Add Addresses

Users can add their multiple addresses by filing proper details.

➤ **List All Addresses**

Users will have a list view of all the Addresses of him/her.

➤ **List All Blood Donations**

Users will have a list view of all the donations done by him/her.

➤ **Book Appointments**

Users can book an appointment for the patient according to the blood group.

➤ **List All Appointments**

Users will have a list view of all the appointments generated by him/her.

SYSTEM ANALYSIS

System analysis is the process of gathering and interpreting facts, diagnosing problems, and using information to recommend improvements on the system. System analysis is a problem solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified, and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

2.1 EXISTING SYSTEM

In the existing system, Blood bank management functionalities are handled manually by Blood bank staff by keeping the records of every donor and consumer in a register. There is no record present for billings of patients and no history is available for appointments.

- It is less user-friendly.
- It is difficult to know the user's history.
- Difficult to maintain a user record.

2.2 PROPOSED SYSTEM

In the proposed system, we are developing a web application for blood bank management. In which two modules have access to the system. The proposed system allows to keep the record for every user and their blood consumptions.

SYSTEM REQUIREMENT SPECIFICATION

2.2.1 Product Description:

Blood For Lives is an online website, an outstanding way of bringing users on an online platform to provide blood bank related facilities in an efficient manner. This website provides an Interactive interface through which a user can interact with different areas of the application easily by maintaining the blood stocks as well as user's information. OBBMS provides a simple interface and platform to ease the process of getting the information about blood availability online. It includes smooth functionality and efficiency that get the user's work done. Blood For Lives keeps the information about the blood stock and user data updated.

Problem Statement:

- Lack of immediate retrievals :

It is very difficult to retrieve information and to find particular information like- E.g. - To find out the user's history, the admin has to go through various registers. This results in wastage of time.

- Lack of prompt updating :

Various changes to information like blood stocks history are difficult to make as paper work is involved.

- Error prone manual calculation:

By manual calculations error can be occur and take a lot of time this may result in incorrect information. For example calculation of available blood stocks based on various donations and consumptions.

- Preparation of accurate and prompt reports:

It is difficult task to collect information about patients from various register.

SYSTEM OBJECTIVES

- To provide a Web application for blood bank management
- To provide an web app for users to donate or consume blood easily.

2.2.2 SYSTEM REQUIREMENTS

2.3.3.1 NON-FUNCTIONAL REQUIREMENTS

Following Non-Functional Requirements will be there in the insurance to the internet:

- Secure access to Users confidential data.
- 24x7 availability.
- Better component design to get better performance at peak time.
- Various other Non-Functional requirements are:

1) Security:

- System use shall not cause any harm to human users.
- This web-based application will be password protected and any update of any entries of data will be carried out by privileged users.
- System shall use a secured database.
- Users can just read information but they cannot edit or modify anything except their personal information details. Only the System Administrator will do system administration and maintenance work.

2) Data Integrity:

- All the data in the OBBMS must be accurate and reliable.

3) Maintainability:

- OBBMS must have a high level of maintainability.

4) Portability:

- The application must be easily portable on any system.

5) Error Handling:

- The system should handle expected and unexpected errors in ways that prevent loss in information and long downtime periods.

6) Serviceability:

If any issue arises in the OBBMS, it should be programmed in such a way that developers can service it again easily.

FUNCTIONAL REQUIREMENTS**Admin:**

- **Database Management:**

Admin can control the database and keep track of all the records of users, products and categories.

- **User Details:**

Admin can view the personal details of the users.

- **Managing requests from Users:**

Admin will be responsible for approval and rejection of the requests made by the consumers and donor for getting the blood of required type and donation of the blood respectively.

- **View blood Stocks:**

Admin will keep track of blood's stocks for the management purpose.

- **Organize a blood donation camp:**

Admin can organize a blood donation camp and he can assign the dates and centres for the blood donation camp.

- **Login & Logout:**

Admin will have to login to monitor and perform the functions mentioned above.
Admin can logout after completion of his work

User:

- **Login:**

Registered users can log in using their credentials, granting them access to personalized features and a customized shopping experience.

- **Product Catalog Browsing:**

Users can view an extensive catalog of plants available for purchase. The catalog displays plant names, images, prices, and brief descriptions to help users make informed decisions.

- **view profile:**

users can see the orders he/she has done in the past in the form of order history. Also they can edit/update his/her profile.

