MANGALORE UNIVERSITY

"Badminton Hub"

A Project Report
Submitted in partial fulfillment for the award of the degree in

BACHELOR'S IN COMPUTER APPLICATION

By

Dhanraj J Poojary

Reg.No- 201381522107

Nachiketh Joshi

Reg.No- 201381522138

Vinay

Reg.No- 201381522153

Under the guidance of

Smt. Deepa Shetty
Lecturer
Department of Computer Science
Sri Bhuvanendra College
Karkala

DEPARTMENT OF COMPUTER SCIENCE

SRI BHUVANENDRA COLLEGE

KARKALA-574104

2022-23

DEPARTMENT OF COMPUTER SCIENCE
SRI BHUVANENDRA COLLEGE
KARKALA- 574104

CERTIFICATE

This is to certify that the project work entitled **“Badminton Hub”** has been successfully carried out By **Dhanraj J Poojary** [Reg No- 201381522107**], Nachiketh Joshi [**Reg.No- 201381522138**] and Vinay [**Reg.No- 201381522153**]** Students of sixth semester BCA under the supervision and guidance of **Smt. Deepa Shetty, Lecturer,** Department of Computer Science, Sri Bhuvanendra College, Karkala. This dissertation is submitted in partial fulfillment of the requirement for the award of **Bachelor’s of Computer Application** by Mangalore University during the academic year 2022-2023.

Smt. Deepa Shetty
Internal Guide
Lecturer, Dept of Computer Science
Sri Bhuvanendra College, Karkala

Smt. Swathi K
H.O.D ,
Dept of Computer Science
Sri Bhuvanendra College, Karkala

Examination Center : Sri Bhuvanendra College, Karkala

Date of Examination : ……………………………………

Internal Examiner

External Examiner

TELEPHONE : 08258-230234 (O)
08258-233214 (PRI)

E-MAIL : principal@sribhuvanendra.org
FAX : 0825825114

SRI BHUVANENDRA COLLEGE

"College With Potential For Excellence" –Awarded By Ugc New Delhi
[Reaccredited with B by NAAC]
KARKALA - 574 104, UDUPI DISTRICT
(Affiliated to Mangalore University & Sponsored by The Academy of General Education, Manipal- 576104)

Date : 15-06-2023

TO WHOM SO EVER IT MAY CONCERN

This is to certify that the project work entitled " **Badminton Hub** " has been successfully carried by **Dhanraj J Poojary** [Reg No- 201381522107**], Nachiketh Joshi [**Reg.No- 201381522138**] and Vinay [**Reg.No- 201381522153**]** Students of sixth semester BCA, in our college, as partial fulfillment of the requirement for the award of **Bachelor's of Computer Application** by Mangalore University during the academic year 2022-2023.

Further it is certified that, to the best of my knowledge that the matter embodied in this work has not been submitted for the award of any other degree.

Smt. Swathi K	**Dr. Manjunatha A Kotian**
H.O.D ,	**PRINCIPAL**
Dept of Computer Science	Sri Bhuvanendra College
Sri Bhuvanendra College, Karkala	Karkala

DECLARATION

The Project work entitled **“Badminton Hub”** has been successfully carried out by **us Dhanraj J Poojary** [Reg No- 201381522107**], Nachiketh Joshi [**Reg.No- 201381522138**] and Vinay [**Reg.No- 201381522153**]** students of sixth semester BCA, under the supervision and guidance of **Smt. Deepa Shetty, Lecturer**, Department of Computer Science, Sri Bhuvanendra College, Karkala. This dissertation is submitted in partial fulfillment for the award of degree in **Bachelor’s of Computer Application (BCA)** by **Mangalore University** during the academic year 2022-2023.

Place: Karkala

Date :15-6-2023

Dhanraj J Poojary

[Reg No- 201381522107**],**

Nachiketh Joshi

[Reg.No- 201381522138**] ,**

Vinay

[Reg.No- 201381522153**]**

ACKNOWLEDGMENT

The magnitude of this project demanded the co-operation, guidance and assistance from a number of people and by the grace of God we have been fortunate enough to have this in the entire process of completion of our project work. Many have guided and helped us and whatever we have been able to put forward in terms of work, is only due to such guidance and help.

Our heart-felt gratitude to our internal guide, **Smt. Deepa Shetty**, Lecturer, Department of Computer Science, Sri Bhuvanendra College Karkala, for her continual support and motivation throughout our project.

We take this opportunity to express our gratitude to **Smt. Swathi K,** HOD, Department of Computer Science, Sri Bhuvanendra College Karkala, for her support and advices throughout our course.

Our heart-felt gratitude to our Principal, **Dr. Manjunatha A Kotian**, for his support and encouragement.

Our sincere thanks to all the Lecturers of Computer Science Department, Sri Bhuvanendra College for their help and guidance during our project.

We express our special thanks and heartfelt gratitude to Thribhuvan Badminton Shop Karkala for their constant support and guidance throughout our project.

Thanking from,

Dhanraj J Poojary
Nachiketh Joshi
Vinay

INDEX

1.INTRODUCTION

1.1 INTRODUCTION OF THE SYSTEM

Our Online Badminton shopping system “Badminton Hub” is committed to providing you with the lowest priced badminton equipment and gear on the web. Our diverse badminton product line caters to beginners, intermediate and advanced level badminton players. We also stock badminton shoes, bags, accessories and we also repair your rackets to help you get the most out of your game.

1.1.1 Title of the Project:

Badminton Hub

1.1.2 PROJECT CATEGORY:

Web Application using php

1.1.3 OVERVIEW:

This website is an Online Platform designed for Managing Badminton shop and related Operation.Here user can send service request and purchase products online.

1.2 OBJECTIVES OF THE SYSTEM:

*****Choose Products faster and easier at one place.

*To Develop user friendly Software with ease of Maintenance.

*Provides Security to Data using Login & Password.

*Keeps Track of Transactions done .

1.3 SCOPE OF THE SYSTEM

- Mobile application can be developed.
- Website can be developed in multiple languages like Hindi, Kannada, etc.. ➢ Online payment can be added

1.4 STRUCTURE OF THE SYSTEM:

Module Description

1.4.1 LOGIN & REGISTRATION MODULE:

The project has login & regiter module where new user has to register his details and set password and then enter the password and login to the site .

1.4.2 SHOP MODULE:

The Project has Shop Module where the user can view the products added by the admin and add the product to the cart and later place order for the product

1.4.3 CHECKOUT MODULE:

In this module the products placed on the cart is proceeded to the payment where bill is generated and payment is done by the customer.

1.4.4 FEEDBACK MODULE:

In this module the user can give feedback and rating about the user experience of the individual products delivery or return which help the admin to improve each time.

1.4.5 SERVICE MODULE:

In this module user can book the appointment for particular date and for particular time ,if it is already booked thrn he can choose the different slot and can cancel the booked slot before 24 hours of booking day.

1.4.6 ADMIN MODULE:

Admin logins to the site.The admin manages the all the operation of the site,the admin adds the product to the shop and manages the order and all other operations of the site

1.5 END USERS:

The end users here are customer. The customer can buy the products in this platform.

1.6 SOFTWARE/HARDWARE ARE USED FOR THE DEVELOPMENT /IMPLEMENTATION:

1.6.1 SOFTWARE REQUIREMENTS:

Operating System	: Windows 10
Front End	: HTML, CSS, JavaScript,PHP
Back End	: MySQL

1.6.2 HARDWARE REQUIREMENTS:

Processor	: Intel Core i3
RAM	:4GB
Hard Disk	:100GB

2.SOFTWARE REQUIREMENT SPECIFICATION

2.1 INTRODUCTION :

SRS is a complete description of the behaviour of the development. It includes a set of use cases that describe all the interactions the user will have with the software. It deals with the requirement of the purposed system. It describes what the system should do without describing how the system will do it. An SRS provide a reference for the validation of the final product i.e., the SRS helps the client to determine if the system meets the requirements.

2.2 OVERALL DESCRIPTION:

This section of the SRS describes all general factor of the products and requirements.

2.2.1 PRODUCT PERSPECTIVE:

The software produces simple database rather than complex ones for high requirements and it provides good and easy graphical interface to both new and well as experienced users of the computers.

2.2.2 PRODUCT FUNCTION:

- Maintains the details about products and services.
- The user in this website can require login id and password to login to the system

2.2.3 USER CHARACTERISTICS:

The main user of the system will be authorized user(admin) who has the privilege over the other users.

This web application is built in such a way that person having little knowledge about the internet can use this site easily. It is basically based on clicking or entering the details.

2.2.4 GENERAL CONSTRAINTS:

- It is necessary to fill all mandatory field with proper information.
- The system requires internet connectivity.
- Only authorized personnel can use the system.

The application have relational database.
The application will be implemented using MySQL.

2.2.5 ASSUMPTIONS:

A basic assumption is made about end user that he/she has some knowledge about computer. The project depends on the user's ability to understand the features of the software application and able to use the best of his/her use.

2.3 SPECIAL REQUIREMENT:

2.3.1 EXTERNAL INTERFACE REQUIREMENT:

All the interactions of the website with the user, hardware and software are specification here.

1 USER INTERFACE:

The user interface is designed in PHP. A validation is provided to login the page. On successful validation, the permission to use the website is provided.

2 HARDWARE INTERFACE:

The system should have this minimum hardware requirement:

- Processor: Intel core i3
- RAM: 8GB
- Hard disk: 256 GB

3 SOFTWARE INTERFACE:

- Front End: HTML, CSS, Java Script ,PHP
- Back End: My SQL

4 COMMUNICATION INTERFACE:

- HTML ,PHP AND CSS

2.4 FUNCTIONAL REQUIREMENTS:

2.4.1 USER MODULE:

The project has Customer module where user have to Login and fill the details .The user can select the services he wants and Order the product and give feedback to admin.

2. 4.2 ADMIN MODULE:

Login for the Admin. Dashboard for Admin User. Admin can manage the Orders and service requirement and, he/she can edit the existing products and and the new products to shop

2.4.3 SHOP MODULE:

The main aim for developing this module is to provide the new products.This module is which makes the products available for customer where he can select and order the product of his choice..

2.4.4 CHECKOUT MODULE:

This module provides the user to Order the product and after the Ordering the product he can choose the mode of payment and user has to fill card details and the Invoice is generated in pdf form to user.

2.4.5 FEEDBACK MODULE:

In this module user can give his feedback and mark the the rating as per his/her satisfaction,which can be viewed by the admin

2.4.6 SERVICE MODULE:

In this module user can choose the type of service he wants and book the slots as per his convenience

2.5 DESIGN CONSTRAINTS:

The system must be designed to allow web usability. the system must be designed in such a way that it can be easy to use and visible on most of the browser.There are number of factors in client environment that may restrict the choice of designer.Search factor include standard that must be followed,resource limits,operating environment and reliability and security requirement.

2.6 SYSTEM ATTRIBUTES:

The quality of the websites is such a way that it can be very user friendly to all the user of the website.

- Reliability: We can assure a fully security to the user that we will not provide their personal information to other or any other third parties.
- Maintainability: During maintenance stage, the SRS can be referred for the validation.
- Portability: This system can be run in any operating system and browser.
- Compatibility: This system will be compatible with almost all the web site.
- Flexibility: The system keeps on updating the data according to the changes the changes that takes place

2.7 OTHER REQUIREMENT:

2.7.1 SAFETY REQUIREMENT:

- The user should provide his contact number and valid email it will help the owner or employee to make communication about the services.
- Data will not be shared with others.
- Authorization: Checking for the entity and provide features for them.

2.7.2 SECURITY REQUIREMENT:

The proposed website is secured. It means the administrator can modify and delete the data. All other only have the rights to retrieve the information from the website that is concerned to them.

3. SYSTEM DESIGN

3.1 INTRODUCTION:

The purpose of the design phase is to plan a solution of the problem specified by the requirements of the document. This is the phase in moving from problem specified by the requirement of the document. The design process is the set of iterative steps that enable the designer to describe all aspect the software to be built. Design is essentially the bridge between the requirement specification and final solution from satisfying the requirement. The design of the system essentially for blueprint or plan for the system.

The design process is often divided into separate phases:

- System Design
- Detailed Design

In system design, the focus is on deciding which modules are needed for the system, the specification of those modules, and how the modules should be interconnected. It is also called top-level design. The system design controls the major structural characteristics of the system. It has a major impact on the testability and modifiability of a system design, and it impacts its efficiency. At the end of the system design all the major data structures, file formats and major modules in the system and their specification are decided.

3.2 ASSUMPTIONS AND CONSTRAINTS:

An Assumptions are things that we believe to be true and which we therefore build into the project plan. It is something that we take for granted.

Constraints are things that we know to be true and which must be accounted for in the plan so that we can work around them.

3.3FUNCTIONAL DECOMPOSITION:

The software is decomposed into several modules for the conveniences of the users. Functional decomposition is used to describe a set of steps in which the overall function of a device, system o process are broken into smaller parts. This is used to understand the large and complex process and also helps to solve problems and aids in development of several operations.

3.3.1 SYSTEM SOFTWARE ARCHITECTURE

This refers to the fundamental structures of software system and discipline of creating such structures and systems. Each structure comprises software elements, relationship among them and properties if both elements and relations.

3.3.2 SYSTEM TECHNICAL ARCHITECTURE

It is a form of IT architecture that is used to design computer systems. It involves the development of technical blueprint with regards to the arrangement, interaction and independence of all elements so that system relevant requirements are met.

