

CAMPUS CENTRIC SECOND HAND EXCHANGE

*A Project Report submitted
in partial fulfillment of the requirements
for the award of the degree of*

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

1. Mohammed Afreen - 18B01A05G9
2. Sunnapuram Sai Durga – 18B01A05D7
3. Veluguri Lakshmi Sindhu - 18B01A05E6
4. Bathula Gowthami - 18B01A05F6
5. Tadikonda Tejaswi - 18B01A05D8

Under the esteemed guidance of

Mr. M. Ramesh Babu, M-Tech (Ph.D)

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)

(Approved by AICTE, accredited by NBA & NAAC, Affiliated to JNTU Kakinada)

BHIMAVARAM – 534 202

2021 – 2022

SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)

(Approved by AICTE, accredited by NBA & NAAC, Affiliated to JNTU Kakinada)

BHIMAVARAM – 534202

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

*This is to certify that the Project entitled "**Campus Centric Second Hand Exchange**", is being submitted by **Mohammed Afreen, Sunnapuram Sai Durga, Veluguri Lakshmi Sindhu, Bathula Gowthami, Tadikonda Tejaswi** bearing the **Regd. No. 18B01A05G9, 18B01A05D7, 18B01A05E6, 18B01A05F6, 18B01A05D8** in partial fulfillment of the requirements for the award of the degree of "**Bachelor of Technology in Computer Science & Engineering**" is a record of Bonafede work carried out by them under my guidance and supervision during the academic year **2021–2022** and it has been found worthy of acceptance according to the requirements of the university.*

Internal Guide

Head of the Department

External Examiner

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance has been a source of inspiration throughout the course of this project. I take this opportunity to express our gratitude to all those who have helped me in this project.

I wish to place my deep sense of gratitude to **Sri. K. V. Vishnu Raju, Chairman of SVES**, for his constant support on each and every progressive work of mine.

My deep sense of gratitude and sincere thanks to **Dr. G. Srinivasa Rao, Principal of SVECW** for being a source of an inspirational constant encouragement.

My deep sense of gratitude and sincere thanks to **Dr. P. Srinivasa Raju, Vice- Principal of SVECW** for being a source of an inspirational constant encouragement.

I deeply indebted and sincere thanks to **Dr. P. Kiran Sree, Head of the Department**, for his valuable advices in completing this project successfully.

My deep sense of gratitude and sincere thanks to **Mr. M. Ramesh Babu, Assistant Professor** for his unflinching devotion and valuable suggestion throughout my project.

MOHAMMED AFREEN	- 18B01A05G9
SUNNAPURAM SAI DURGA	- 18B01A05D7
VELUGURI LAKSHMI SINDHU	- 18B01A05E6
BATHULA GOWTHAMI	- 18B01A05F6
TADIKONDA TEJASWI	- 18B01A05D8

ABSTRACT

In light of the circumstances, people are now calling for a green civilization, with resource conservation as a top priority. In today's world, technology has a significant impact on our daily life. The current generation enjoys the great convenience that sophisticated technology brings to us. Using it to deal with real problems is what we ought to do. It serves as an incentive for creating a professional on-campus website for trading used products. This project aims to create and implement a second-hand trading website for the college, providing a platform for students to effortless trading of old products. It not only saves students money but also helps to safeguard the environment. The influence that advanced technology provided had motivated us to develop a professional on-campus website for trading old items.

CONTENTS

S.NO	TOPIC	Page No
1	Introduction	1
2	System Analysis 2.1 Existing System 2.2 Proposed System 2.3 Feasibility Study	3 3 3
3	System Requirements Specification 3.1 Software Requirements 3.2 Hardware Requirements	5 5
4	System Design 4.1 Introduction 4.2 UML Diagrams 4.3 ER Diagram	6 7 19
5	System Implementation 5.1 Introduction 5.2 Project Modules 5.3 Screens	21 22 23
6	System Testing 6.1 Introduction 6.2 Testing Methods 6.3 Test Cases	36 37 39
7	Conclusion	45
8	Bibliography	46
9	Appendix 9.1 Introduction to Java 9.2 Introduction to HTML 9.3 Introduction to CSS 9.4 Introduction to JavaScript and TypeScript 9.5 Introduction to MySQL 9.6 Introduction to Angular, Bootstrap, Rest API and Hibernate	48 50 52 53 55 56

LIST OF FIGURES

S.NO	UML DIAGRAMS	PAGE NO.
1	Use case diagram	10
2	Class Diagram	12
3	Sequence Diagram	14
4	Activity Diagram	16
5	Collaboration Diagram	18
6	ER Diagram	20

LIST OF TABLES

S.NO	TABLES	PAGE NO.
1	Register Student Test cases	39
2	Login Student Test Cases	40
3	Upload Book Test Cases	41
4	Upload Product Test Cases	42
5	Search Test Cases	43
6	Feedback Test Cases	44

CHAPTER - 1
INTRODUCTION

For international students or others, when first coming to the United State, everything is new. There are several problems as a beginner and foreigner here. For example, renting an apartment, getting new furniture, buying new textbooks and so on. Since the culture and custom vary from other countries, the immigrant finds that these tasks are disasters. To start with, the new furniture or textbooks are very expensive, and it is even not worth buying a new one.

However, there are no known platforms, online or offline, which allow people to trade used items. Even searching the results online, the buyer still could not know whom to trust. Even if the buyer could find reliable websites later, transportation is still a big problem. Therefore, this idea evolves, as a computer science project, to develop a platform focused on the CSUN. With the supervision of authorities from school, the safety issues could also be guaranteed.

According to situations that we are advocating green civilization, saving resources as much as we can should a vital problem. There are over 3,900 international students our campus currently that means students come and go constantly so that there might be giant old items that cannot be taken away. We all know that the books are expensive and could easily cost 1,000 dollars per year. A second-hand one could save students money.

Besides personal problem, nearly 50 percent of municipal waste generated in Finland in 2014 was burned. The percentage of burning has risen very fast. Ten years ago, just 12 percent of waste was burned. Disposal at landfill sites has, in turn, fallen more or less at the same rate. In practice, all municipal solid waste delivered for burning reduces the waste in landfill sites. Just under one-fifth of the total volume of waste was disposed at landfill sites. Municipal waste as an additional resource in energy production is running low.

This emphasis on the politics of sanitary reform and technologies of sewage treatment obscures the ways in which urban nature has been incorporated into technological waste-disposal strategies. Natural systems have been typically seen as outside urban networks: as a receiving environment that is subject to pollution by waste-waters, or as a water supply requiring purification by artificial methods such as filtration or chlorination.

As our country pushes forward the Internet, Internet industry flourishes, and online shopping has become an indispensable part of people's life. Most young people have the experience of shopping on the Internet. It is because there are so swollen and broad market prospects that make more and more entrepreneurship people can get through online platform.

An environment-friendly society have been advocated; everyone should start from resource conservation. In school, especially many second-hand goods are thrown away in universities and colleges, which both damages the environment, and wastes of resources. For such rich second-hand resources, we can take reasonable way to handle it, build an Internet platform, make full use of the mature technology and social environment, let the students used resources can be reasonable shared and effectively used so as to make resources communication on campus, save cost, and make waste useful. So, this website creates campus second-hand trading platform is feasible and has certain practical value.

Existing websites do have fundamental customers. There are still some limitations. Since these platforms are open to the whole citizens, we cannot easily find what we need for school and the transportation is not convenient. We are trying to not move heavy items back to home. A trade application which just focuses on a single university is necessary.

Nowadays, advanced technology is greatly influencing our lives. People are taking smart phones everywhere for chatting with friends, paying for items, ordering take-out and so on. It has become a custom to university students. That provides motivations for developing a professional on campus application for trading old items.

The main goal of the application would be achieved to help the students deal with daily problems and satisfy them as most as possible.

CHAPTER - 2
SYSTEM ANALYSIS

2.1 Existing System

- Existing websites do have fundamental customers.
- Since these platforms are open to the whole citizens, we cannot easily find what we need for campus.

2.2 Proposed System

- A trade website which just focuses on a single university.
- This website is a on campus massive potential market of second-handing trading to save money and also to help in protecting the environment.
- The students usually generate a great deal of spare items such as books, drafters, chairs, etc. can upload their items in this website and sell them.

2.3 Feasibility Study

Feasibility Study in Software Engineering is a study to evaluate feasibility of proposed project or system. Feasibility study is one of stage among important four stages of Software Project Management Process. As name suggests feasibility study is the feasibility analysis or it is a measure of the software product in terms of how much beneficial product development will be for the organization in a practical point of view. Feasibility study is carried out based on many purposes to analyse whether software product will be right in terms of development, implantation, contribution of project to the organization etc.

There are three types of feasibility which are equally important are:

- Technical feasibility
- Economic feasibility
- Behavioral feasibility

Technical Feasibility

Technical feasibility deals with the existing technology, software and hardware requirements for the proposed system. The proposed system "Campus Centric Second Hand Exchange" is planned to run on Java. Thus, the project is considered technically feasible for the development. The work for the project can be done with current equipment, existing software technology and available personnel. Hence the proposed system is technically feasible.