- **Logout:**

After placing order, the users can logout himself.

SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. Its emphasis is on translating design specifications to performance specification. System design has two phases of development.

- Logical Design
- Physical Design

During logical design phase the analyst describes inputs (sources), outputs (destinations), databases (data stores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

3.1 INPUT AND OUTPUT DESIGN

3.1.1 INPUT DESIGN:

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common

cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer-based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

3.1.2 OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and help sin decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

DATABASE DESIGN

3.2 DATABASE

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- Primary key - the field that is unique for all the record occurrences
 - Foreign key - the field used to set relation between tables
- Normalization is a technique to avoid redundancy in the tables.

3.3 SYSTEM TOOLS

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

3.3.1 FRONT END

REACT(v18.2.0)

React is a library which is developed by Facebook is utilized to implement the frontend. React (also known as React.js or React JS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

BACKEND:

The back end is implemented using MySQL which is used to design databases.

MySQL (v8):

management system (RDBMS). The SQL phrase stands for Structured Query Language. Application software called Navicert was used to design the tables inMySQL.

Spring-Boot[STS-4]:

This is used to connect MYSQL and fetch data from database and store the data in database. The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is Open-source Framework.



Figure 1 Zero Level DFD

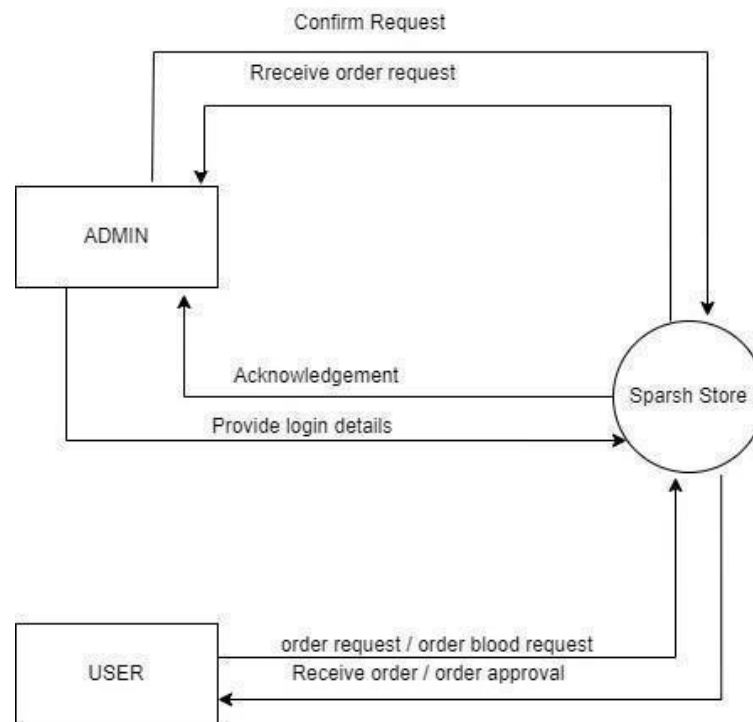