3.3.3 SYSTEM HARDWARE ARCHITECTURE

This is primarily concerned with internal electrical interfaces among the systems component or subsystems and the interfaces between the system and its external environment.

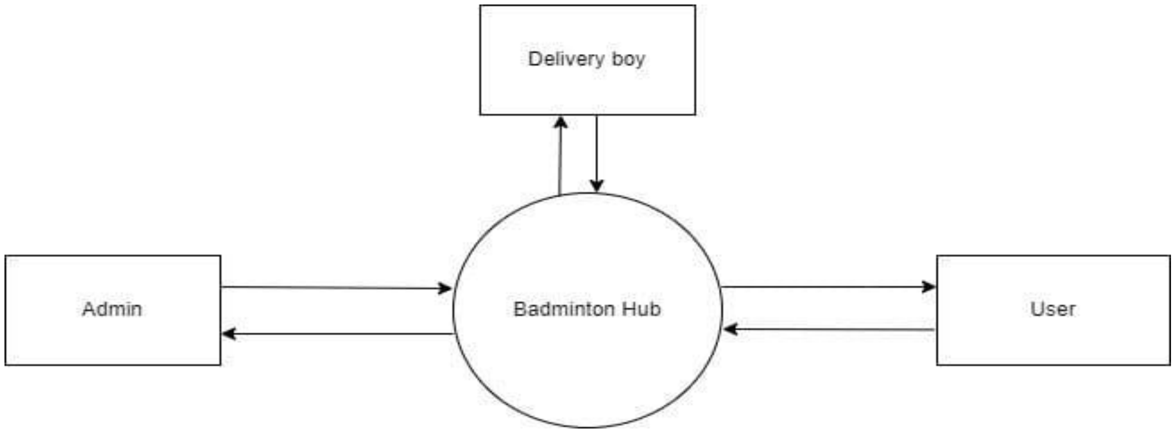
3.3.4 EXTERNAL INTERFACES

External interfaces are typically a products lifeline to the outside world. Such interfaces may be used for the number of purposes including connecting to peripherals, field programming or testing during product manufacturing.

3.4 DESCRIPTION OF PROGRAMS

3.4.1 CONTEXT FLOW DIAGRAM

A context diagram, sometimes called a level 0 data-flow diagram, is drawn in order to define and clarify the boundaries of the software system. It identifies the flows of information between the system and external entities. The entire software system is shown as a single process.



3.4.2 DATA FLOW DIAGRAM

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

There are 4 kinds of system components:

1.Process

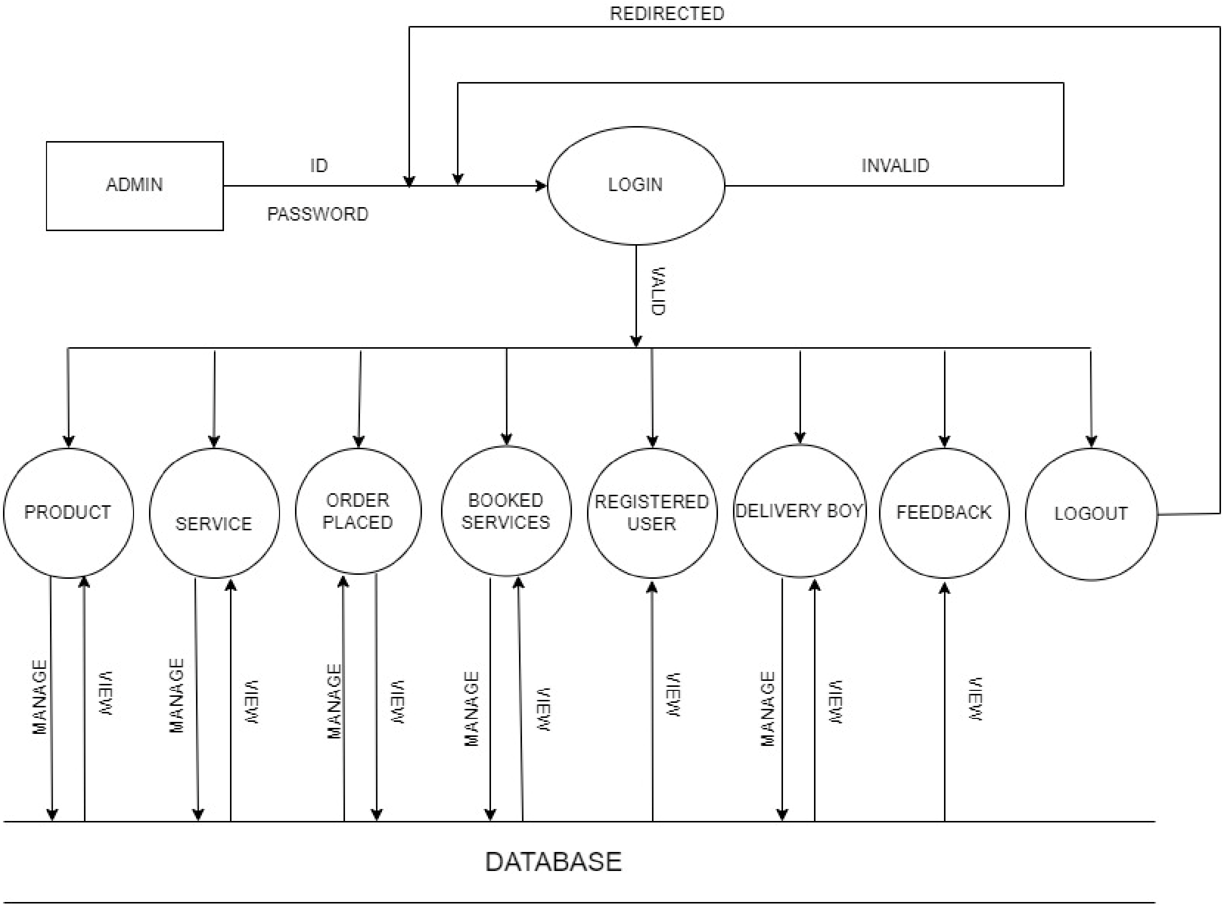
2.External entity

3.Data flow

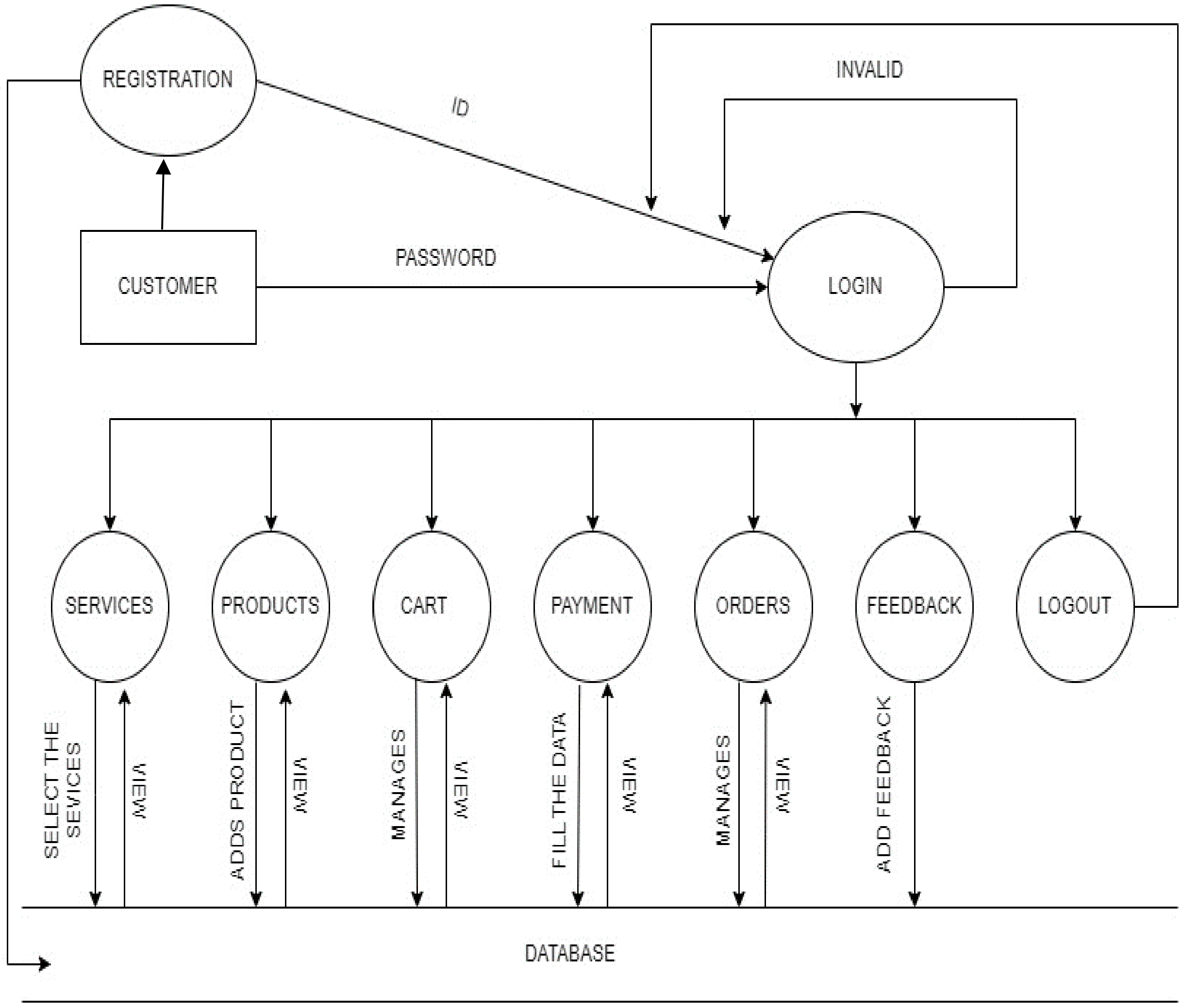
4.Data stores

SYMBOLS	NAME	DESCRIPTION
○	Process	The circle is used to distinguish between data input and comes from a process. It represents functions or process.
▭	Entity	The rectangle represents a simple source or termination of the diagram by mapping real world entities.
→	Data flow	The arrow represents the bidirectional lines and depicts the flow of data from one location to another.
⇄	Data storage	The parallel lines represented here displays the location where the data is stored in the system.

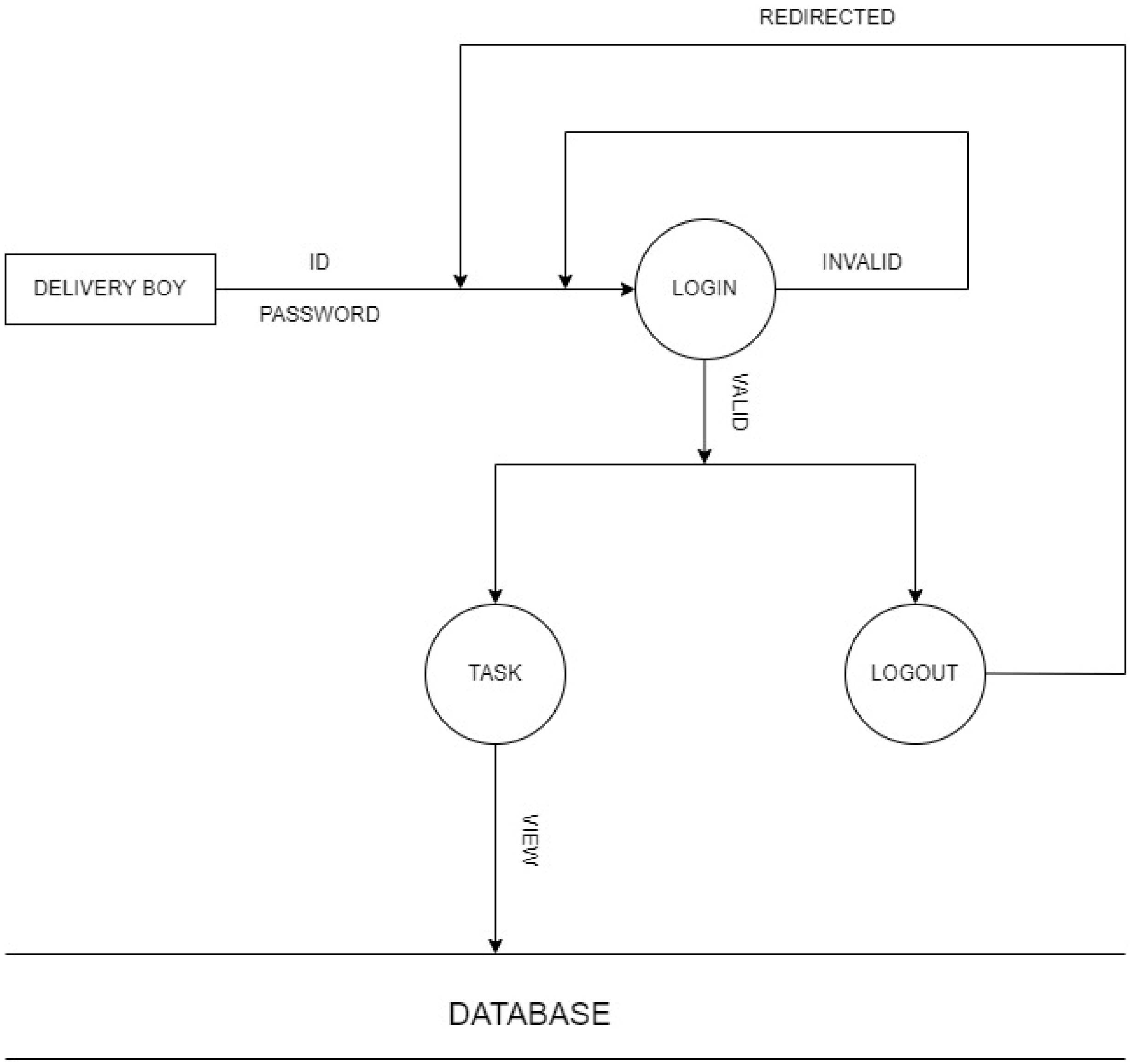
ADMIN LEVEL DFD:



USER LEVEL DFD:



DELIVERY BOY LEVEL DFD:



3.5 DESCRIPTION OF COMPONENTS

3.5.1 Login Module

Input : Entering the Email ID, password.