Economic Feasibility

This method is most frequently used for evaluating the effectiveness of a Java. It is also called as benefit analysis. In this project "Campus Centric Second Hand Exchange" is developed on current equipment, existing software technology. Since the required hardware and software for developing the system is already available in the organization, it does not cost must developing the proposed system.

Behavioral Feasibility

This project has been implemented by Java and it satisfies all conditions and norms of the organization and the users. This proposed system "Campus Centric Second Hand Exchange" has much behavioral feasibility because users are provided with a better facility.

CHAPTER - 3

System Requirements specification

3.1 Software Requirements:

Tools Used:

- Eclipse
- Visual Studio Code

Technologies Used:

- Frontend:
 - Angular
 - HTML
 - CSS
 - Bootstrap
- Backend:
 - Rest API
 - Hibernate

Database:

- MySQL

3.2 Hardware Requirements:

- Operating System: Windows 8 and above
- RAM: 4GB and above
- Processor: Intel Core I3 and above
- Hard disk space: 500GB and above

CHAPTER - 4

System Design

4.1 INTRODUCTION

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

System Analysis is the process that decomposes a system into its component pieces for the purpose of defining how well those components interact to accomplish the set requirements. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from problem domain to the solution domain. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The output of this phase is the design document. This document is similar to a blue print or plan for the solution, and is used later during implementation, testing and maintenance.

The design activity is often divided into two separate phase-system design and detailed design. System design, which is sometimes also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats, output formats, as well as the major modules in the system and their specifications are decided.

A design methodology is a systematic approach to creating a design by application of set of techniques and guidelines. Most methodologies focus on system design. The two basic principles used in any design methodology are problem partitioning and abstraction. A large system cannot be handled as a whole, and so for design it's partitioned into smaller systems. Abstraction is a concept related to problem partitioning. When partitioning is used during design, the design activity focuses on one part of the system at a time. Since the part being designed interacts with other parts of the system, a clear understanding of the interaction is essential for properly designing the part.

4.2 UML Diagrams

Unified Modeling Language (UML) is general purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

Use of UML Diagrams:

- Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.
- Businessmen do not understand code. So, UML becomes essential to communicate with non-programmers' essential requirements, functionalities and processes of the system.
- A lot of time is saved down the line when teams are able to visualize processes, user interactions and static structure of the system.

UML is linked with object-oriented design and analysis. UML makes the use of elements and forms associations between them to form diagrams. Diagrams in UML can be broadly classified as:

1. Structural Diagrams – Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.
2. Behaviour Diagrams – Capture dynamic aspects or behaviour of the system. Behaviour diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

Object Oriented Concepts Used in UML –

1. **Class** – A class defines the blue print i.e. structure and functions of an object.
2. **Objects** – Objects help us to decompose large systems and help us to modularize our system. Modularity helps to divide our system into understandable components so that we can build our system piece by piece. An object is the fundamental unit (building block) of a system which is used to depict an entity.
3. **Inheritance** – Inheritance is a mechanism by which child classes inherit the properties of their parent classes.
4. **Abstraction** – Mechanism by which implementation details are hidden from user.
5. **Encapsulation** – Binding data together and protecting it from the outer world is referred to as encapsulation.
6. **Polymorphism** – Mechanism by which functions or entities are able to exist in different forms.

UML Diagrams is a rich visualizing model for representing the system architecture and design. These diagrams help us to know the flow of the system.

Some of them are:

- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Class Diagram

USECASE DIAGRAM:

A Use Case Diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. Interaction among actors is not shown on the use case diagram.

Use cases:

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

Actors:

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

Three relationships among use cases are used often in practice.

- **Include:**

In one form of interaction, a given use case may include another. "Include is a Directed Relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case. This is useful for extracting truly common behaviors from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "«include»". There are no parameters or return values.

- **Extend:**

In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "«extend

- **Identified Use Cases:**

The “user model view” encompasses a problem and solution from the preservative of those individuals whose problem the solution addresses. The view presents the goals and objectives of the problem owners and their requirements of the solution. This view is composed of “use case diagrams”. These diagrams describe the functionality provided by a system to external integrators. These diagrams contain actors, use cases, and their relationships.

In our example the preliminaries are,

Actors: Admin, Student

Use cases: login, register, view orders, search items, place order, payment, feedback, add category, upload items, update profile.

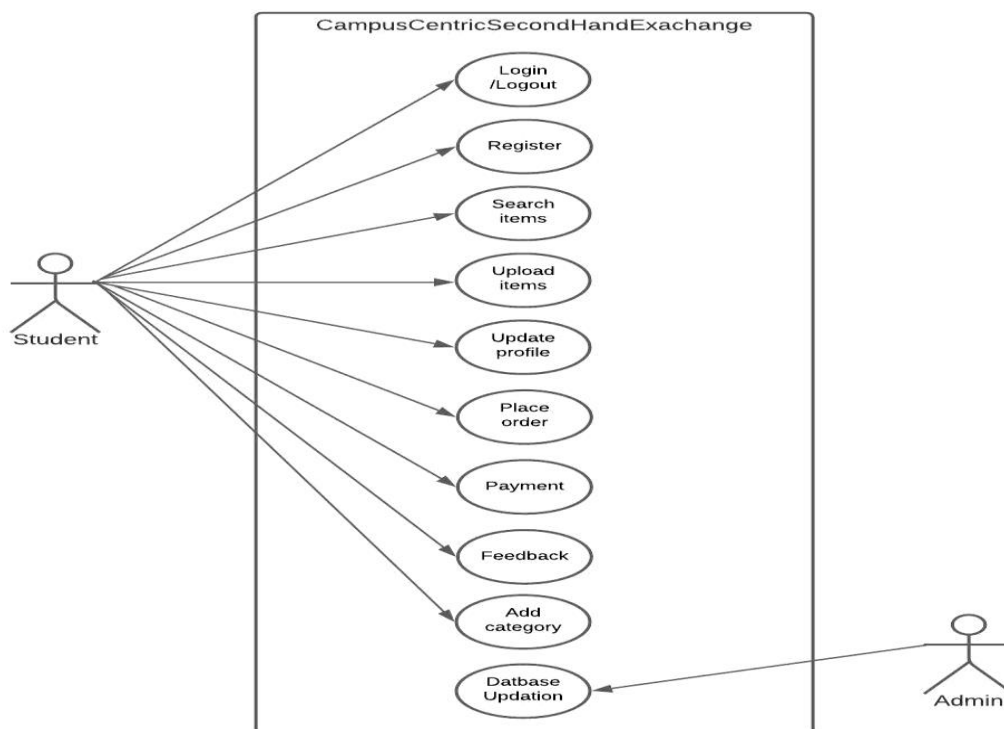


Fig:1 (Use case diagram of Campus Centric Second Hand Exchange)

CLASS DIAGRAM:

The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code. It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

The class diagram is made up of three sections:

- **Upper Section:** The upper section encompasses the name of the class. A class is a representation of similar objects that shares the same relationships, attributes, operations, and semantics. Some of the following rules that should be taken into account while representing a class are given below:
 - a) Capitalize the initial letter of the class name.
 - b) Place the class name in the centre of the upper section.
 - c) A class name must be written in bold format.
 - d) The name of the abstract class should be written in italics format.
- **Middle Section:** The middle section constitutes the attributes, which describe the quality of the class. The attributes have the following characteristics:
 - a) The attributes are written along with its visibility factors, which are public (+), private (-), protected (#), and package (~).
 - b) The accessibility of an attribute class is illustrated by the visibility factors.
 - c) A meaningful name should be assigned to the attribute, which will explain its usage inside the class.

- **Lower Section:** The lower section contains methods or operations. The methods are represented in the form of a list, where each method is written in a single line. It demonstrates how a class interacts with data.

CLASSES: Student, Book, Product, Order, Feedback

The attributes are id, name, mobile, password, qrcode.