Figure 2 Level 1 DFD

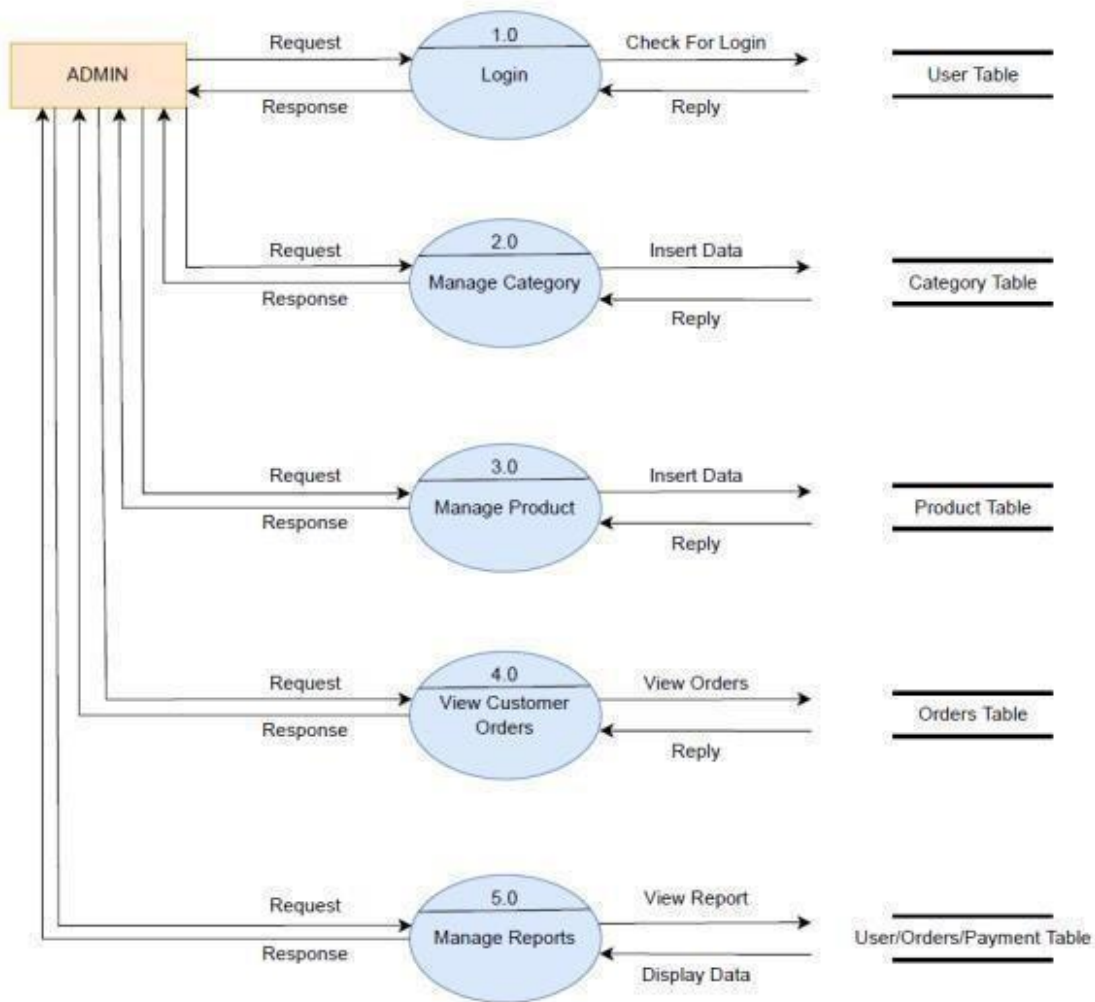


Figure 3 Level 1 DFD For Admin

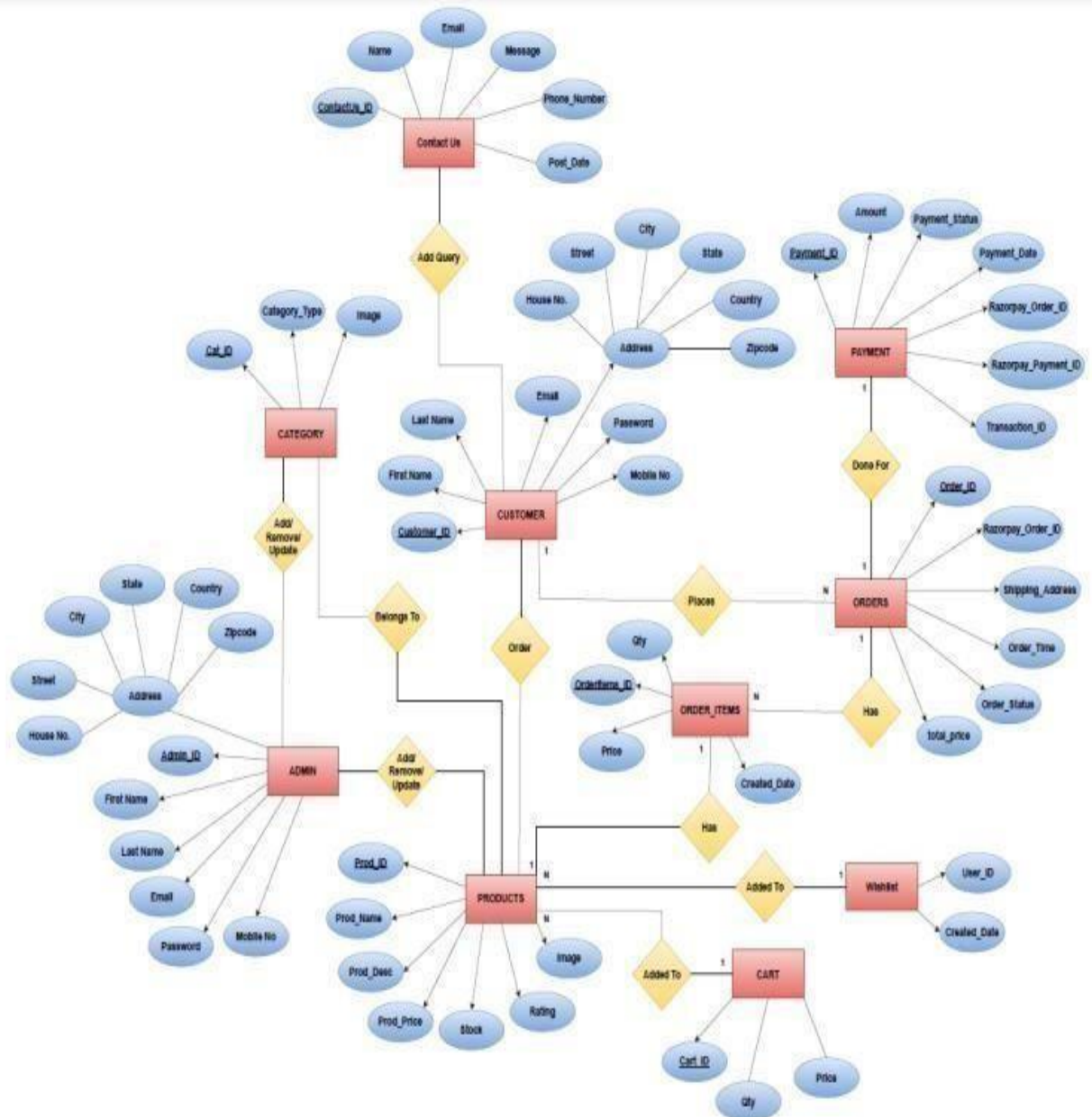


Figure 4 E-R Diagram

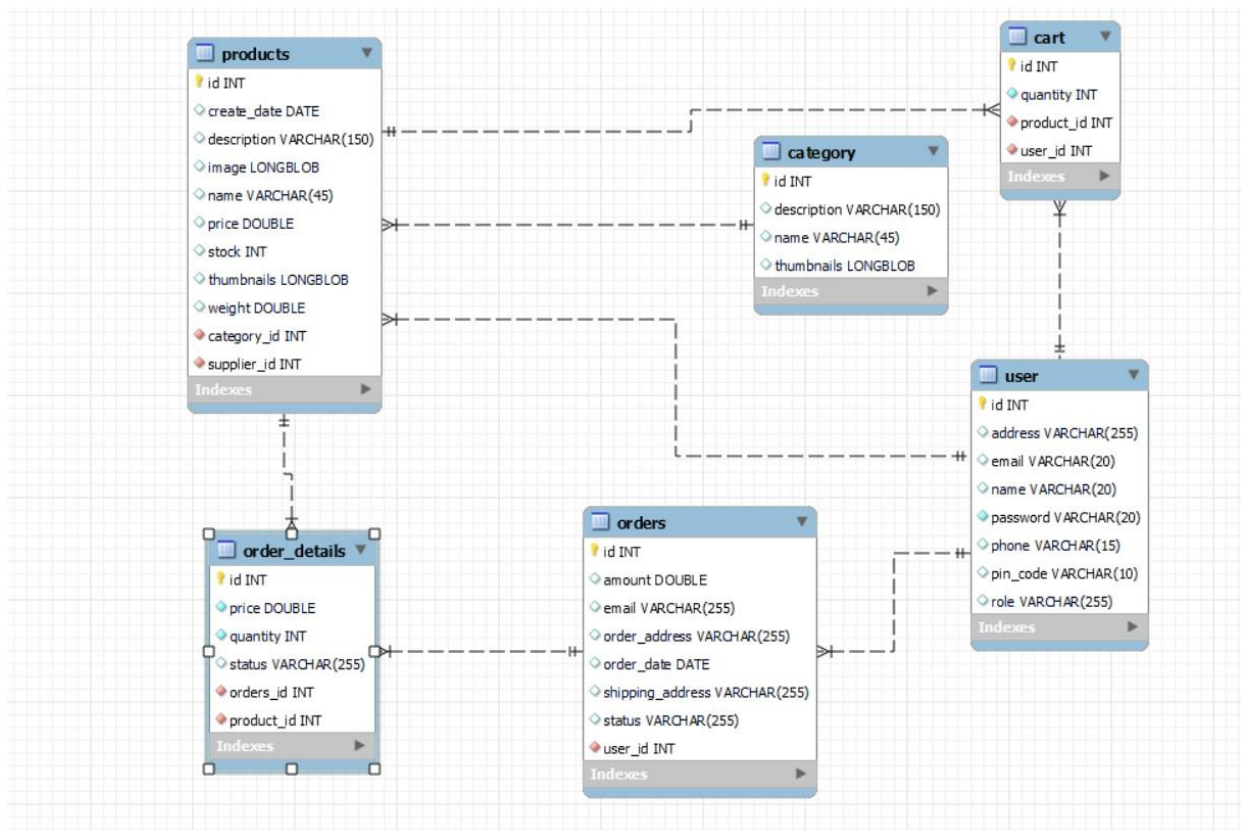


Figure 5 Workbench generated E-R Diagram

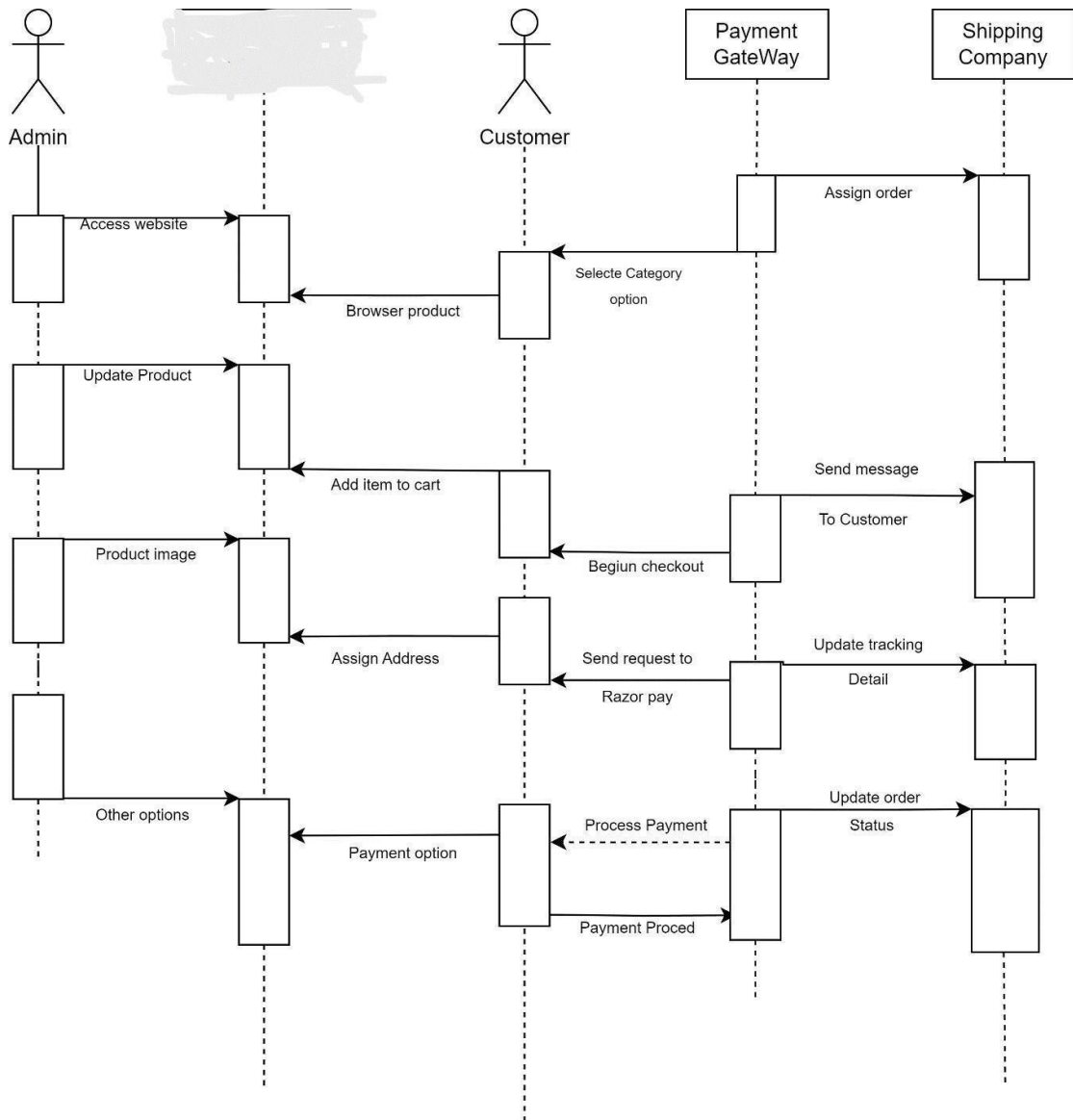


Figure 6 Sequence Diagram

TABLE STRUCTURE:

User:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(50)	YES	UNI	NULL	
first_name	varchar(30)	YES		NULL	
last_name	varchar(30)	YES		NULL	
otp	int	YES		NULL	
password	varchar(350)	YES		NULL	
password_reset_code	varchar(255)	YES		NULL	
phone_number	varchar(255)	YES		NULL	
role	varchar(30)	YES		NULL	
address_id	bigint	YES	MUL	NULL	

Addresses:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
city	varchar(30)	YES		NULL	
country	varchar(30)	YES		NULL	
house_no	varchar(255)	YES		NULL	
state	varchar(30)	YES		NULL	
street	varchar(60)	YES		NULL	
zipcode	varchar(255)	YES		NULL	