Process: If Email & Password is correct allows to enter main page or else error msg

Output: Allows viewing the main page and allows to avail the service.

3.5.2 Admin Module

Input : Entering the Email, password.

Process: Verifies the the Admin and stores the details

Output: Allows the admin to view admin control page

3.5.3 Shop Module

Input : Admin adds the new product to shop

Process: The products added are displayed to user interface on the Shop

Output: Allows the admin to update the product .

3.5.4 Checkout Module

Input : User can order the Product of his intrest

Process: Ordered product leads to choose mode of payment and generation of invoice

Output: displays the invoice of the product

3.5.6 Feedback Module

Input : Full detail like ,Message,and rating

Process: Entered details sent to admin

Output: The message sent by user is displayed to admin

3.5.7 Service Module

Input : Select date & type of service

Process: User can select type of service and date ,if its booked choose the other

Output: Allows the admin to view booked slots and cancel the slot further if it neccessary

4.DATABASE DESIGN

4.1 INTRODUCTION:

A database is a collection of interrelated data stored with minimum redundancy to serve many and efficiently. The general objective of database design is to make the data easy. pensive and flexible to the user. Table list of information organized into fields.

4.2 PURPOSE AND SCOPE:

A good database design is, therefore, one that Divides your information into subject based ales to reduce redundant data. Provides Access with the information it requires to join the information in the tables together as needed. Helps support and ensure the accuracy and integrity of your information. Information about the specific subject range, format, or de range a particular specialized database covers is called its scope. A specialized database may be or broad in scope, depending on whether it, for instance, contains materials on one of many subject areas:

4.3 DATABASE IDENTIFICATION:

The database object name is referred to as its identifier. Everything in Microsoft SQL Server can have an identifier. Servers, databases and database objects, such tables, views, columns, indexes, triggers, procedures, constraints, and rules can have identifiers.

4.4 SCHEMA INFORMATION:

Information schema is an ANSI-standard set of read-only views 9/16 information about all of tables, views, columns, and procedures i provides access to database metadata, information about de MySQL server name of a database or table, the data type of a column, or access privileges. Other terms that are sometimes used for this information are data dictionary and system catalog.

4.5 PHYSICAL DATA:

The physical data model is a representation of a data design as implemented, or intended to be implemented, in a database management system. In the lifecycle of a project it typically derives from a logical data model, though it may be reverse-engineered from a given database implementation.

4.6 DATA DICTIONARY:

A Data Dictionary is a collection of names, definitions, and attributes about data elements that are being used or captured in a database, information system, or part of a research project A data dictionary is a centralized repository of metadata.

4.7 TABLE DEFINITION:

Tables are database objects that contain all the data in a database. In tables, data is logically organized in a row-and-column format similar to a spreadsheet. Each now represents a unique record, d each column represents a field in the record.

4.7.1 DATATABASE TABLE

USER RIGISTRATION:

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
id	int(255)	To store the user id	Primary Key
name	varchar(255)	To store the user name	Not Null
user_type	varchar(255)	To store user type	Not Null
email	varchar(255)	To store email address	Not Null
date	Date	To store date of birth	Not Null
phnum	int(255)	To store phone Number	Not Null
password	varchar(255)	To store the password	Not Null
address	varchar(255)	To store the address	Not Null
code	int(20)	To store OTP	Not Null
otp_expiry	Timestamp	To store expiry time of OTP	Not Null

PRODUCTS DETAIL :

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
p_id	int(11)	To store product id	Primary Key
pname	varchar(255)	To store product name	Not Null
pdescript	varchar(255)	To store product description	Not Null
pcost	int(255)	To Store product cost	Not Null
pcat	varchar(255)	To store category of products	Not Null
imgg	varchar(255)	To store image of products	Not Null
pquantity	int(255)	To store product quantity	Not Null

CART DETAIL:

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
id	int(255)	To store id of items in cart	Primary Key
pname	varchar(255)	To store product name	Not Null
pimage	varchar(255)	To store product image	Not Null
pcost	int(255)	To store product cost	Not Null
date	date	To store date when items added to cart	Not Null
email	varchar(255)	To store email of person added the item to cart	Not Null
pquantity	int(255)	To store Product Quantity	Not Null

SERVICE DETAIL :

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
id	int(255)	To store service Id	Primary Key
s_name	varchar(255)	To store name of service	Not Null
s_description	varchar(255)	To store description about service	Not Null
s_cost	int(255)	To store cost of service	Not Null
s_img	varchar(255)	To store image of service	Not Null

SERVICE BOOKING:

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
s_id	int(255)	To store id of booked service	Primary Key
name	varchar(255)	To store name of person booked service	Null
email	varchar(255)	To store email of person booked service	Null
service_name	varchar(255)	To store name of service booked	Null
service_cost	int(255)	To store service cost	Null
service_date	date	To store the service booked date	Not Null
service_time	varchar(255)	To store service booked time	Not Null
requested_date	date	To store service requested date	Null

FEEDBACK:

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
product_id	int(255)	To store delivered product id on which review is given	Not Null
rating	varchar(255)	To store rating given by customer	Not Null
review	varchar(255)	To store message given by customer	Not Null
name	varchar(255)	To store the customer name	Null

PAYMENT DETAIL:

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
id	int(255)	To store unique id for payment	Primary Key
fname	varchar(255)	To customer first name	Not Null
lname	varchar(255)	To store customer last name	Not Null
email	varchar(255)	To store email of customer	Not Null
address	varchar(255)	To store address of customer	Not Null
pin	int(255)	To store pin code of customer	Not Null
amount	int(255)	To store total amount paid by customer	Not Null
o_date	date	To store product ordered date	Not Null
d_date	date	To store product delivered date	Null
r_within	date	To store the date within which delivered product must be returned if necessary	Null
return_reason	varchar(255)	To store product return reason, if it is returned	Null
pname	varchar(255)	To store array of products ordered	Not Null
pcost	varchar(255)	To store individual products price in array	Not Null
pquantity	varchar(255)	To store individual products quantity in array	Not Null
delivery_stat	varchar(255)	To store the delivery status of ordered products	Not Null
dcode	varchar(255)	To store delivery code for ordered products	Not Null
r_order	varchar(255)	To specify if order was returned or not	Null

DISCOUNT:

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
id	int(11)	To store the id of discount event	Primary Key
start_datetime	datetime(6)	To store start time of discount event	Not Null
end_datetime	datetime(6)	To store end time of discount event	Not Null
raquets_discount	int(255)	To store percentage of discount to be applied on racquet	Not Null
shuttles_discount	int(255)	To store percentage of discount to be applied on shuttles	Not Null
shoes_discount	int(255)	To store percentage of discount to be applied on shoes	Not Null
grippers_discount	int(255)	To store percentage of discount to be applied on grippers	Not Null

DISCOUNT_PRODUCTS:

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
i	int(255)	To store id for discounted products	Primary Key
p_id1	int(255)	To store the p_id field of table products for specific products	Foreign key
pname1	varchar(255)	To store the name of discounted products	Not Null
pdescript1	varchar(255)	To store the description of discounted products	Not Null
pcost1	int(255)	To store the cost of discounted products	Not Null
pcat1	varchar(255)	To store the category of discounted products	Not Null
imgg1	varchar(255)	To store the image of discounted products	Not Null
pquantity1	int(255)	To store the quantity discounted of products	Not Null

DELIVERY OR RETURN:

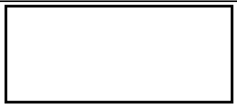
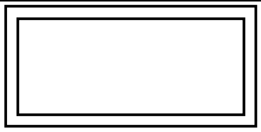
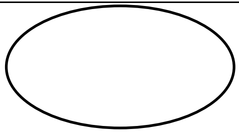
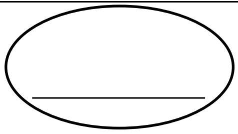
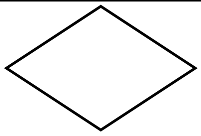
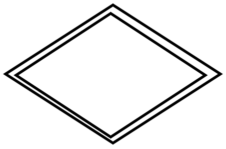
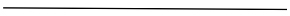
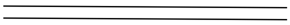
FIELD NAME	**DATATYPE**	**DISCRIPTION**	**CONSTRAINTS**
id	int(255)	To store id for package that is to be delivered or to be received	Primary Key
o_id	int(255)	It store the field o_id from from table payment on specified condition	Foreign Key
fname	varchar(255)	To store first name of customer	Not Null
lname	varchar(255)	To store last name of customer	Not Null
o_date	date	Stores the product ordered date	Not Null
amount	int(255)	Stores the amount of products	Not Null
address	varchar(255)	Stores the address where products to be delivered or returned	Not Null
delivery_stat	varchar(255)	Stores the delivery status of ordered products	Not Null
assigned_to	varchar(255)	Stores the name of delivery boy assigned for delivery	Not Null

DELIVERY BOY DETAILS:

FIELD NAME	DATATYPE	DISCRIPTION	CONSTRAINTS
id	int(255)	Stores the id of delivery boy	Primary Key
name	varchar(255)	Stores the name of the delivery boy	Not Null
email	varchar(255)	Stores the email of delivery boy	Not Null
date	date	Stores the stores registered date of delivery boy	Not Null
phnum	int(255)	Stores the phone number of delivery boy	Not Null
password	varchar(255)	Stores the password of delivery boy	Not Null
address	varchar(255)	Stores the address of delivery boy	Not Null

4.8 ENTITY RELATIONSHIP DIAGRAM(ER):

The basic object that the ER model represents in an entity which is a thing in a real world an pendent existence" An Entity is an object with a physical existence of with a conceptual existence Each entity has attributes particular properties that describes it Whenever an attribute of entity type refers to another entity sons relationships exists. The degree of relationship type is the number of participate in entity type. The cardinality ratio for binary relationship specifics a number of relationship exist like that an entities.

SYMBOL	DESCRIPTION
	Entity
	Weak Entity
	Attribute
	Key Attribute
	Relationship
	Weak Relationship
	Partial Participation
	Total Participation

An Entity Relationship Model (ERM), in software engineering is an abstract And conceptual representation of data. Entity relationship schema database modelling method, used to produce a type conceptual schema or schematic data model of a system, often relational database, and it is represented in a top-down fashion

4.8.1 Entity:

Entity is the thing which we want to store the information. It is an elementary basic building block of storing information about business process. An entity represents an object defined within the information system about which we want to store information. Entities are distinct things in the enterprise.

Relationship:

A Relationship is a named collection or association between entities or used to relate Two or more entities with some common attributes or meaningful interaction between the objects.

Attributes:

Attributes are the properties of the entities and relationship,descriptor of the entity Attributes are elementary pieces of inform attached to an entity.

Cardinality:

Specifies how many instances of an entity relate to one instance of another entity attribute are elementary pieces of information attached to an entity.