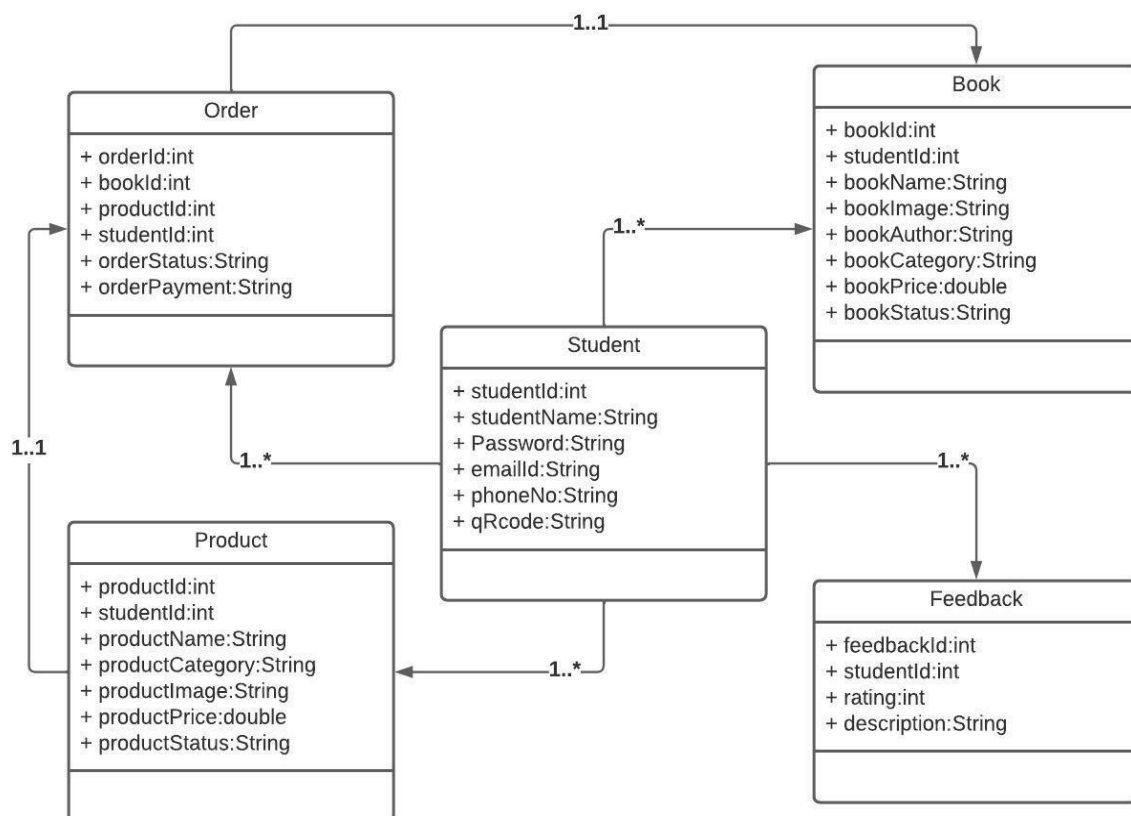


Fig:2 (Class diagram of Campus Centric Second Hand Exchange)

SEQUENCE DIAGRAM:

A **sequence diagram** shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

Sequence Diagram Notations:

- **Actors –**

An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.

- **Lifelines –**

A lifeline is a named element which depicts an individual participant in a sequence diagram. So, basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.

The standard in UML for naming a lifeline follows the following format –

Instance Name: Class Name

- **Messages –**

Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

- a) Synchronous messages
- b) Asynchronous Messages
- c) Create message
- d) Delete Message
- e) Self-Message
- f) Reply Message
- g) Found Message
- h) Lost Message

Guards – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

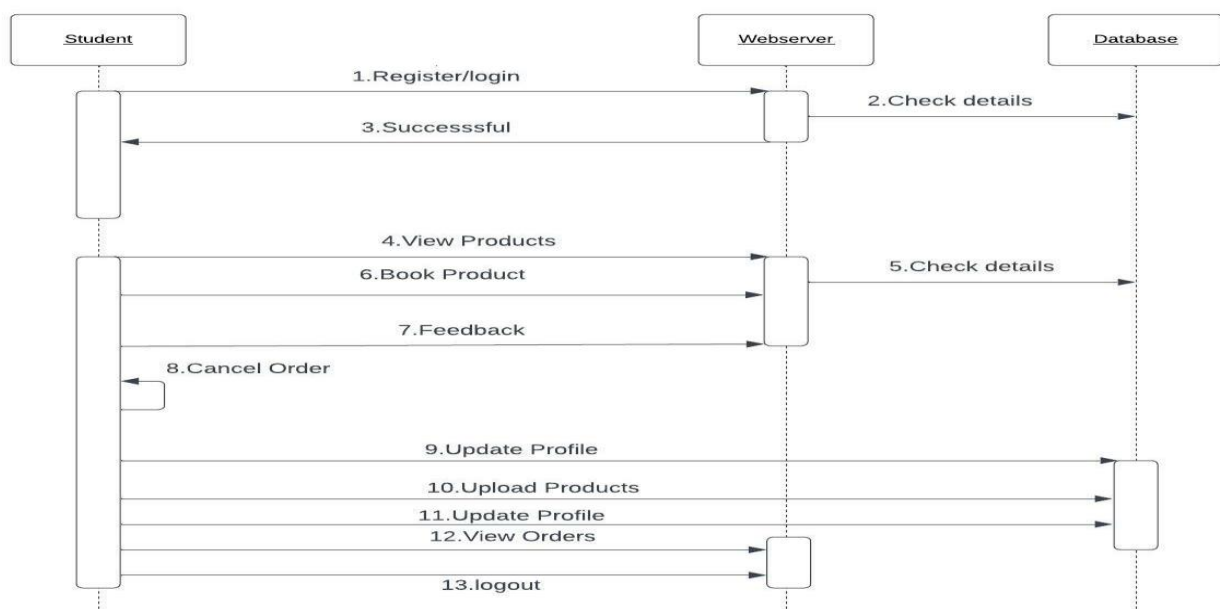


Fig:3 (Sequence Diagram of Campus Centric Second Hand Exchange)

ACTIVITY DIAGRAM

Activity Diagram is another diagram in UML to describe dynamic aspects of the system. Activity Diagram is basically a flow chart to represent the flow from one activity.

Activity Diagram Notations

- **Initial State** – The starting state before an activity takes place is depicted using the initial state.
- **Action or Activity State** – An activity represents execution of an action on objects or by objects. We represent an activity using a rectangle with rounded corners.
- **Action Flow or Control flows** – Action flows or Control flows are also referred to as paths and edges. They are used to show the transition from one activity state to another.
- **Decision node and Branching** – When we need to make a decision before deciding the flow of control, we use the decision node.
- **Decision node and Branching** – When we need to make a decision before deciding the flow of control, we use the decision node.
- **Guards** – A Guard refers to a statement written next to a decision node on an arrow sometimes within square brackets.
- **Fork** – Fork nodes are used to support concurrent activities.
- **Join** – Join nodes are used to support concurrent activities converging into one. For join notations we have two or more incoming edges and one outgoing edge.
- **Merge or Merge Event** – Scenarios arise when activities which are not being executed concurrently have to be merged. We use the merge notation for such scenarios. We can merge two or more activities into one if the control proceeds onto the next activity irrespective of the path chosen.
- **Swimlanes** – We use Swimlanes for grouping related activities in one column. Swimlanes group related activities into one column or one row. Swimlanes can be vertical and horizontal. Swimlanes are used to add modularity to the activity diagram.

- **Time Event** – We can have a scenario where an event takes some time to complete. We use an hourglass to represent a time event.
- **Final State or End State** – The state which the system reaches when a particular process or activity ends is known as a Final State or End State. We use a filled circle within a circle notation to represent the final state in a state machine diagram. A system or a process can have multiple final states.

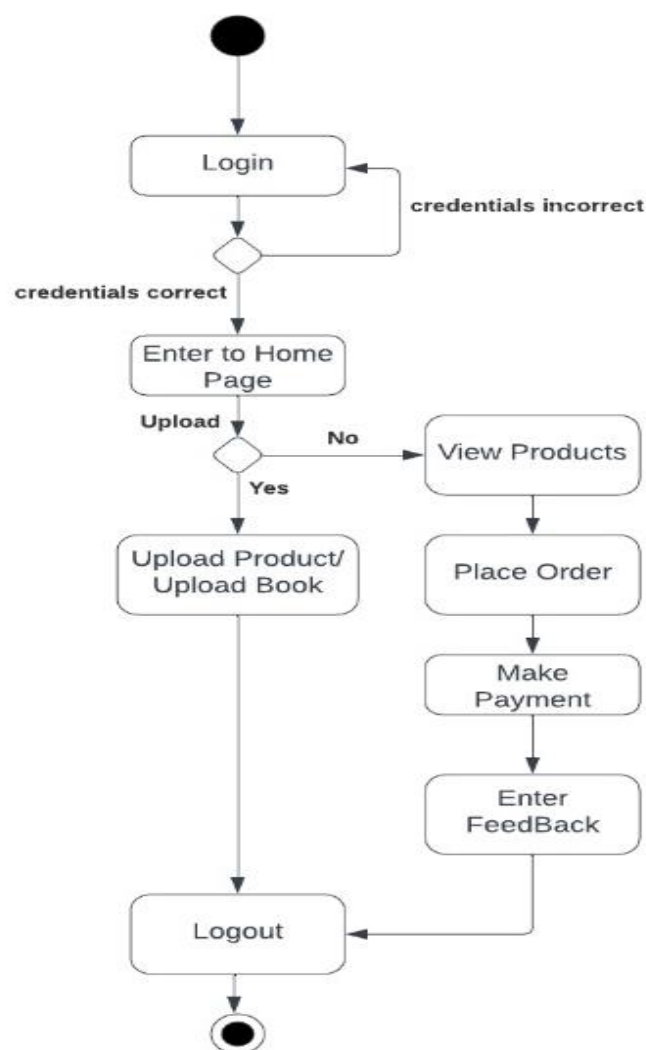


Fig: 4 (Activity Diagram of Campus Centric Second Hand Exchange)

COLLABORATION DIAGRAM:

A collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send receive message. A collaboration diagram is very similar to sequence diagram. Collaboration diagram shows the objects and their association with other objects. Sequence diagrams and collaboration diagrams shows same information but sequence diagram focus on the temporal aspect and collaboration diagram focus on communication between the objects of system.