Carts:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
created_date	date	YES		NULL	
quantity	bigint	YES		NULL	
product_id	bigint	YES	MUL	NULL	
user_id	bigint	NO	MUL	NULL	

Categories:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
category_name	varchar(100)	YES		NULL	
image_path	varchar(255)	YES		NULL	

ContactUs:

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(255)	YES		NULL	
message	varchar(500)	YES		NULL	
name	varchar(255)	YES		NULL	
phone_number	varchar(255)	YES		NULL	
post_date	date	YES		NULL	

Order Items:

```
mysql> desc order_items;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
created_date	date	YES		NULL	
price	double	YES		NULL	
quantity	bigint	YES		NULL	
order_id	bigint	YES	MUL	NULL	
product_id	bigint	YES	MUL	NULL	

Users:

```
mysql> desc users;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(50)	YES	UNI	NULL	
first_name	varchar(30)	YES		NULL	
last_name	varchar(30)	YES		NULL	
otp	int	YES		NULL	
password	varchar(350)	YES		NULL	
password_reset_code	varchar(255)	YES		NULL	
phone_number	varchar(255)	YES		NULL	
role	varchar(30)	YES		NULL	
address_id	bigint	YES	MUL	NULL	

Payments:

```
mysql> desc payments;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
payment_status	varchar(255)	YES		NULL	
payment_amount	double	YES		NULL	
payment_date	datetime(6)	YES		NULL	
razorpay_order_id	varchar(255)	YES		NULL	
razorpay_payment_id	varchar(255)	YES		NULL	
transaction_id	varchar(255)	YES		NULL	
order_id	bigint	YES	MUL	NULL	

Wishlist:

```
mysql> desc wishlist;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
created_date	date	YES		NULL	
product_id	bigint	YES	MUL	NULL	
user_id	bigint	NO	MUL	NULL	

Contactus:

```
mysql> desc contactus;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
email	varchar(255)	YES		NULL	
message	varchar(500)	YES		NULL	
name	varchar(255)	YES		NULL	
phone_number	varchar(255)	YES		NULL	
post_date	date	YES		NULL	

Wishlist:

```
mysql> desc wishlist;
```

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
created_date	date	YES		NULL	
product_id	bigint	YES	MUL	NULL	
user_id	bigint	NO	MUL	NULL	

SYSTEM AND HARDWARE REQUIREMENTS

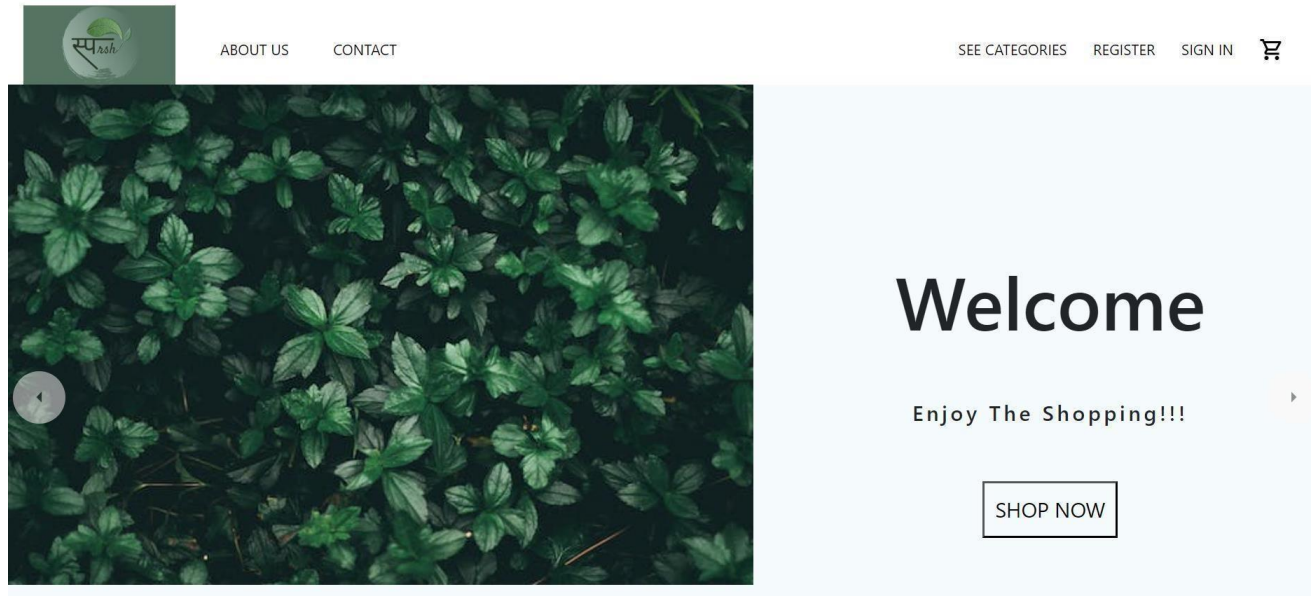
Software Requirements:

- Technology: J2SE and J2EE, Hibernate Spring Boot
- Web-Technologies: React, CSS, JavaScript
- Web Server: Tomcat 9.0
- Java Version: JAVA Version 11
- Database: MySQL 8.0
- IDE: IntelliJ

Hardware Requirements (Minimum):

- Processor: Intel i3 or Ryzen 3
- Ram Capacity: 4GB
- Hard Disk: 100GB

PROJECT DIAGRAM





Enjoy The Shopping!!!

SHOP NOW



Plants can add beauty and life to any space, whether it's an office, home, or outdoor garden. They come in various shapes, sizes, and colors, making them versatile and appealing.



Useful Links

Home
Shop Now

About us
View Categories

Contact

📍 Jaipur, India

📞 +91 9119*****

✉ contact Info@sparhstore.com



ABOUT US

CONTACT

SHOW CATEGORIES

Rohit Kum

ABOUT US




Sparsh


Get the & Plants to make your home
look modern & Society looks
beautiful!!!

Sparsh Store is an Indian Platform to provide plants for your farm, home and society. Can order variety of plants from our website.

OUR MISSION

Making india green and beautiful in every street, in every city, in every home.


ABOUT USCONTACT


SEE CATEGORIESREGISTERSIGN IN

CREATE AN ACCOUNT

By creating an account, I consent to the processing of my personal data in accordance with the [PRIVACY POLICY](#)


CREATE


ABOUT USCONTACT

SEE CATEGORIESREGISTERSIGN IN

SIGN IN

This reCAPTCHA is for testing purposes only. Please report to the site admin if you are seeing this.

I'm not a robot

reCAPTCHA
Privacy - Terms

LOGIN

[Forgot Password?](#)
[New to SparshStore ? Create an account](#)

[ABOUT US](#)[CONTACT](#)[SHOW CATEGORIES](#)

Rohit Kumar ▾

My Profile

[Update
Profile](#)

[ABOUT US](#)[CONTACT](#)[SEE CATEGORIES](#)

Suprit Sharma ▾

My Address

L-32, Sureksha, Mahindra City , Jaipur, Rajasthan, India. Zipcode: 302017

House No. :

L-32, Sureksha

Street or Landmark :

Mahindra City

City :

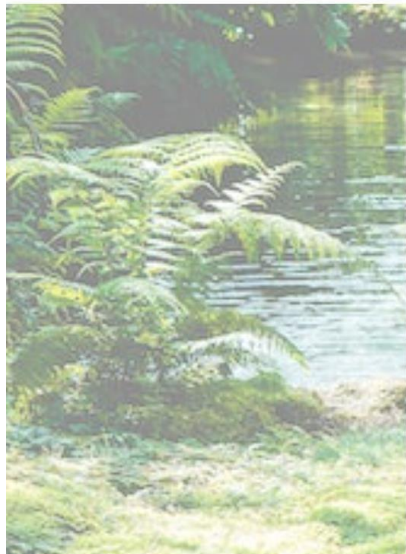
Jaipur

State :

Select State ▾

Country :

India

[ABOUT US](#)[CONTACT](#)[SHOW CATEGORIES](#)

Rohit Kumar ▾

Click On Submit To Change
Password

jangirrohit03@gmail.com

Submit



[ABOUT US](#)

[CONTACT](#)

[SHOW CATEGORIES](#)

Rohit Kumar ▾

- [My Profile](#)
- [Update Password](#)
- [See Categories](#)
- [See Products](#)
- [See All Users](#)
- [List All Orders](#)
- [View Sales](#)

[Sign out](#)

Orders List

All Orders	Pending Orders	Shipped Orders	Cancelled Orders	
------------	----------------	----------------	------------------	--

No Orders



Money Plant

Price : ₹240



barad Plant

Price : ₹240



fgghh

SHOP NOW

Indoor

SHOP NOW

OutDoor

SHOP NOW



[ABOUT US](#)

[CONTACT](#)

[SEE CATEGORIES](#)



Suprit Sharma ▾

YOUR CART

CONTINUE
SHOPPING

CLEAR
CART

Shopping Bag (1) Your
Wishlist (0)

Product : barad Plant
Category : Indoor

− 2 +

₹240



ORDER SUMMARY

Subtotal	₹ 480
Estimated Shipping	₹ 40