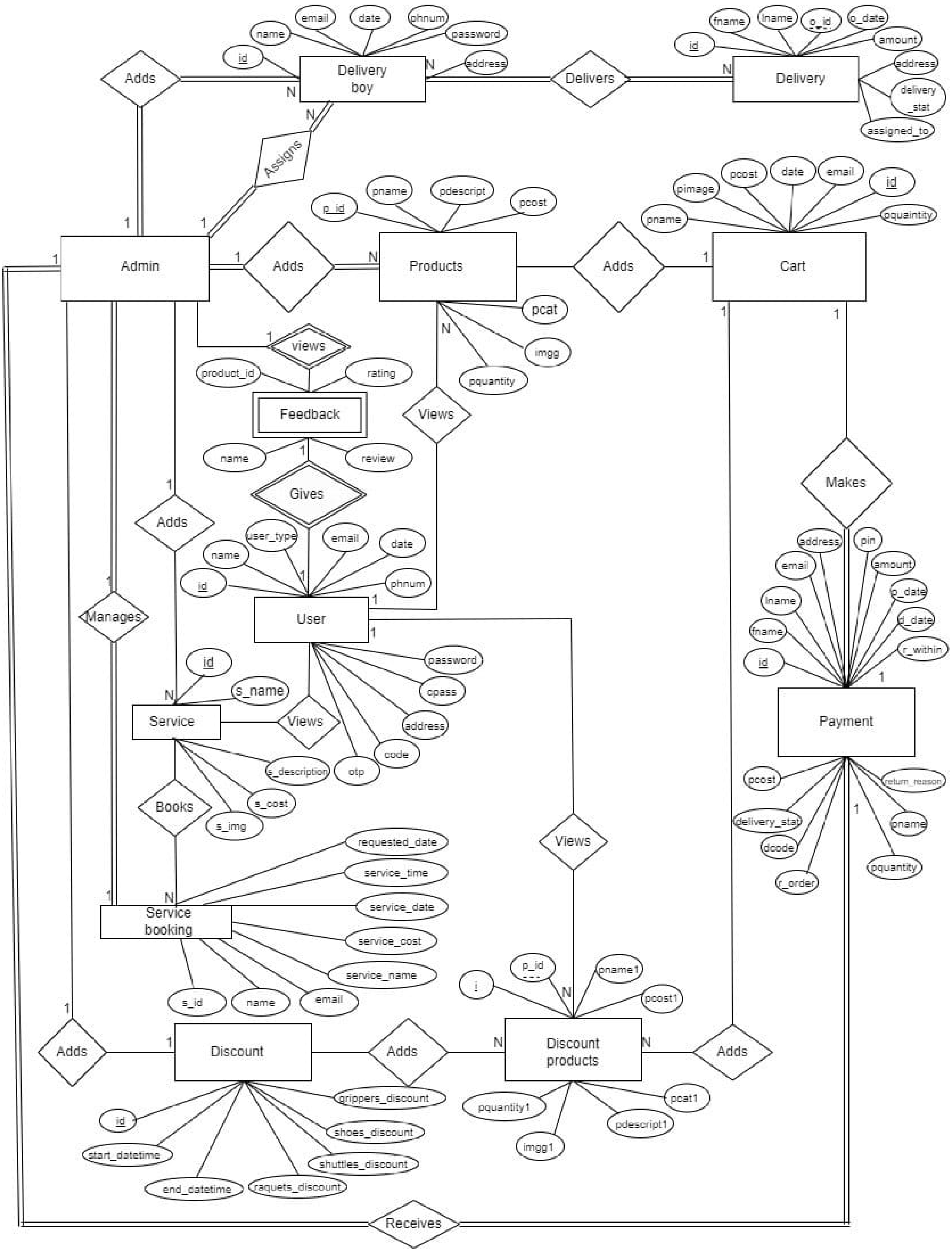
Weak Entity:

A weak entity does not have a primary key attribute and depends on other entity via a foreign key attribute.

4 Primary Key:

A primary key is either an existing table column or a column that is specifically generated by the database according to a defined sequence

Entity Relationship Diagram:



4.9 DATABASE ADMINISTRATION:

Database Administrator (DBA) in Database Management System (DBMS) is an IT who works on creating, maintaining. querying, and tuning the database of the organisation.

They are also responsible for maintaining data security and integrity.

4.9.1 System information:

A database management system (or DBMS) is essentially nothing more than a computerized data-keeping system. Users of the system are given facilities to perform several kinds of operations on such a system for either manipulation of the data in the database or the management of the database structure itself.

4.9.2 DBMS Configuration:

DBMS Server Configuration. The Ingres client server architecture allows multiple users access to databases through connections to one or more DBMS Server processes. The DBMS Server (iidbms) is a multi-threaded daemon process that performs asynchronous disk input and output.

4.9.3 Support software required:

1. Solar Wind Database Performance Analyzer:

Solar Wind Database Performance Analyzer pinpoints performance issues while providing expert advice with tuning advisor.

2. Db Visualizer:

Db Visualizer is a database management solution that helps businesses monitor and analyse the performance of multiple databases through various operating systems including Windows, macOS and Linux.

3. Manage Engine Application Manager:

Manage Engine Applications Manager is an enterprise-ready, easy-to use, and affordable application management solution that helps enterprises monitor their mission- critical applications and data canters effectively.

4. Oracle RDBMS:

Oracle Database is an RDBMS. An RDBMS that implements object- oriented features such as user-defined types, inheritance, and polymorphism is called an object relational database management system (ORDBMS).

4.9.4 Storage Requirement:

Primary Storage: The memory storage that is directly accessible to the CPU comes under this category backup.

Secondary Storage: Secondary storage devices are used to store data for future use or as backup

Tertiary Storage: Tertiary storage is used to store huge volumes of data. Since such storage devices are external to the computer system, they are slowest in speed. These storage devices are mostly used to take the backup of an entire system. Optical disks and magnetic tapes are widely used as teritary storage.

4.9.5 Backup and recovery:

It enables the creation of a duplicate instance or copy of a database in case the primary database crashes, is corrupted or is lost. Recovery is the process of restoring the database to a correct (consistent) state in the event of a failure. In other words, it is the process of restoring the database to the most recent consistent state that existed shortly before the time of system failure.

5.DETAILED DESIGN

5.1 INTRODUCTION

The detail design phase involves completing the product’s design. The design team works toward completion of the specifications for the product and its subassemblies, product elements, and manufacturing processes. Like the other phases of product development, detail design is an iterative process.

5.2 STRUCTURE OF THE SOFTWARE PACKAGE

A structure package bundles a set of packages at the top level of the package hierarchy. Structure packages do not contain any repository objects except for its own package interfaces and subpackages

5.3 MODULAR DECOMPOSITION AND COMPONENTS

5.3.1 Login Module

Input : Entering the Email, password.

Process: If Email & password is correct allows to enter main page or else error msg

Output: Allows viewing the main page and allows to avail the service.

5.3.2 Admin Module

Input : Entering the Email, password.

Process: Verifies the the Admin and stores the details

Output: Allows the admin to view admin control page

5.3.3 Shop Module

Input : Admin adds the new product to shop

Process: The products added are displayed to user interface on the Shop Module

Output: Allows the admin to update the product .

5.3.4 Checkout Module

Input : User can order the Product of his intrest

Process: Ordered product leads to choose mode of payment and generation of invoice

Output: displays the invoice of the product

5.3.5 Feedback Module

Input :,Message,and rating

Process: Entered message & rating is sent to admin

Output: The message sent by user is displayed to admin

5.3.6 Service Module

Input : Select date & type of service

Process: User can select type of service and date ,if its booked choose the other

Output: Allows the admin to view the booked the slot and cancel the slot further if necessary

6.PROGRAM CODE LISTING

6.1 DATABASE CONNECTION:

```
<?php

$conn = mysqli_connect('localhost','root', ",'badminto');

?>
```

6.2 AUTHORISATION /AUTHENTICATION

LOGIN CODE:

```
<?php
@include 'config.php';
session_start();
if (isset($_POST['submit'])) {
   $email = mysqli_real_escape_string($conn, $_POST['email']);
   $pass = mysqli_real_escape_string($conn, $_POST['password']);
   $select = "SELECT * FROM register WHERE email = '$email' AND password = '$pass'";
   $result = mysqli_query($conn, $select);
   $count = mysqli_num_rows($result);
 if ($count > 0) {
      $row = mysqli_fetch_assoc($result);
      $_SESSION['email'] = $row['email'];
      $_SESSION['user_type'] = $row['user_type'];
 if ($row['user_type'] == 'admin') {
         header('Location: ad.html');
         exit();
      } elseif ($row['user_type'] == 'user') {
         header('Location: home.php');
         exit();
      }
   } else {
      echo '<script>alert("Incorrect user details! Please enter correct credentials.");</script>';
   }
}
?>


<!DOCTYPE html>
<html lang="en">
  <head>
```

```
     <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
    <link
      href="https://fonts.googleapis.com/css2?family=Ubuntu:wght@300&display=swap"
      rel="stylesheet"
    />
    <title>Login Page </title>
    <link rel="stylesheet" href="login.css" />
     </head>
    <body>
    <div class="maincontainer">
      <img src="download.jpg" alt="" />
    </div>
     <div class="imgcontainer">
      <p
        style="
          font-family: 'Ubuntu-Bold', sans-serif;
          color: white;
          font-size: 30px;
        "
      >
         Login Page
      </p>
       <div class="line"></div>
    </div>
     <form action=""  method="post">
      </div>
      <div class="divcontainer">
        <div class="container">
          <input
            type="email"
            placeholder="Enter Email ID"
            name="email"
            required
          />
            <input
            type="password"
            id="pass"

            placeholder="Enter Password"
            name="password"
            required
          />
          <button type="submit" name="submit" ><span></span>Login</button>
          <a
            href="register.php"
            style="color: white; font-family: 'Ubuntu-Bold', sans-serif"
            >Not yet registered?</a>
           <a
            href="forgot.php"
            style="color: white; font-family: 'Ubuntu-Bold', sans-serif"
            >Forgot Password?</a>
        </div>
        </div>
    </form>
  </body>
</html>
```

FORGOTPASSWORD CODE:

```
<?php
    $db_name="mysql:host=localhost;dbname=badminto";$username="root";
    $password="";
    $conn=new PDO($db_name,$username,$password);
    use PHPMailer\PHPMailer\PHPMailer;
    use PHPMailer\PHPMailer\Exception;
    require ('PHPmailer/Exception.php');
    require ('PHPmailer/SMTP.php');
    require ('PHPmailer/PHPMailer.php');
    // Connection Created Successfully
    $errors = [];
    session_start();


    // if forgot button will clicked
    if (isset($_POST['forgot_password'])) {
        $email = $_POST['email'];
        $_SESSION['email'] = $email
        $emailCheckQuery = $conn->prepare("SELECT * FROM register WHERE email =?");
        $emailCheckQuery->execute([$email]);
       // if query  user_form
```

```
        if ($emailCheckQuery) {
            // if email matched
            if ($emailCheckQuery->rowCount() > 0) {
                $code = rand(999999, 111111);
                $updateQuery =$conn->prepare("UPDATE register SET code = ?, otp_expiry =
DATE_ADD(NOW(), INTERVAL 1 MINUTE) WHERE email = ?");
                $updateQuery->execute([$code,$email]);
                if ($updateQuery) {
                    $mail = new PHPMailer();
                    //$code = rand(999999, 111111);
                   // Store All Errors user_form
                    $errors = [];
                      // SMTP configuration
                      $mail->isSMTP();
                      $mail->Host = 'smtp.gmail.com';
                      $mail->SMTPAuth = true;
                      $mail->Username = 'dhanrajjpoojary25@gmail.com'; // Your email address
                      $mail->Password = 'kqbwbseeysdvftjg'; // Your email password
                      $mail->SMTPSecure = 'tls';
                      $mail->Port = 587;
                    // Sender and recipient information
                      $mail->setFrom('dhanrajjpoojary25@gmail.com', 'Badminton Hub');
                      $mail->addAddress($email);
                      // Email subject and body
                      $mail->isHTML(true);
                      $mail->Subject = 'Password Reset OTP Verification';
                      $mail->Body = "Your OTP for password reset is: $code<br>Please enter this
OTP in the verification form.";
                    $mail->send();
                      $_SESSION['message'] = $message;
                      header('location: verifyEmail.php');
                } else {
                    $errors['db_errors'] = "Failed while inserting data into database!";
                }
            }else{
                $errors['invalidEmail'] = "Invalid Email Address";
            }
        }else {
            $errors['db_error'] = "Failed while checking email from database!";
        }
    }
  // check if the reset button is clicked
if(isset($_POST['Resend'])) {
    $email = $_SESSION['email'];
        $_SESSION['email'] = $email;
        $emailCheckQuery = $conn->prepare("SELECT * FROM register WHERE email =?");
        $emailCheckQuery->execute([$email]);
       // if query  user_form
        if ($emailCheckQuery) {
            // if email matched
            if ($emailCheckQuery->rowCount() > 0) {
                 $code = rand(999999, 111111);
                 $updateQuery =$conn->prepare("UPDATE register SET code = ?, otp_expiry =
DATE_ADD(NOW(), INTERVAL 1 MINUTE) WHERE email = ?");
                $updateQuery->execute([$code,$email]);
                if ($updateQuery) {
                    $mail = new PHPMailer();
                    //$code = rand(999999, 111111);
                    // Store All Errors user_form
                    $errors = [];
                      // SMTP configuration
                      $mail->isSMTP();
                      $mail->Host = 'smtp.gmail.com';
                      $mail->SMTPAuth = true;
                      $mail->Username = 'dhanrajjpoojary25@gmail.com'; // Your email address
                      $mail->Password = 'kqbwbseeysdvftjg'; // Your email password
                      $mail->SMTPSecure = 'tls';
                      $mail->Port = 587;
                     // Sender and recipient information
                      $mail->setFrom('dhanrajjpoojary25@gmail.com', 'Badminton Hub');
                      $mail->addAddress($email);
                      // Email subject and body
                      $mail->isHTML(true);
                      $mail->Subject = 'Password Reset OTP Verification';
                      $mail->Body = "Your OTP for password reset is: $code<br>Please enter this
OTP in the verification form.";
                    $mail->send();
                      $_SESSION['message'] = $message;
                      header('location: verifyEmail.php');
                } else {
                    $errors['db_errors'] = "Failed while inserting data into database!";
                }
            }else{
                $errors['invalidEmail'] = "Invalid Email Address";
            }
        }else {
            $errors['db_error'] = "Failed while checking email from database!";
        }
    }
if(isset($_POST['verifyEmail'])){
    $_SESSION['message'] = "";
    $OTPverify = $_POST['OTPverify'];
    $verifyQuery = $conn->prepare("SELECT * FROM register WHERE code = ? AND
otp_expiry > NOW()");
    $verifyQuery->execute([$OTPverify]);
    if($verifyQuery){
        if($verifyQuery->rowCount() > 0){
               $newQuery = $conn->prepare("UPDATE register SET code = 0, otp_expiry = NULL
WHERE code = ?"); // set code to 0 and remove expiry time
            $newQuery->execute([$OTPverify]);
            header("location: newPassword.php");
        }else{
            $errors['verification_error'] = "Invalid Verification Code or otp expired";
        }
    }else{
        $errors['db_error'] = "Failed while checking Verification Code from database!";
    }
}
// change Password
if(isset($_POST['changePassword'])){
    $password =  $_POST['password'];
    $confirmPassword = $_POST['confirmPassword'];
     if(strlen($_POST['password'])<8){
        echo 'Password should conatain atleast 8 characters';
     }
     elseif (!preg_match('/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[^A-Za-z0-9]).{8,}$/',
$_POST['password'])) {
        echo "Password should contain at least one uppercase letter, one lowercase letter, one
digit, and one special character";
     }
     else {
        // if password not matched so
        if ($_POST['password'] != $_POST['confirmPassword']) {
            $error['password-error'] ='Password not matched';
        } else {
            $query=$conn->prepare("SELECT * from register WHERE password=?");
            $query->execute([$password]);
            if($query->rowCount()>0){
                $error['password']='password already exist';
            }else{
            $email = $_SESSION['email'];
            $updatePassword = $conn->prepare("UPDATE register SET password = ? WHERE
email = ?");
            $updatePassword->execute([$password,$email]) or die("Query Failed");
           //s session_unset($email);
            session_destroy();
            header('location: login.php');
        }
    }
    }
}
?>
```