Notations of a Collaboration Diagram

1. Objects:

The representation of an object is done by an object symbol with its name and class underlined, separated by a colon.

In the collaboration diagram, objects are utilized in the following ways:

- The object is represented by specifying their name and class.
- It is not mandatory for every class to appear.
- A class may constitute more than one object.
- In the collaboration diagram, firstly, the object is created, and then its class is specified.
- To differentiate one object from another object, it is necessary to name them.

2. Actors:

In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.

3. Links:

The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.

4. Messages:

It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labelled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

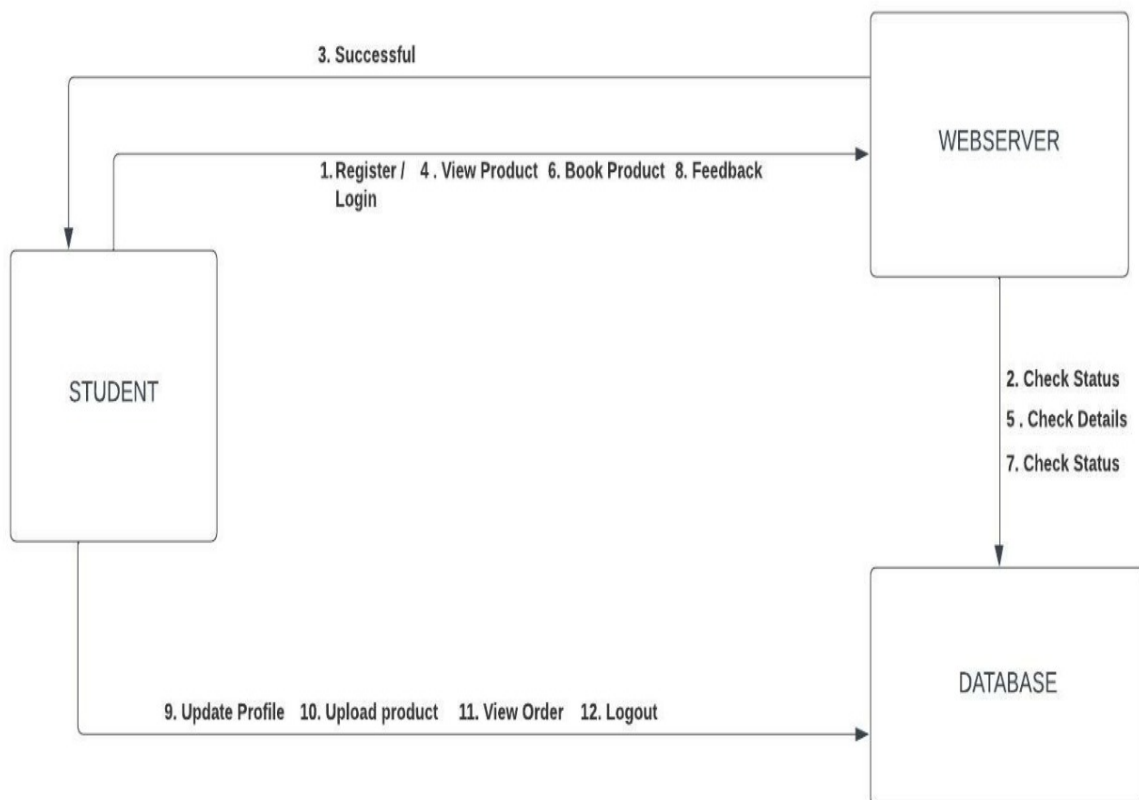


Fig:5 (Collaboration Diagram of Campus Centric Second Hand Exchange)

4.3 ER Diagram

- ER model allows you to draw Database Design
- It is an easy-to-use graphical tool for modeling data
- Widely used in Database Design
- It is a GUI representation of the logical structure of a Database
- It helps you to identifies the entities which exist in a system and the relationshipsbetween those entities

ER Diagrams Symbols & Notations

These are mainly containing three basic symbols which are rectangle, oval and diamond to represent relationships between elements, entities and attributes. There are some sub-elements which are based on main elements in ERD Diagram. ER Diagram is a visual representation of data that describes how data is related to each other using different ERD Symbols and Notations.

Following are the main components and its symbols in ER Diagrams:

- **Rectangles:** This Entity Relationship Diagram symbol represents entity types
- **Ellipses:** Symbol represent attributes
- **Diamonds:** This symbol represents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined.
- **Double Ellipses:** Represent multi-valued attributes

This model is based on three basic concepts:

1. Entities
2. Attributes
3. Relationships

- **Entities:** A real-world thing either living or non-living that is easily recognizable and nonrecognizable. It is anything in the enterprise that is to be represented in our database. It may be a physical thing or simply a fact about the enterprise or an event that happens in the real world. An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.
- **Attributes:** It is a single-valued property of either an entity-type or a relationship-type. For example, a lecture might have attributes: time, date, duration, place, etc. An attribute in ER Diagram examples, is represented by an Ellipse
- **Relationship:** Relationship is nothing but an association among two or more entities. E.g., Tom works in the Chemistry department. Entities take part in relationships. We can often identify relationships with verbs or verb phrases.

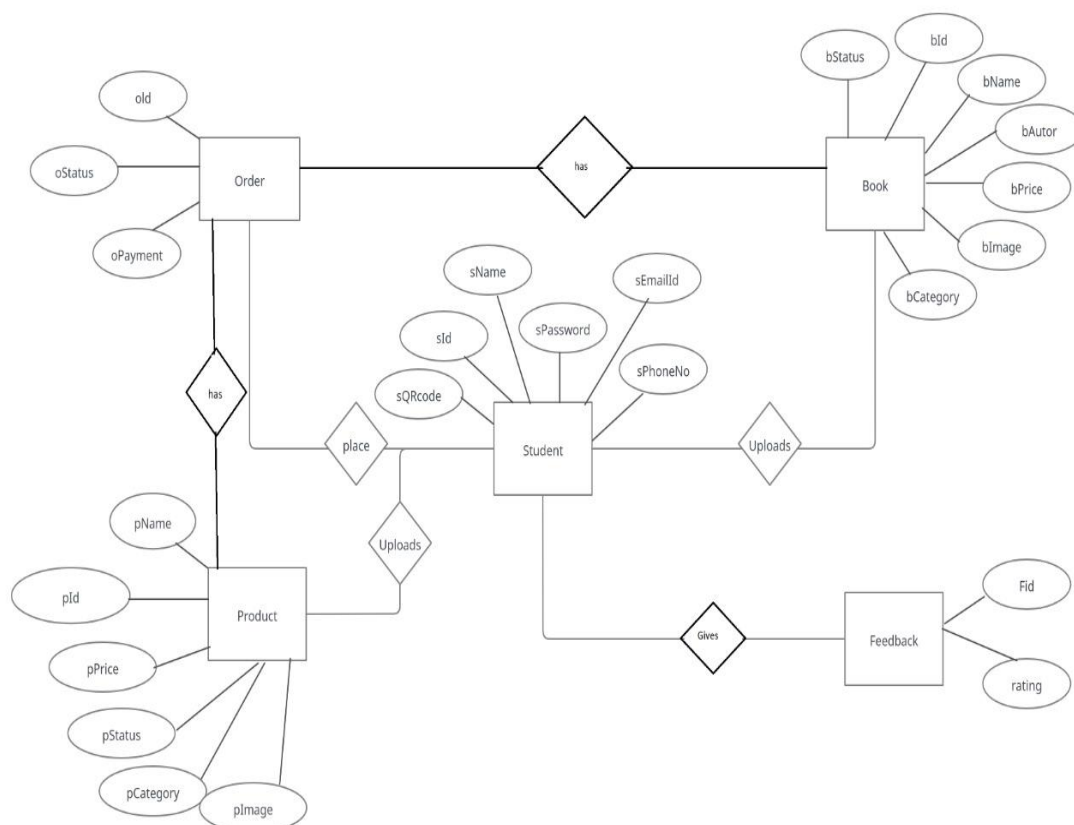


Fig: 6 (ER Diagram of Campus Centric Second Hand Exchange)

CHAPTER - 5

SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

Nowadays, according to our university conditions, the students usually generate a great deal of spare items such as books, clothes or comedies. Especially for graduating students who are leaving soon. Their quilt, 2 pillow, mattress, washbasin, kettles and so on are inconvenient to move back home. Just throwing them away is not only a big waste of money but also wastes resources.

Freshmen, the exact people who need these items, could buy these second-hand items. Therefore, an appropriate way to deal with these things is needed. Considering that, this project designs a website to offer service in order to make the trade safe and convenient. The service could also minimize the cost to our environment.