Shipping Discount	₹ -40



Quick Contact

FUTURE SCOPE

The software products we produce are nursery management systems requested by our customers and are intended to support large-scale nursery management of nurseries. This system will allow you to manage your kindergarten management better. It's web-based so that you can access it over the Internet, including kindergarten registration, clerk details, parents and guardians' details, and your child's details. The functions of the system are as follows.

Registration

The registration function has two main tasks: parent registration and child registration. Parents can create a new account as a member through the online registration service. Parents will then be able to register their children

Child approval by staff

Staff must be able to manage the process of preschool approval of children. Each child should be placed in a class according to age. The team also prioritized deleting useless records of applicants.

Store child's and parent's details

All information from parents and children is stored in the database system. B. Address, age, date of birth, phone number, allergies, etc. Children should be assigned to classes and programs based on their age. Registered parents can see the payment history, kindergarten updates, important information, and status. Parents can also update the data in the system.

CONCLUSION

The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to maintain patient records efficiently and securely for future use.

This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using React.js, usage of responsive templates, designing of android applications, and management of database using MySQL. The entire system is secured. Also, the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project.

This project has given us great satisfaction in having designed an application which can be implemented to Sparsh Store.

Sparsh Store administrators would be able to significantly improve the operational control and thus streamline operations. This would enable to improve the response time to the demands of donor and consumer because it automates the process of collecting, collating and retrieving user's information

There is a scope for further development in our project to a great extent. A number of features can be added to this system in future like combining hospitals with this project. Also making dashboard more user's friendly. Also generate blood donation certificate.

These features could have implemented unless the time did not limit us.

REFERENCES

- [1] Enlightenment, Cody Lindley-First Edition, based on JavaScript 1.5, ECMA-262, Edition
- [2] Mc Graw Hill's, Java: The complete reference 7thEdition, Herbert Scheldt
- [3] Complete CSS Guide, Maxine Sherrin and John Allsopp-O'ReillyMedia; September 2012
- [4] Ali Bahrami, Object-Oriented System Development, Third Edition, TataMcGraw Hill Edition

ONLINE REFERENCE

- [1] www.Google.com
- [2] www.w3school.com
- [3] www.javatpoint.com
- [4] <https://docs.spring.io/spring-framework/docs>
- [5] <https://getbootstrap.com/docs/4.0>