6.3 DATA ADD/STORE/RETRIVE/UPDATE/DELETE PRODUCT AND CART :

products and cart:

```
<?php
session_start();
include 'config.php';
if (isset($_SESSION['email'])) {
  $userId = $_SESSION['email'];
   // Delete functionality
  if (isset($_POST['delete'])) {
    $delete = $_POST['cdelete'];
   // Retrieve the quantity from the cart1 table
    $select_query = "SELECT pquantity FROM cart1 WHERE pname = '$delete' AND email =
'$userId'";
    $select_result = mysqli_query($conn, $select_query);
    $cart_data = mysqli_fetch_assoc($select_result);
     if ($cart_data) {
      $quantity = $cart_data['pquantity'];
      $select_query = "SELECT pquantity FROM products WHERE pname = '$delete'";
      $result = mysqli_query($conn, $select_query);
      $select = mysqli_fetch_assoc($result);
      $updatedq = intval($select['pquantity']) + intval($quantity);
      $update_query = "UPDATE products SET pquantity = $updatedq WHERE pname =
'$delete'";
      mysqli_query($conn, $update_query);
    $delete_query = "DELETE FROM cart1 WHERE pname = '$delete' AND email =
'$userId'";
      mysqli_query($conn, $delete_query);
      // Rest of your code here...
    } else {
      echo" no rows are found";
    }
 }
 $query = "SELECT * FROM cart1 WHERE email='$userId'";
  $result = mysqli_query($conn, $query);
```

```
}
?>

<!DOCTYPE html>
<html lang="en">
 $ch = 0;
    // Check if a specific category is selected
    if (isset($_GET['search']) && isset($_GET['select_product'])) {
      $selectedCategory = $_GET['select_product'];
       if ($selectedCategory === 'all') {
        // Fetch all products from the database
        $query = "SELECT * FROM products";
      } else {
        // Perform a database query to retrieve products of the selected category
        $query = "SELECT * FROM products WHERE pcat = '$selectedCategory'";
      }
    } else {
      // Fetch all products from the database
      $query = "SELECT * FROM products";
}
     $result = mysqli_query($conn, $query);
     // Display the products
    if (mysqli_num_rows($result) > 0) {
    $productCount = 0; // Initialize product count
   echo '<div class="product-wrapper">'; // Start product wrapper
   while ($data = mysqli_fetch_assoc($result)) {
    // Display product information here
    $productId = $data['p_id']; // Unique identifier for each product
    ?>
   <div class="product-container" id="product<?php echo $productId; ?>">
      <div class="product-img">
        <img src="./image/<?php echo $data['imgg']; ?>" height="420" width="327" />
      </div>
      <div class="product-info">
        <div class="product-text">
          <h1 class="product-title"><?php echo $data['pname']; ?></h1>
          <p class="product-category">Type: <?php echo $data['pcat']; ?></p>
          <p class="product-description">
            <?php echo $data['pdescript']; ?>
          </p>
          <?php if ($data['pquantity'] == 0) : ?>
            <p class="product-stock out-of-stock">Out of Stock</p>
          <?php elseif ($data['pquantity'] < 10) : ?>
```

```
            <p class="product-stock limited-stock">Hurry! Only a few left</p>
          <?php else : ?>
            <p class="product-stock in-stock">In Stock: <?php echo $data['pquantity']; ?></p>
          <?php endif; ?>
           <?php if ($data['pquantity'] > 0) : ?>
            <div class="quantity-input-container">
              <label class="quantity-input-label" for="quantity-input-<?php echo $productId;
?>">Quantity:</label>
              <input type="number" id="quantity-input-<?php echo $productId; ?>"
class="quantity-input" name="quantity[<?php echo $productId; ?>]" value="1" min="1"
max="<?php echo $data['pquantity']; ?>" oninput="updateHiddenInput(this.value, '<?php echo
$productId; ?>')">
            </div>
          <?php endif; ?>
        </div>
        <div class="product-price-btn">
          <p class="product-price">
            <span class="currency-symbol">&#x20B9;</span><?php echo $data['pcost'];
?><span></span>
          </p>
            <?php if ($data['pquantity'] > 0) : ?>
            <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post" class="add-
to-cart-form">
              <input type="hidden" name="pname" value="<?php echo $data['pname']; ?>">
              <input type="hidden" name="pcost" value="<?php echo $data['pcost']; ?>">
              <input type="hidden" name="imgg" value="<?php echo $data['imgg']; ?>">
              <input type="hidden" name="quantity" value="1" min="1" max="<?php echo
$data['pquantity']; ?>" id="hiddenQuantity-<?php echo $productId; ?>">
              <button type="submit" name="addcart" class="add-to-cart-btn">Add To
Cart</button>
            </form>
          <?php endif; ?>
        </div>
      </div>
    </div>
    <?php
    $productCount++; // Increment product count
    // Check if two products have been displayed
    if ($productCount % 2 == 0) {
      echo '</div><div class="product-wrapper">'; // Start a new product wrapper for the next
row
    }
  }
```

```
 echo '</div>'; // Close the last product wrapper
 echo '<p style="color: white; position: relative; left: 90%; top: 20%">That\'s All!!!</p>';
} else {
  echo '<p style="color: white; position: relative; left: 90%; top: 20%">No Products To
Display!!!</p>';
}
 }
?>
</html>
```

6.4 SEARCH

```
<html lang="en">
  <head>
    <title>Products</title>
    <link

href="https://fonts.googleapis.com/css?family=Bentham|Playfair+Display|Raleway:400,500|S
uranna|Trocchi"
      rel="stylesheet"
    />
    <link rel="stylesheet" href="home.css" />
  </head>
  <div class="select">
  <label style="color: white; font-size: large; padding-right: 10px">What are you looking
for?</label>
  <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="get">
    <select name="select_product" id="select1">
    <option value="all" <?php if(!isset($_GET['select_product']) || $_GET['select_product']
=== 'all') echo 'selected'; ?>>All Products</option>

    <option value="Racquet" <?php if(isset($_GET['select_product']) &&
$_GET['select_product'] === 'Racquet') echo 'selected'; ?>>Racquets</option>
      <option value="Shuttle" <?php if(isset($_GET['select_product']) &&
$_GET['select_product'] === 'Shuttle') echo 'selected'; ?>>Shuttle</option>
      <option value="Shoes" <?php if(isset($_GET['select_product']) &&
$_GET['select_product'] === 'Shoes') echo 'selected'; ?>>Shoes</option>
      <option value="Gripper" <?php if(isset($_GET['select_product']) &&
$_GET['select_product'] === 'Gripper') echo 'selected'; ?>>Grippers</option>
    </select>
    <button class="Search-btn" type="submit" name="search">
      <span class="search-span"></span>Search
    </button>
  </form>
</div>

<div class="container">
```

```
   <?php
@include 'config.php';

$ch = 0;

// Check if a specific category is selected
if (isset($_GET['search']) && isset($_GET['select_product'])) {
  $selectedCategory = $_GET['select_product'];

  if ($selectedCategory === 'all') {
    // Fetch all products from the database
    $query = "SELECT * FROM products";
  } else {
    // Perform a database query to retrieve products of the selected category
    $query = "SELECT * FROM products WHERE pcat = '$selectedCategory'";
  }
} else {
  // Fetch all products from the database
  $query = "SELECT * FROM products";
}

$result = mysqli_query($conn, $query);

// Display the products
if (mysqli_num_rows($result) > 0) {
  while ($data = mysqli_fetch_assoc($result)) {
    // Display product information here
    $productId = $data['p_id']; // Unique identifier for each product
?>
</div>
</html>
```

6.5 DATA VALIDATION

REGISTRATION CODE:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
  <link
    href="https://fonts.googleapis.com/css2?family=Ubuntu:wght@300&display=swap"
    rel="stylesheet"
  />
  <title>Create Profile</title>
   <script>
    function validateForm() {
      var name = document.forms["myForm"]["name"].value;
      var email = document.forms["myForm"]["email"].value;
      var phnum = document.forms["myForm"]["phnum"].value;
      var password = document.forms["myForm"]["password"].value;
      var cpassword = document.forms["myForm"]["cpassword"].value;
     // Initialize an error variable
      var error = "";
     // Regular expression patterns
      var namePattern = /^[a-zA-Z\s]+$/;
      var emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
      var phonePattern = /^\d{10}$/;
      var passwordPattern = /^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%^&*()_+\-
=[\]{};':"\\|,.<>/?]).{8,}$/;
     if (!namePattern.test(name)) {
        error += "Name must contain only letters.\n";
        document.getElementById("nameError").innerText =
          "Name must contain only letters.";
      } else {
        document.getElementById("nameError").innerText = "";
      }
     if (!emailPattern.test(email)) {
        error += "Invalid email format. Must be example@gmail.com.\n";
        document.getElementById("emailError").innerText =
          "Invalid email format. Must be example@gmail.com.";
      } else {
        document.getElementById("emailError").innerText = "";
      }
     if (!phonePattern.test(phnum)) {
        error += "Phone number must be 10 digits.\n";
        document.getElementById("phnumError").innerText =
          "Phone number must be 10 digits.";
      } else {
        document.getElementById("phnumError").innerText = "";
      }
       if (!passwordPattern.test(password)) {
        error +=
          "Password must be at least 8 characters long and contain at least one lowercase letter,
one uppercase letter, one digit, and one special character.\n";
        document.getElementById("passwordError").innerText =
          "Password must be at least 8 characters long and contain at least one lowercase letter,
one uppercase letter, one digit, and one special character.";
      } else {
        document.getElementById("passwordError").innerText = "";
      }
       if (password !== cpassword) {
        error += "Passwords do not match.\n";
        document.getElementById("cpasswordError").innerText =
          "Passwords do not match.";
      } else {
        document.getElementById("cpasswordError").innerText = "";
      }
      if (error !== "") {
        alert(error);
        return false; // Prevent form submission
      }
    }
  </script>
</head>
<body>
  <div class="background-image"></div>
  <div class="maincontainer">
    <div class="imgcontainer">
      <p>Create profile</p>
      <form
        name="myForm"
        action="register.php"
        method="post"
        onsubmit="return validateForm()"
      >
        <div class="divcontainer">
          <div class="container">
            <input
              type="text"
              placeholder="Enter Username"
              name="name"
              required
            />
            <div class="error" id="nameError"></div>
             <input
              type="email"
              placeholder="Enter Email ID"
              name="email"
              required
            />
            <div class="error" id="emailError"></div>
            <input type="date" name="date" required />
           <input
              type="number"
              placeholder="Enter Phone number"
              name="phnum"
              required
            />
            <div class="error" id="phnumError"></div>
           <input
              id="pass1"
              type="password"
              placeholder="Enter Password"
              name="password"
              required
            />
            <div class="error" id="passwordError"></div>
           <input
              id="pass2"
              type="password"
              placeholder="Confirm Password"
              name="cpassword"
              required
            />
            <div class="error" id="cpasswordError"></div>
            <textarea
              name="address"
              id=""
              cols="50"
              rows="5"
              placeholder="Enter Address"
            ></textarea>
            <input type="submit" name="create" value="Create" />
            <a href="login.php">Already created?</a>
          </div>
        </div>
      </form>
    </div>
```

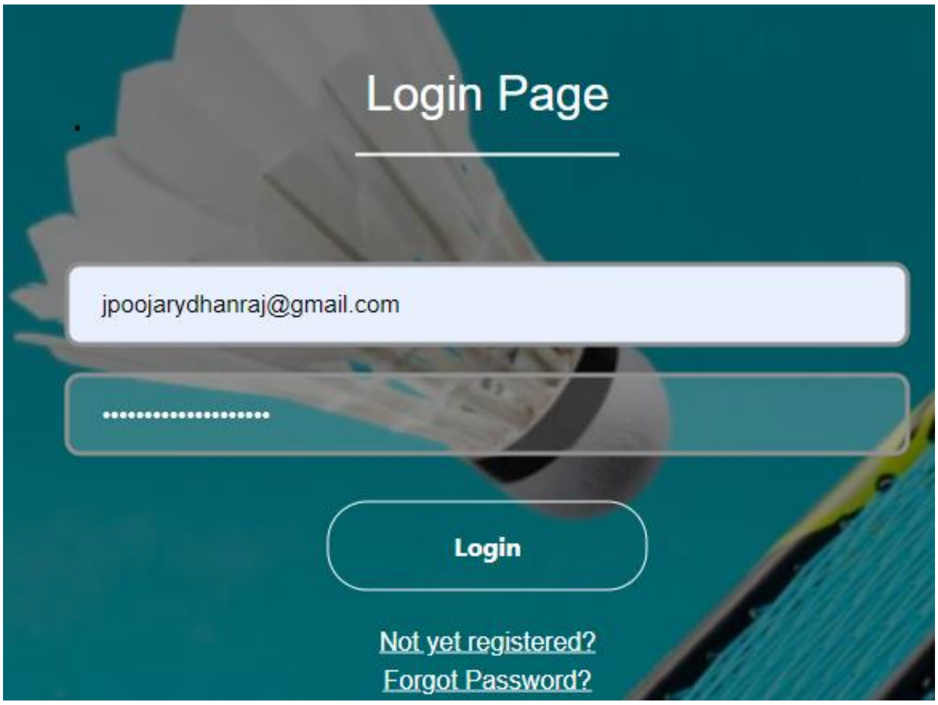
```
    </div>
  </body>
</html>