Existing websites do have fundamental customers. There are still some limitations. Since these platforms are open to the whole citizens, we cannot easily find what we need for school and the transportation is not convenient. We are trying to not move heavy items back to home. It makes no difference if they are sold locally or in a remote place? In this case, a trade website which just focuses on a single university is necessary.

Firstly, in the project the research on the environment and conditions of second-hand trading nowadays, was conducted. The positive attitude about the application was obtained after data collection of students thoughts. Secondly, the ideas from the students helped design the interface and functions. The goal of the application would be achieved to help the users deal with daily problems and satisfy them as most as possible.

5.2 PROJECT MODULES

The system consists of three major modules with their sub-modules as follows.

Admin:

- Maintain Database like add, delete, update.
- Monitors all the work.

Student

- Register: Student can register and obtain credentials.
- Login: Student can login using credentials.
- Forget password: Send an email with OTP and reset password.
- Search for Products.
- Upload Products with image and details
- View and can Update Profile.
- Place and view orders
- View the uploaded products
- Provide rating and feedback

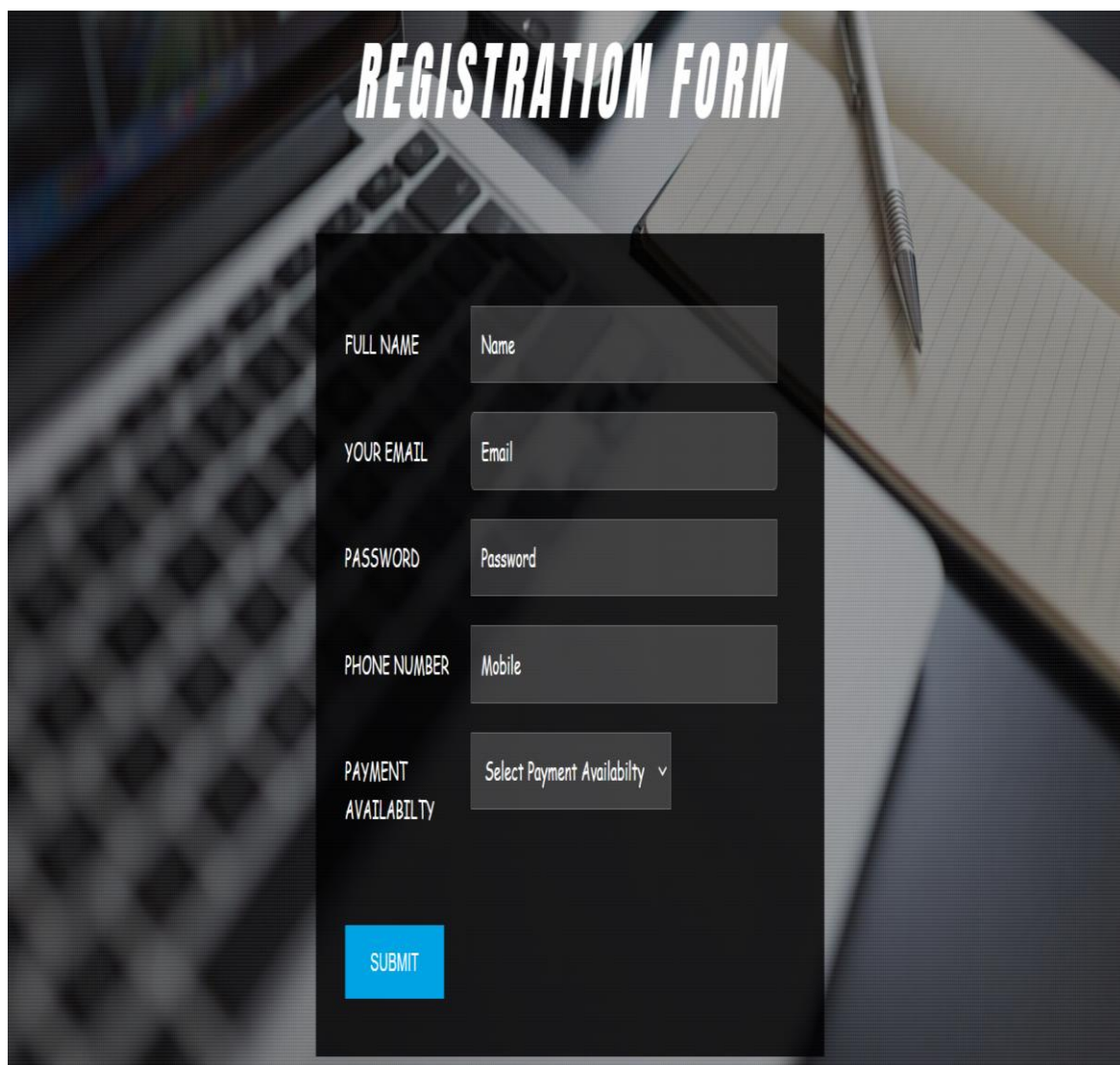
5.3 SCREENS:

HOME:



Home Page

REGISTER:



REGISTRATION FORM

FULL NAME

YOUR EMAIL

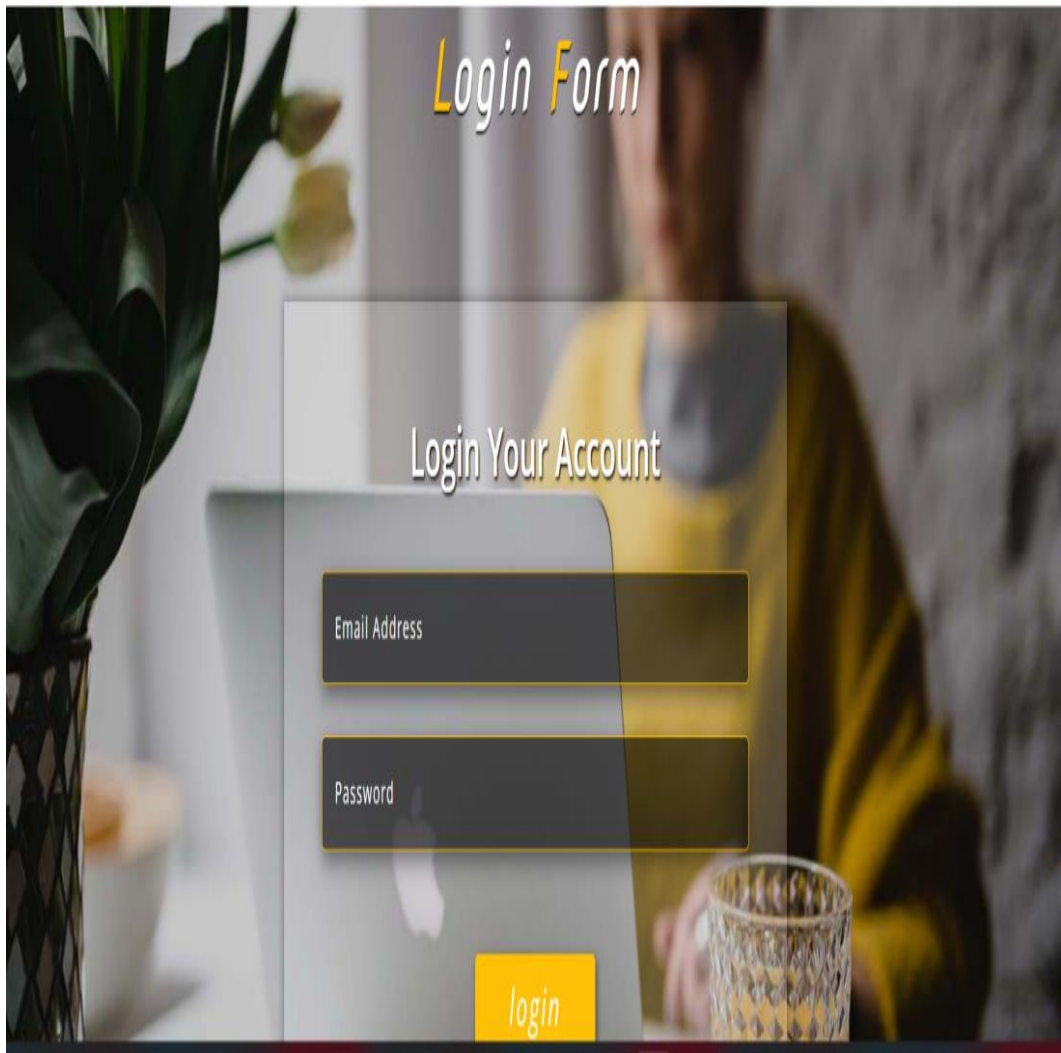
PASSWORD

PHONE NUMBER

PAYMENT AVAILABILITY

Registration Form

LOGIN:



Login Page

DASHBOARD:



Books

If you don't like to read, you haven't found the right book..

Try now



Daily Use

Your day in College life isn't complete without me.!