<?php
@include 'config.php';
if(isset($_POST['create'])){
  $name = mysqli_real_escape_string($conn, $_POST['name']);
   $email = mysqli_real_escape_string($conn, $_POST['email']);
   $date = mysqli_real_escape_string($conn,$_POST['date']);
   $number = mysqli_real_escape_string($conn,$_POST['phnum']);
   $pass = mysqli_real_escape_string($conn,$_POST['password']);
   $cpass = mysqli_real_escape_string($conn,$_POST['cpassword']);
   $address = mysqli_real_escape_string($conn,$_POST['address']);
   $user = 'user';
   $error = array(); // Initialize error array
// Check if email already exists
   $select = "SELECT * FROM register WHERE email = '$email' && password = '$pass'";
   $result = mysqli_query($conn, $select);
    if(mysqli_num_rows($result) > 0){
      $error['email'] = 'User already exists!';
   }
    //inserting into table
      $insert = "INSERT INTO
register(name,email,date,phnum,password,cpassword,address,user_type)
VALUES('$name','$email','$date','$number','$pass','$cpass','$address','$user')";
      mysqli_query($conn, $insert);
      $error ='<div class="Registration successful!!" role="alert">Success</div>';
header('location:home.php');
}
?>
```

7.USER INTERFACE

7.1 LOGIN

USER LOGIN:

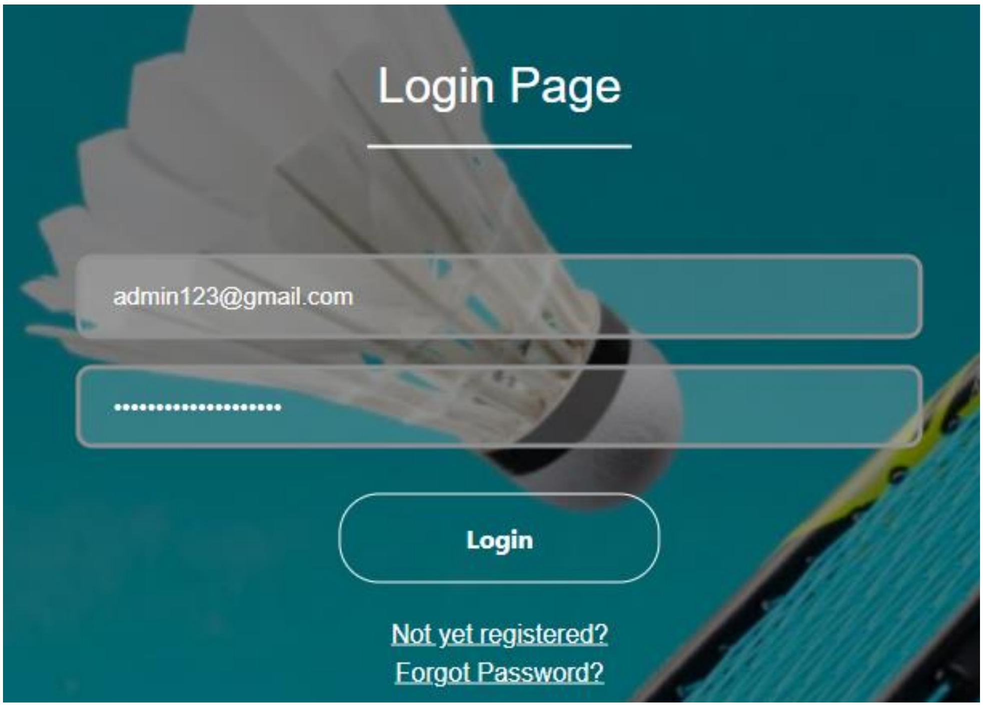


Login Page

jpoojarydhanraj@gmail.com

•••••••••••••••••••••

ADMIN LOGIN:

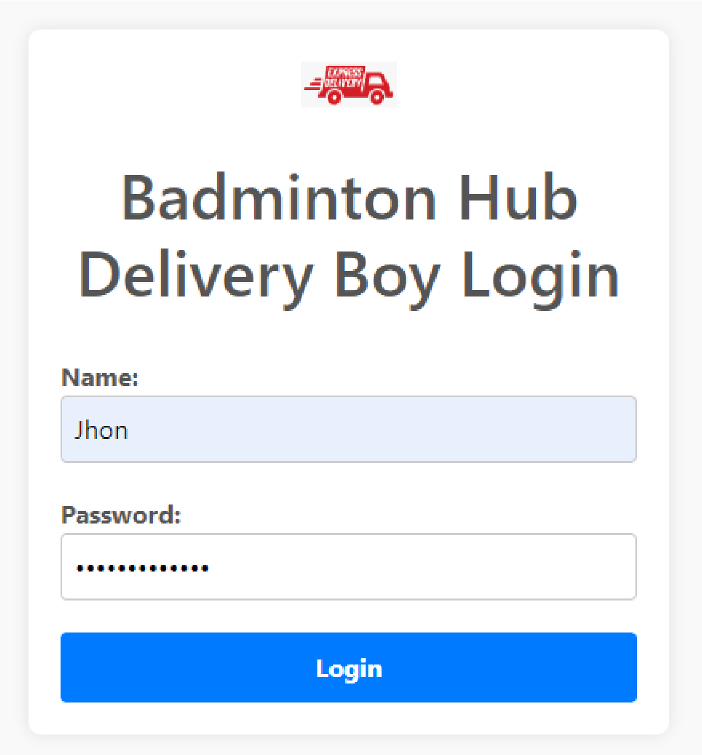


Login Page

admin123@gmail.com

•••••••••••••••••••••

DELIVERY BOY LOGIN:



Badminton Hub Delivery Boy Login

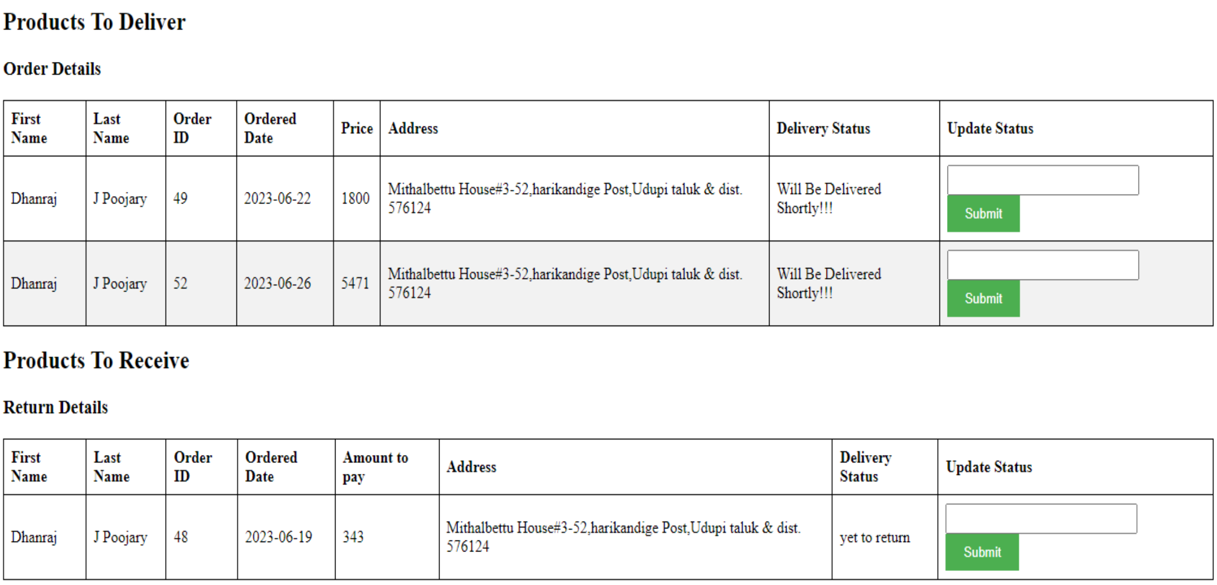
Name:

Jhon

Password:

•••••••••••••

DELIVERY BOY PAGE:



Products To Deliver

Order Details

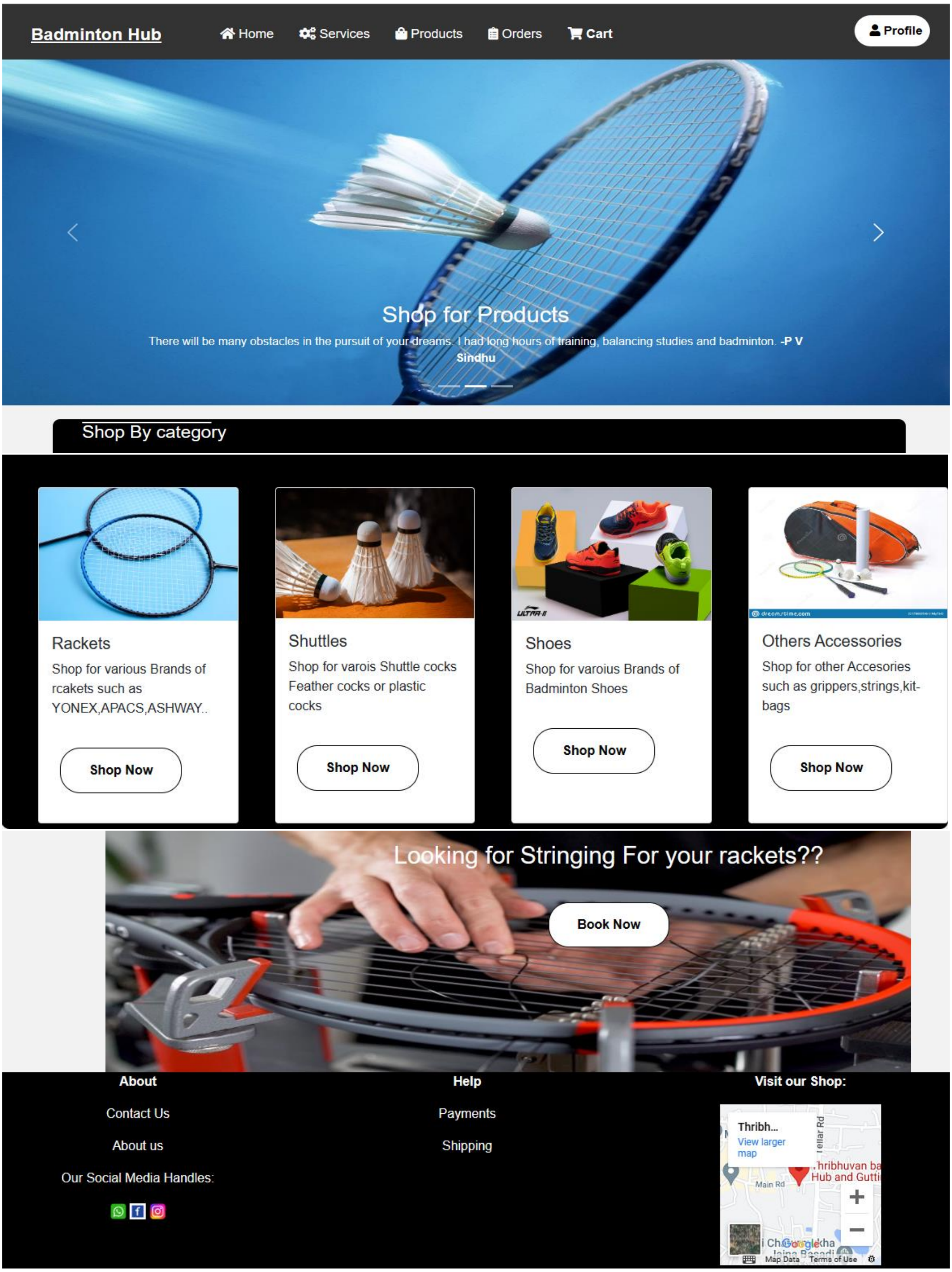
First Name	Last Name	Order ID	Ordered Date	Price	Address	Delivery Status	Update Status
Dhanraj	J Poojary	49	2023-06-22	1800	Mithalbettu House#3-52,harikandige Post,Udupi taluk & dist. 576124	Will Be Delivered Shortly!!!	Submit
Dhanraj	J Poojary	52	2023-06-26	5471	Mithalbettu House#3-52,harikandige Post,Udupi taluk & dist. 576124	Will Be Delivered Shortly!!!	Submit

Products To Receive

Return Details

First Name	Last Name	Order ID	Ordered Date	Amount to pay	Address	Delivery Status	Update Status
Dhanraj	J Poojary	48	2023-06-19	343	Mithalbettu House#3-52,harikandige Post,Udupi taluk & dist. 576124	yet to return	Submit

7.2 HOME PAGE



Shop for Products

There will be many obstacles in the pursuit of your dreams. I had long hours of training, balancing studies and badminton. **-P V Sindhu**

Shop By category

Rackets

Shop for various Brands of rcakets such as YONEX,APACS,ASHWAY..

Shuttles

Shop for varois Shuttle cocks Feather cocks or plastic cocks

Shoes

Shop for varoius Brands of Badminton Shoes

Others Accessories

Shop for other Accesories such as grippers,strings,kit-bags

Looking for Stringing For your rackets??

About

Contact Us

About us

Our Social Media Handles:

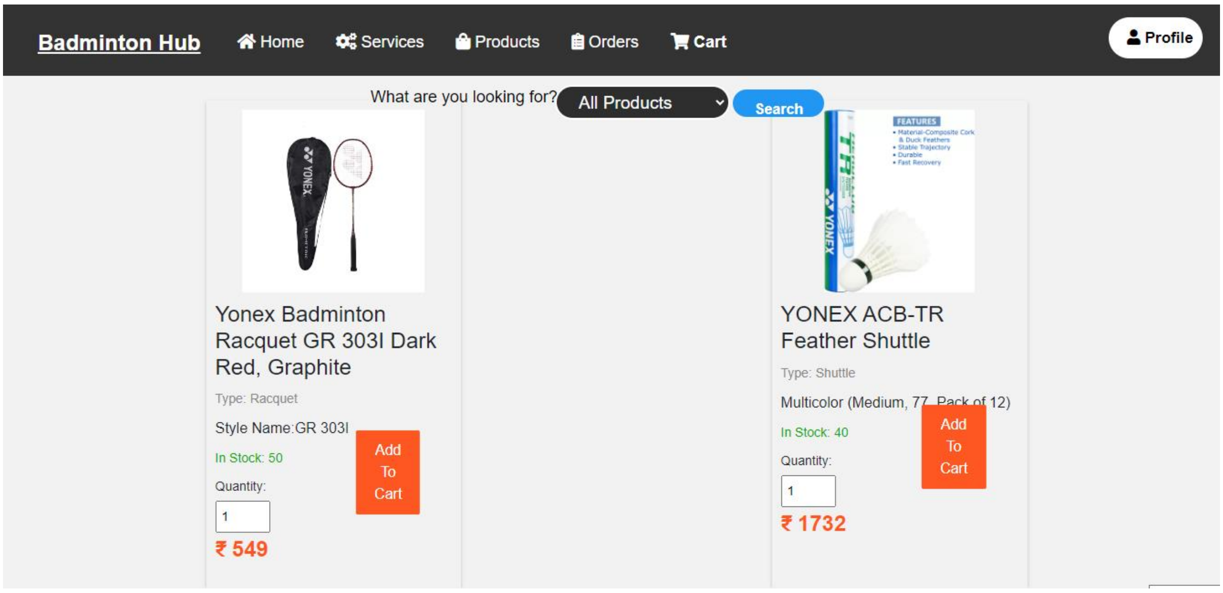
Help

Payments

Shipping

Visit our Shop:

7.3 DATA STORE/RETRIEVAL/UPDATE SHOP



Yonex Badminton Racquet GR 303I Dark Red, Graphite

Type: Racquet

Style Name:GR 303I

In Stock: 50

Quantity: 1

₹ 549

YONEX ACB-TR Feather Shuttle

Type: Shuttle

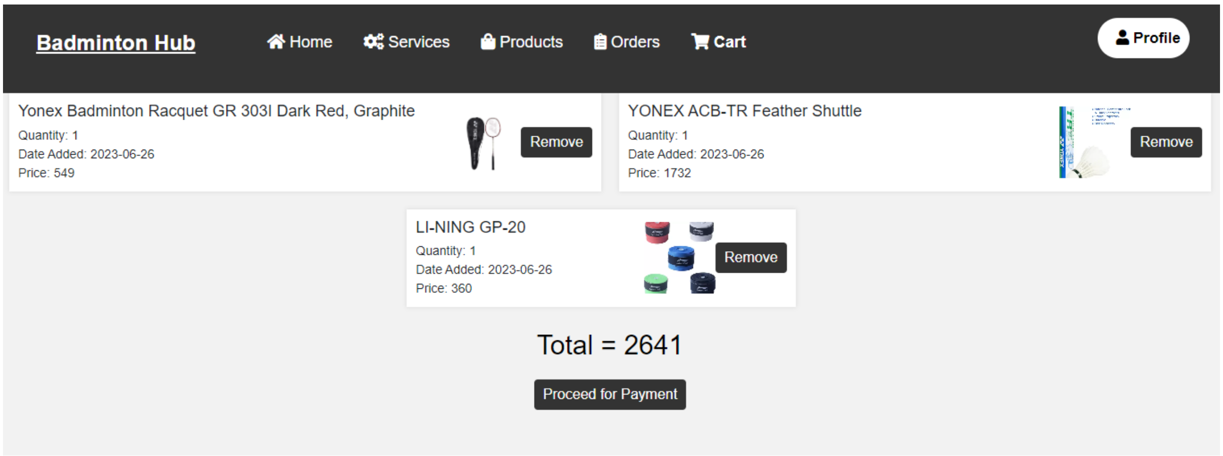
Multicolor (Medium, 77, Pack of 12)

In Stock: 40

Quantity: 1

₹ 1732

CART



Yonex Badminton Racquet GR 303I Dark Red, Graphite

Quantity: 1

Date Added: 2023-06-26

Price: 549

YONEX ACB-TR Feather Shuttle

Quantity: 1

Date Added: 2023-06-26

Price: 1732

LI-NING GP-20

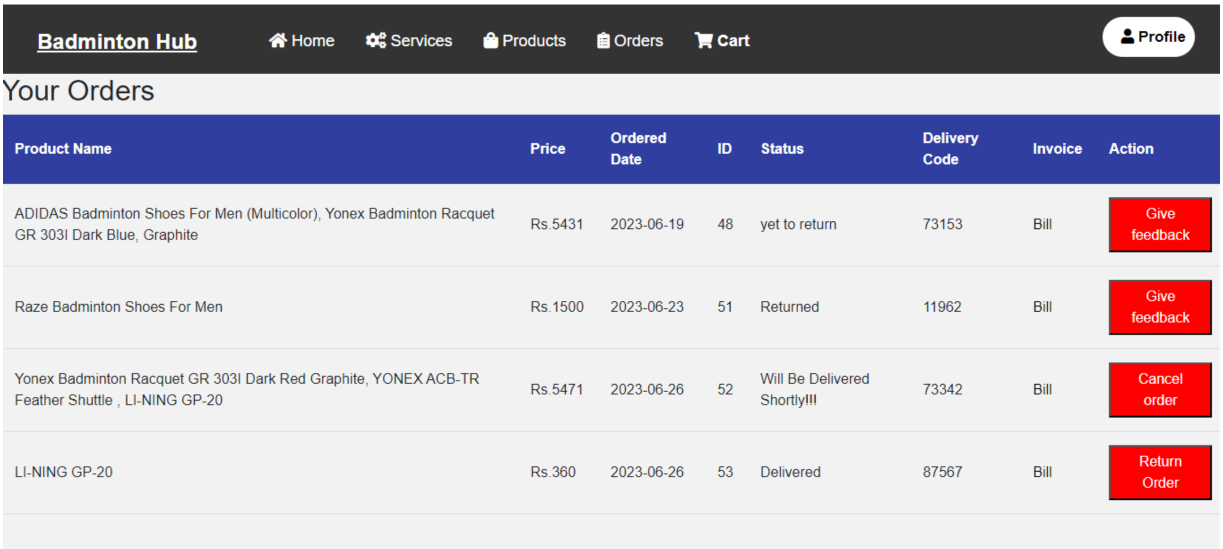
Quantity: 1

Date Added: 2023-06-26

Price: 360

Total = 2641

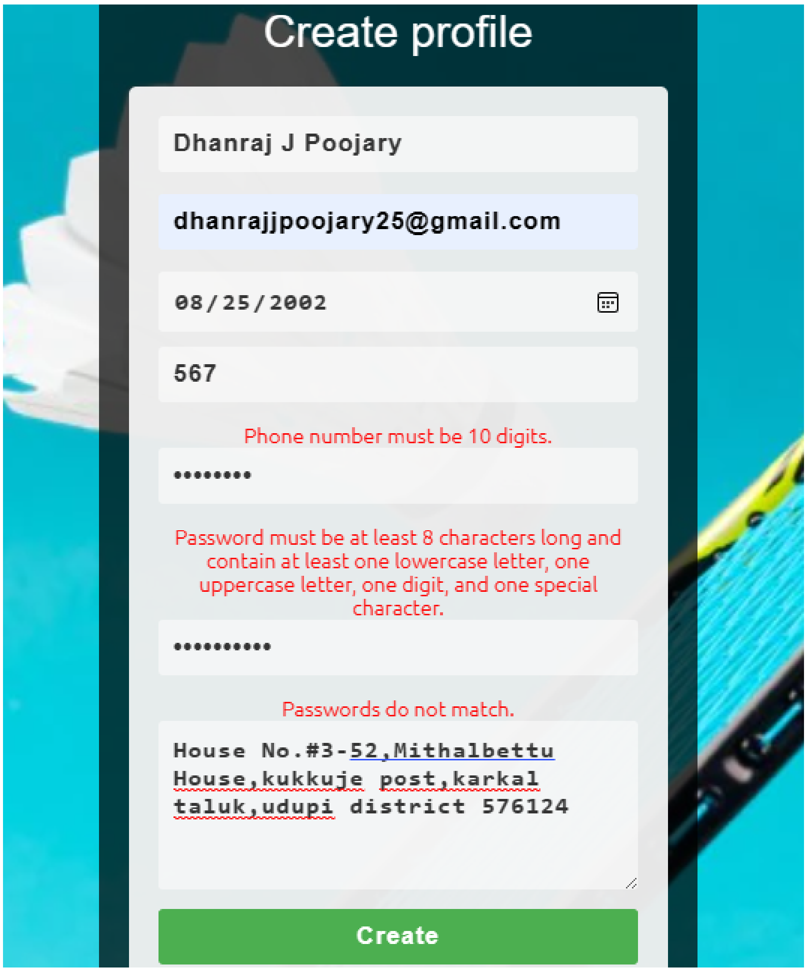
ORDER:



Your Orders

Product Name	Price	Ordered Date	ID	Status	Delivery Code	Invoice	Action
ADIDAS Badminton Shoes For Men (Multicolor), Yonex Badminton Racquet GR 303I Dark Blue, Graphite	Rs.5431	2023-06-19	48	yet to return	73153	Bill	Give feedback
Raze Badminton Shoes For Men	Rs.1500	2023-06-23	51	Returned	11962	Bill	Give feedback
Yonex Badminton Racquet GR 303I Dark Red Graphite, YONEX ACB-TR Feather Shuttle , LI-NING GP-20	Rs.5471	2023-06-26	52	Will Be Delivered Shortly!!!	73342	Bill	Cancel order
LI-NING GP-20	Rs.360	2023-06-26	53	Delivered	87567	Bill	Return Order

7.4 VALIDATION



Create profile

Dhanraj J Poojary

dhanrajjpoojary25@gmail.com

08/25/2002

567

Phone number must be 10 digits.

••••••••

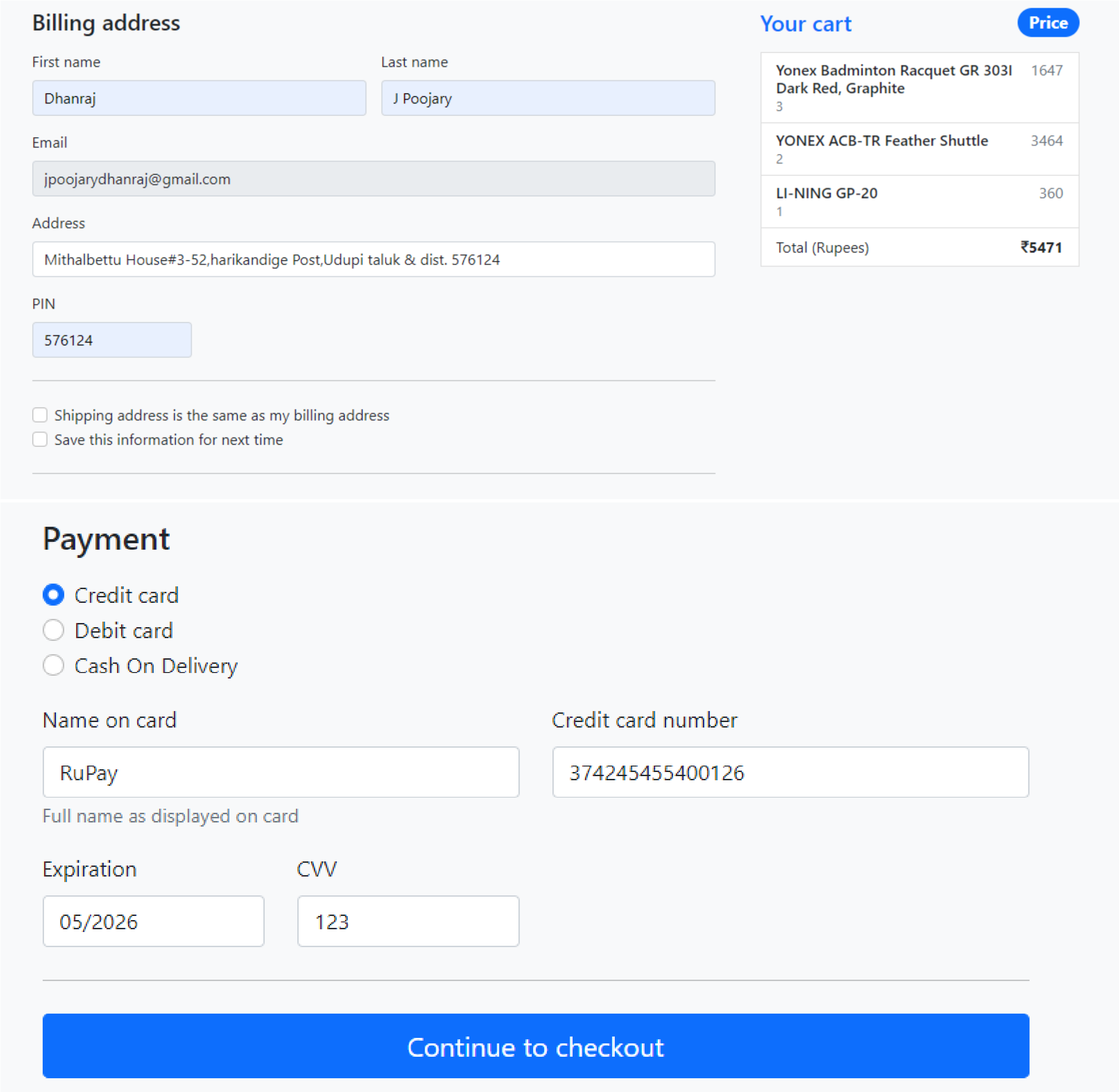
Password must be at least 8 characters long and contain at least one lowercase letter, one uppercase letter, one digit, and one special character.

••••••••••

Passwords do not match.

House No.#3-52,Mithalbettu House,kukkuje post,karkal taluk,udupi district 576124

7.5 PAYMENT



Billing address

First name: Dhanraj

Last name: J Poojary

Email: jpoojarydhanraj@gmail.com

Address: Mithalbettu House#3-52,harikandige Post,Udupi taluk & dist. 576124

PIN: 576124

□ Shipping address is the same as my billing address

□ Save this information for next time

Your cart

	Price
Yonex Badminton Racquet GR 303I Dark Red, Graphite 3	1647
YONEX ACB-TR Feather Shuttle 2	3464
LI-NING GP-20 1	360
Total (Rupees)	₹5471

Payment

● Credit card

○ Debit card

○ Cash On Delivery

Name on card: RuPay

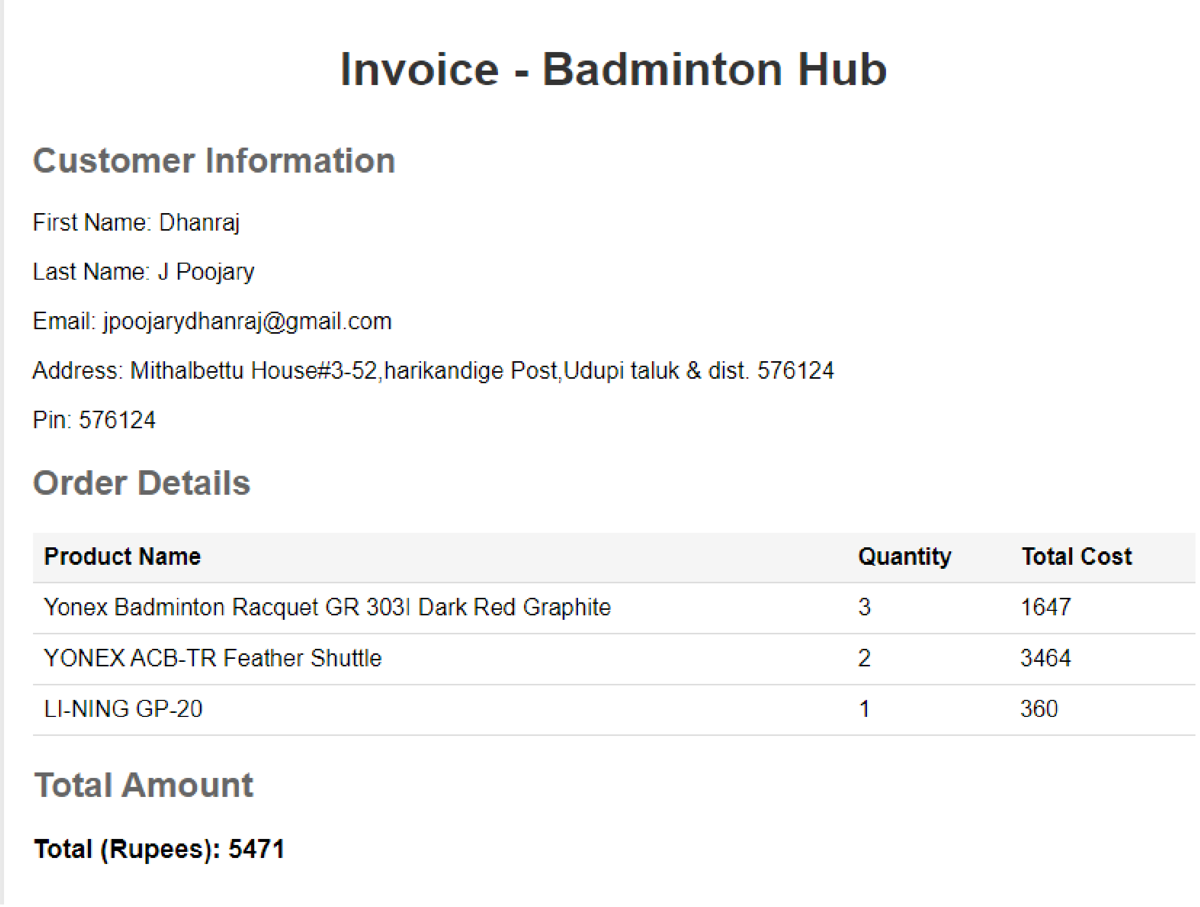
Full name as displayed on card

Credit card number: 374245455400126

Expiration: 05/2026

CVV: 123

7.6 REPORT



Invoice - Badminton Hub

Customer Information

First Name: Dhanraj

Last Name: J Poojary

Email: jpoojarydhanraj@gmail.com

Address: Mithalbettu House#3-52,harikandige Post,Udupi taluk & dist. 576124

Pin: 576124

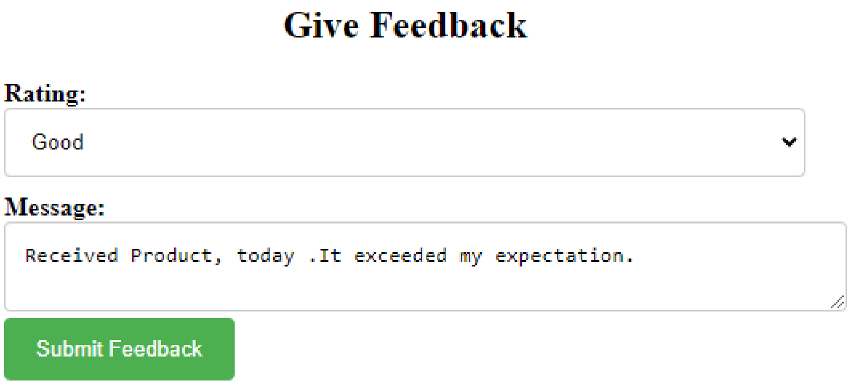
Order Details

Product Name	Quantity	Total Cost
Yonex Badminton Racquet GR 303I Dark Red Graphite	3	1647
YONEX ACB-TR Feather Shuttle	2	3464
LI-NING GP-20	1	360

Total Amount

Total (Rupees): 5471

7.7 FEEDBACK



Give Feedback

Rating:

Good

Message:

Received Product, today .It exceeded my expectation.

8.TESTING

8.1 Introduction

Testing is a process of detecting errors.Testing performs very special role for quantity assurance and for ensuring reliability of software.The result of testing used later on during maintenance too.

8.2 Test Reports

A test report is organized summary of testing objectives,activities and results.It is created and used to help stakeholders(product manager,analysts,testing team and developers)understand product quality and decide whether aproduct,feature or a defect resolution is on track of release.Beyond product quality,a test report also provides insight into quality of yyour testing and test automation activities.Organizations typically have four highlevel questions about

8.2.1 Unit Testing

Purpose of Unit Testing is to uncover errors in the smallest software unit in the routine.Each routine will be tested individually using block box-oriented tests.The programmer of each module will design set of test casesbfor that module is fully tested.Separately to ensure that it is working before being combined with another module.

8.2.2 Integrate Testing

This type of testing describes the integration stratergy and produces for the system.It gives the order in which will be developed and how they will be integrated.It also describes the specific tests that will be performed on the integration.Integration Testing of unit testing modules is necessary to ensure that modules interface currently with each other one module does not have undesirable effects on other module and sub modules combines to produce the desired function of major modules

The Purpose of testing is :

*To verify the interaction between objects.

*To verify the proper integration of all component of the software.

*To verify that all the requirement have been implemented correctly.

*To identify and ensure defects are addressed prior to deployed of the software

8.2.3 System Testing