Try now



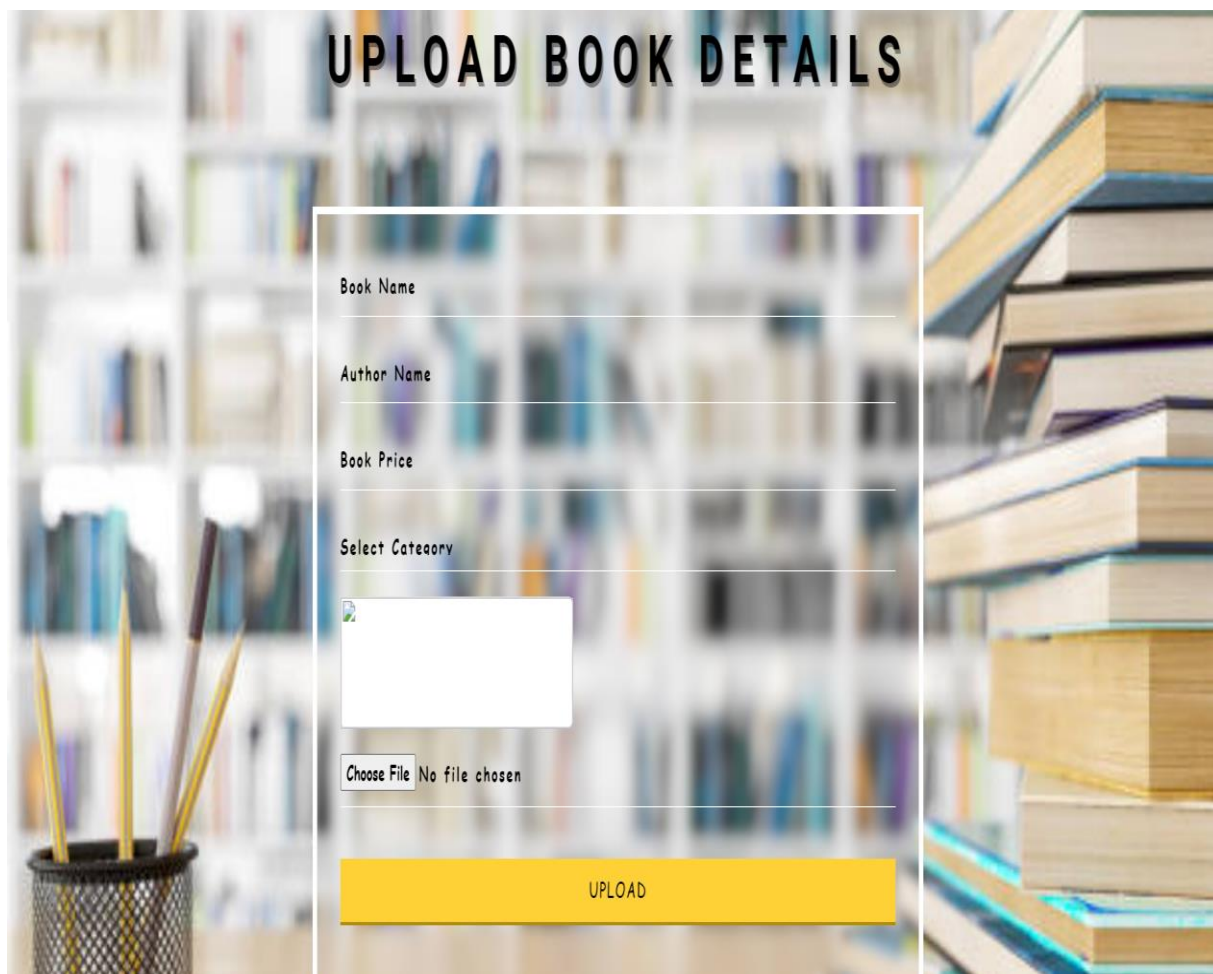
Electronic Device

Dreams about the future are always filled with gadgets.

Try now

Dashboard

UPLOAD BOOK:




UPLOAD BOOK DETAILS

Book Name

Author Name

Book Price

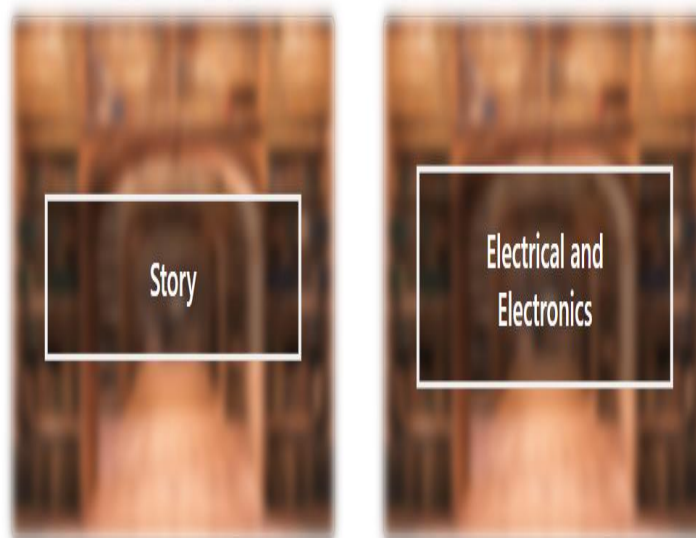
Select Category



No file chosen

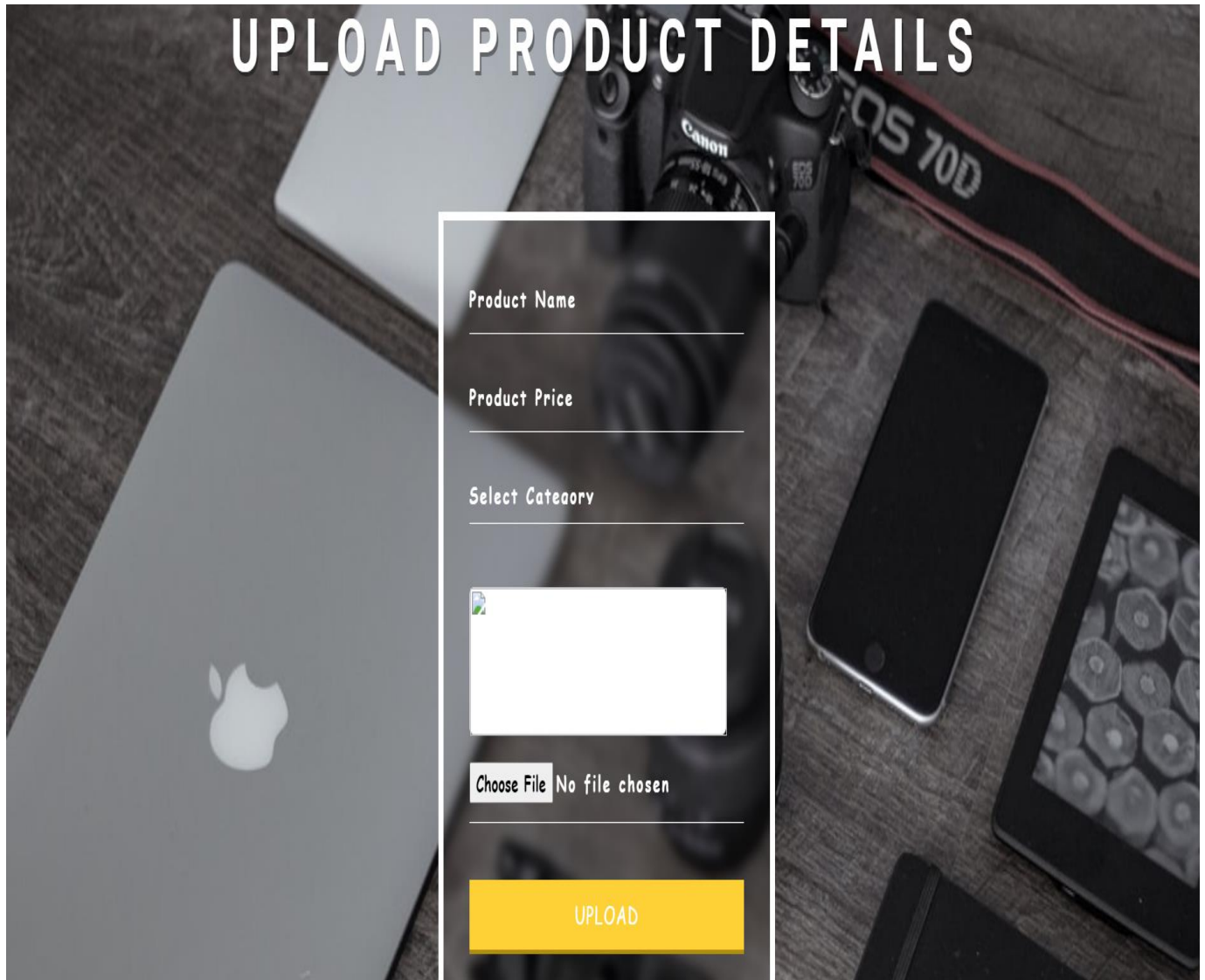
Book Upload

CATEGORY WISE BOOKS:



Category Division

PRODUCT UPLOAD:


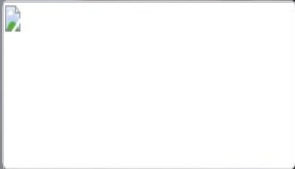
The image shows a semi-transparent white form overlay on a background of various electronic devices including a Canon EOS 70D camera, an Apple laptop, a smartphone, and a tablet. The form is titled 'UPLOAD PRODUCT DETAILS' in large, bold, white capital letters at the top. Below the title, there are four input fields: 'Product Name', 'Product Price', 'Select Category', and a file upload section. The file upload section includes a small icon of a document, a large white rectangular area for the file preview, and a button labeled 'Choose File' next to the text 'No file chosen'. At the bottom of the form is a prominent yellow button labeled 'UPLOAD' in white capital letters.