System Testing,also referred to as system-level tests or system integration testing,is the process in which quality assurance(QA) team evaluates how the various components of an application interact together in the full,integrated system or application.System testing verifies that an applications performs tasks as designed.This step,a kind of black box testing,focuses on the functionality of an application.System testing,for example might check that every kind of user input produces the intended output across the application.It involoves end-to-end testing of a system to find the behavior of a system with respect to pectations

8.3 Test Cases

The testing phase involves the testing of the developed system using kinds of data.The customers test is developed system when changes are made according to the needs .The testing phase involves the testing of development system using various kinds of data test cases are to check output with different sets of inputs

Login Form

SL.NO	Test Condition	Expected Output	Result
1	When the admin clicks “LOGIN”(only if entered email & password is valid)	Admin page is displayed where he can manage all the activites of the user	SUCCESSFUL
2	When the customer clicks the “LOGIN”(only if the entered email & password is valid)	Customer page is displayed where he can view the product and also payment	SUCCESSFUL
3	When the delivery boy clicks “LOGIN”(only if entered name & password is valid)	Delivery page is displayed where he can performs the tasks assigned to him by admin	SUCCESSFUL
4	In LOGIN page if entered email & password is incorrect	Invalid Username and password	SUCCESSFUL

Registration Form

SL.NO	Test Condition	Excepted	Result
1	When user clicks “sign up”(if entered data is valid)	The user account is created & his details are successfully added to database	SUCCESSFUL
2	When the fields are empty and sign up is clicked	Validation occurs and get the message as registration failed	SUCCESSFUL

Forgot Password

SL.NO	Test Condition	Expected output	Result
1	When the user clicks forgot password link displayed on the login form	The user gets the page where they have to enter the email	SUCCESSFUL
2	Then on clicking the submit button	The otp is sent to email address & verified where it allows user to set new password	SUCCESSFUL

Feedback

SL.NO	Test Condition	Expected Output	Result
1	When the user writes message and gives rating and clicks on submit button	Feedback sent suceesully is the message displayed	SUCCESSFUL
2	When the fields are not entered and clicks submit	Displays the message to fill the fields which are empty	SUCCESSFUL

Payment Form

SL.NO	Test Condition	Excepted Output	Result
1	When the user clicks proceed checkout	It displays payment form which includes details related to payment	SUCCESSFUL
2	When pay now button is clicked	It displays the invoice	SUCCESSFUL

Conclusion

The project work titled “Badminton Hub” has been desgined and developed using php in such a way to make it flexible.It provides all necessary features as required by the users.

In today’s technology-oriented world where change takes place frequently,changes in the website may also be required to keep up with demands of the user.The Requirements may vary with demands of the user.So there will always be need to change some part of the project to keep with demands of the user.

Limitations

- Payments are done by online throught credit cards only
- User cannot track the order
- Booked services are only paid at the time of delivery and user has to visit our shop
- The software works only on Windows based Operating System

Future Scope

- Providing online payment option
- Tracking the purchased products

Abbreviations and Acronyms

SRS	Software Requirement Specification
OS	Operating System
SQL	Structured Query Language
RAM	Random Acess Memory
DFD	Data Flow Diagram
CFD	Control Flow Diagram

Bibiliography

Websites reffered:

https://openai.com/blog/chatgpt

https://www.geeksforgeeks.org/

https://colorlib.com/wp/free-bootstrap-admin-dashboard-templates/

https://youtu.be/1SnPKhCdlsU