UPLOAD PRODUCT DETAILS

Product Name

Product Price

Select Category

Choose File No file chosen

UPLOAD

Product Upload

SEARCH PAGE:

Category

▼

Enter Keywords?

Q




What are you looking for?

Search

ORDER DETAILS:

DETAILS



Book Name : Electrical Machines - 3

Author Name : U.A.Bakshi

Category Name: Electrical and Electronics




Book Price : ₹800.00

Seller Name

Details


PAYMENT PAGE:

Payment



Choose Payment Method


Pay by Scan



Seller Mobile Number is : 7036792449

Order Summary

1 item



\$ 800
Electrical Machines - 3
Qty:1

Subtotal	\$ 800
Delivery	Free
Total to pay	\$ 800

Place order

Complimentary Shipping & Returns

Payment Page

FEEDBACK:

Vyshnavi★★★★★
Good website to upload items and this website provides affordable products to purchase.

Vyshnavi★★★★★
good

Rate Us

Out of 5 Star

Rating

Rating

Provide feedback

Submit Feedback

Feedback

PROFILE:



The image shows a 'Profile Update' form overlaid on a background of a single dark feather against a light, textured surface. The form contains three input fields for 'First Name', 'Email Id', and 'Mobile', each with a corresponding label. Below these fields is a grey 'Update' button. The text 'Profile Update' is written in a large, black, cursive font at the top of the form area.

Profile Update

First Name:

Email Id :

Mobile :

Profile

FORGOT PASSWORD:



Forgot Password

CHAPTER - 6
SYSTEM TESTING

6.1. INTRODUCTION:

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested.

Software Testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for III planned through testing.

Testing objectives

These are several rules that can save as testing objectives:

- Testing is a process of executing program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.

Test Levels

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or darkness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product.

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

6.2. TESTING METHODS

6.2.1 Unit Testing: Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application.

6.2.2 Integration Testing: Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

6.2.3 Functional Testing: Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

6.2.4 System Testing: System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test.

6.2.5 White Box Test: White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.2.6 Black Box Test: Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

6.2.7 Unit Testing: Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

6.2.8 Integration Testing: Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

6.2.9 Acceptance Testing: User Acceptance Testing is a critical phase of any project and requires significant participation by the end user.

6.3. TEST CASES:

Test Cases for Register in Student:

S. No	Test Cases	Result
1	Email Id: soumyaparuchuri2000@gmail.com Name: Soumya Password: password Mobile: 912345678 Payment: Offline	Failed Invalid User Entered mobile number is of less than 10 digits
2	Email Id: gchyuigyhbhu Name: Soumya Password: password Mobile: 912345678 Payment: Offline	Failed Invalid User Entered mail is not valid
3	Email Id: soumyaparuchuri2000@gmail.com Name: Soumya Password: password Mobile: 9123456780 Payment: Offline	Passed Valid User

Table: 1 (Register Student Test Cases)

Test Cases for Log-in:

S. No	Test Cases	Result
1	Email Id: soumyaparuchuri2000@gmail.com Password: password	Passed Valid User
2	Email Id: gchyuigyhjbhuj Password: password	Failed Invalid User Entered credentials are not registered.

Table: 2 (Login Student Test Cases)

Test cases for Upload book:

S. No	Test Cases	Result
1	Name: Wings of fire Author: APJ Abdul Kalam Price:500 Category: Story Image: wingsoffire.jpg	Book Uploaded Successfully
2	Name Programming Author: APJ Rhema Thereja Price:600 Category: Other Category Name: Computers Image: wingsoffire.jpg	Book Uploaded Successfully

Table: 3 (Upload Book Test Cases)

Test cases for Upload Product:

Sl. No	Test Cases	Result
1.	Name: Drafter Price:500 Category: Electronic Device Image: calci.jpg	Product Uploaded Successfully
2.	Name: Chair Price: 600 Category: Daily use Image: chair.jpg	Product Uploaded Successfully

Table: 4 (Upload Product Test Cases)

Test cases for Search:

Sl. No	Test Cases	Result
1.	Category: Books Keyword: wing	Image: wingsoffire.jpg Name: Books Of Fire Author: APJ Abdul Kalam
2.	Category: Electronic and daily use Keyword: calc	Image: calci.jpg Name: calculator
3.	Category: Books Keyword: electrical	No results found

Table: 5 (Search Test Cases)

Test Cases for Feedback:

Sl. No	Test Cases	Result
1.	Rating:4 Description: Good website to upload items and this website provides affordable products to purchase.	Feedback uploaded successfully
2.	Rating:5 Description: Good	Feedback uploaded successfully

Table: 6 (Feedback Test Cases)

CHAPTER – 7

CONCLUSION

The development of the Internet greatly changes people's life, and brings a lot of convenience. This system is to develop a second-hand market application system for students, which uses the Internet characteristics. It provides a quick and convenient online trading platform for current students in school, which can let the students spare yard to realize its value, let the students in need access to cheap and satisfied second-hand items. Give full play to its performance, save the costs, and make reasonable use of network resources.

School students now do not have a very good handling for second-hand resources, which not only wastes the resources, also can't save economic cost. Based on these characteristics, this website designs campus second-hand market application system, systematically expounds the implement and main functions of campus second-hand market application system.

The website designed and implemented commodity information browsing, commodity classification inquiries, online communication and background management system, and other functions. It provides a quick and convenient online trading platform mainly for student's second-hand goods, let the students spare yard realize its value, and let the students in need access to cheap and satisfied second-hand items. It is both environmental and economic, and has high practical value.

List of objectives that are implemented are:

- Student can upload the product details to sell.
- Student can search the product by its keyword.
- Student can place order for a particular product.
- Students can share their feedback to the website.

The aim of developing this project is to provide a platform for students to effortless trading of old products. It not only saves students money but also helps to safeguard the environment.

CHAPTER - 8
BIBLIOGRAPHY

1. "Why Do College Books Cost So Much?", thoughtco, 2017. [Online]. Available: <https://www.thoughtco.com/why-are-textbooks-so-expensive-78849>. [Accessed: 14-Mar- 2018].
2. "Second Hand Trading Website for Student", Google, 2018. [Online]. Available: https://www.google.com.hk/search?safe=strict&client=aff-cs-360se&hs=OWF&ei=BTyiWtjRCMuAjwOEirzAAw&q=second+hand+trading+website+for+students&oq=second+hand+trading+website+for+stu&gs_l=psy-ab.3.0.33i160k1l3.292603.294576.0.296320.8.8.0.0.0.209.792.6j1j1.8.0....0...1.1.64.psy-ab..0.3.283....0.MmE3Tm9ZiU4. [Accessed: 19- Mar- 2018].
3. "Sustainable School Shop - Second hand Textbooks, Uniforms etc.", Sustainableschoolshop.com.au, 2018. [Online]. Available: <https://www.sustainableschoolshop.com.au/>. [Accessed: 17- Feb- 2018].
4. "Angular Documentation for project development" [Online]. Available: <https://www.typescriptlang.org/docs/handbook/angular.html>
5. "Eclipse documentation for backend" [Online]. Available: <https://www.eclipse.org/m2e/documentation/m2e-documentation.html>
6. "Hibernate documentation for database access" [Online]. Available: https://www.tutorialspoint.com/hibernate/hibernate_quick_guide.htm
7. "Java Textbook" [Online]. Available: <https://gfgc.kar.nic.in/sirmv-science/GenericDocHandler/138-a2973dc6-c024-4d81-be6d-5c3344f232ce.pdf>

8. "HTML Textbook" [Online]. Available: <https://www.dcpehvpvm.org/E-Content/BCA/BCA-II/Web%20Technology/the-complete-reference-html-css-fifth-edition.pdf>
9. "Software Engineering Textbook" [Online]. Available: <https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf>
10. "Unified Modeling Language Textbook" [Online]. Available: https://personal.utdallas.edu/~chung/Fujitsu/UML_2.0/Rumbaugh--UML_2.0_Reference_CD.pdf

CHAPTER – 9

APPENDIX

9.1 Introduction to Java:

Java is an object-oriented programming language with its runtime environment. It is a combination of features of C and C++ with some essential additional concepts. Java is well suited for both standalone and web application development and is designed to provide solutions to most of the problems faced by users of the internet era.

- Java is an object-oriented programming language developed by Sun Microsystems, and it was released in 1995.
- James Gosling initially developed Java in Sun Microsystems (which was later merged with Oracle Corporation).
- Java is a set of features of C and C++. It has obtained its format from C, and OOP features from C++.
- Java programs are platform independent which means they can be run on any operating system with any processor as long as the Java interpreter is available on that system.
- Java code that runs on one platform does not need to be recompiled to run on another platform; it's called write once, run anywhere (WORA).
- Java Virtual Machine (JVM) executes Java code, but it has been written in platform specific languages such as C/C++/ASM, etc. JVM is not written in Java and hence cannot be platform independent, and Java interpreter is a part of JVM.

Types of Java Applications

- **Web Application** - Java is used to create server-side web applications. Currently, Servlet, JSP, Struts, JSF, etc. technologies are used.

- **Standalone Application** - It is also known as the desktop application or window-based application. An application that we need to install on every machine or server such as media player, antivirus, etc. AWT and Swing are used in java for creating standalone applications.
- **Enterprise Application** - An application that is distributed in nature, such as banking applications, etc. It has the advantage of high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.
- **Mobile Application** - Java is used to create application software for mobile devices. Currently, Java ME is used for building applications for small devices, and also Java is a programming language for Google Android application development.

Features of Java

- **Object-Oriented** - Java supports the features of object-oriented programming. Its object model is simple and easy to expand.
- **Platform independent** - C and C++ are platform dependency languages hence the application programs written in one Operating system cannot run in any other Operating system, but in platform independence language like Java application programs written in one Operating system can able to run on any Operating system.
- **Simple** - Java has included many features of C / C ++, which makes it easy to understand.
- **Secure** - Java provides a wide range of protection from viruses and malicious programs. It ensures that there will be no damage and no security will be broken.

- **Portable** - Java provides us with the concept of portability. Running the same program with Java on different platforms is possible.
- **Robust** - During the development of the program, it helps us to find possible mistakes as soon as possible.
- **Multi-threaded** - The multithreading programming feature in Java allows you to write a program that performs several different tasks simultaneously

9.2 Introduction to HTML:

HTML stands for Hyper Text Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages.

This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g., HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

HTML is a markup language used by the browser to manipulate text, images, and other content, in order to display it in the required format. HTML was created by Tim Berners-Lee in 1991. The first-ever version of HTML was HTML 1.0, but the first standard version was HTML 2.0, published in 1999.

Elements and Tags: HTML uses predefined tags and elements which tell the browser how to properly display the content. Remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of the page.

HTML page structure: The basic structure of an HTML page is laid out below. It contains the essential building-block elements (i.e., doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.

<DOCTYPE! html>: This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case sensitive.

<html>: This is called the HTML root element. All other elements are contained within it.

<head>: The head tag contains the “behind the scenes” elements for a webpage. Elements within the head aren’t visible on the front-end of a webpage.

HTML elements used inside the <head> element include:

- <style>
- <title>
- <script>
- <noscript>
- <meta>
- <link>
- <base>

<body>: the body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front-end.

An HTML document can be created using any text editor. Save the text file using .html or .htm. Once saved as an HTML document, the file can be opened as a webpage in the browser.

9.3 Introduction to CSS:

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

- **CSS saves time** – You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

- **Superior styles to HTML** – CSS have a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it's a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

9.4 Introduction to JavaScript and TypeScript:

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive (e.g., having complex animations, clickable buttons, popup menus, etc.). There are also more advanced server-side versions of JavaScript such as Node.js, which allow you to add more functionality to a website than downloading files (such as real-time collaboration between multiple computers). Inside a host environment (for example, a web browser), JavaScript can be connected to the objects of its environment to provide programmatic control over them.

JavaScript contains a standard library of objects, such as Array, Date, and Math, and a core set of language elements such as operators, control structures, and statements. Core JavaScript can be extended for a variety of purposes by supplementing it with additional objects; for example:

- *Client-side JavaScript* extends the core language by supplying objects to control a browser and its Document Object Model (DOM). For example, client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation.
- *Server-side JavaScript* extends the core language by supplying objects relevant to running JavaScript on a server. For example, server-side extensions allow an application to communicate with a database, provide continuity of information from one invocation to another of the application, or perform file manipulations on a server.

This means that in the browser, JavaScript can change the way the webpage (DOM) looks. And, likewise, Node.js JavaScript on the server can respond to custom requests from code written in the browser.

TypeScript is an open-source pure object-oriented programming language. It is a strongly typed superset of JavaScript which compiles to plain JavaScript. It contains all elements of the JavaScript. It is a language designed for large-scale JavaScript application development, which can be executed on any browser, any Host, and any Operating System. The TypeScript is a language as well as a set of tools. TypeScript is the ES6 version of JavaScript with some additional features.

TypeScript cannot run directly on the browser. It needs a compiler to compile the file and generate it in JavaScript file, which can run directly on the browser. The TypeScript source file is in ".ts" extension. We can use any valsssid ".js" file by renaming it to ".ts" file. TypeScript uses TSC (TypeScript Compiler) compiler, which convert Typescript code (.ts file) to JavaScript (.js file).

9.5 Introduction to MySQL:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

- MySQL is released under an open-source license. So, you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.

What is a Database?

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as **Foreign Keys**.

A **Relational Data Base Management System (RDBMS)** is a software that –

- Enables you to implement a database with tables, columns and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically.
- Interprets an SQL query and combines information from various tables.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

9.6 Introduction to Angular, Bootstrap, Rest API and Hibernate

Angular:

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications. The architecture of an Angular application relies on certain fundamental concepts.

The basic building blocks of the Angular framework are Angular components that are organized into Ng Modules. Ng Modules collect related code into functional sets; an Angular application is defined by a set of Ng Modules.

An application always has at least a root module that enables bootstrapping, and typically has many more feature modules.

- Components define views, which are sets of screen elements that Angular can choose among and modify according to your program logic and data.
- Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient.

Modules, components and services are classes that use decorators. These decorators mark their type and provide metadata that tells Angular how to use them.

- The metadata for a component class associates it with a template that defines a view. A template combines ordinary HTML with Angular directives and binding markup that allow Angular to modify the HTML before rendering it for display.
- The metadata for a service class provides the information Angular needs to make it available to components through dependency injection (DI). An application's components typically define many views, arranged hierarchically. Angular provides the Router service to help you define navigation paths among views. The router provides sophisticated in-browser navigational capabilities.

Bootstrap:

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

The Bootstrap structure is pretty simple and self-explanatory. It includes precompiled files that enable quick usage in any web project. Besides compiled and minified CSS and JS files, it also includes fonts from Glyphicons, and the optional starting Bootstrap theme.

Bootstrap comes bundled with basic HTML and CSS design templates that include many common UI components. These include Typography, Tables, Forms, Buttons, Glyphicons, Dropdowns, Buttons and Input Groups, Navigation, Pagination, Labels and Badges, Alerts, Progress Bars, Modals, Tabs, Accordions, Carousels, and many others.

Why Bootstrap?

- Faster and Easier Web Development.
- It creates Platform-independent web pages.
- It creates Responsive Web-pages.
- It designed to be responsive to mobile devices too.

Rest API:

REST stands for Representational State Transfer. It is an architectural Style for designing networked applications (i.e., apps that use some form a network to communicate). It is the most popular style for building web APIs. REST determines specifications of the API through a set of rules that are followed when a REST API is created.

REST treats any data (e.g., image, video, text, etc) as a resource that the client can fetch/edit/delete. REST mandates that a client should be able to perform the appropriate operation by accessing a specific URL and sending a request. The server then sends an appropriate response.

Think of it as a contract between the programs: the requester (aka the client) and the responder (aka the server). If the requester sends X to the responder, the responder will give the requester Y. X and Y are specified in contract between the two parties and explained in the documentation for the API.

REST is stateless, which implies that each request from the client must have all the necessary information for the server to understand it. For example, the client cannot assume that the server remembers what they had asked for earlier.

Advantages of RESTful Web Services

- **Fast:** RESTful Web Services are fast because there is no strict specification like SOAP. It consumes less bandwidth and resource.

- **Language and Platform independent:** RESTful web services can be written in any programming language and executed in any platform.
- **Can use SOAP:** RESTful web services can use SOAP web services as the implementation.
- **Permits different data format:** RESTful web service permits different data format such as Plain Text, HTML, XML and JSON.

Hibernate:

Hibernate is a Java framework that simplifies the development of Java application to interact with the database. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.

Hibernate Query Language (HQL) is an object-oriented query language, similar to SQL, but instead of operating on tables and columns, HQL works with persistent objects and their properties. HQL queries are translated by Hibernate into conventional SQL queries, which in turns perform action on database.

Integration - Hibernate can be used both in standalone Java applications and in Java EE applications using servlets, EJB session beans, and JBI service components. It can also be included as a feature in other programming languages.

For example, Adobe integrated Hibernate into version 9 of ColdFusion (which runs on J2EE app servers) with an abstraction layer of new functions and syntax added into CFML.

Entities and Components - In Hibernate jargon, an entity is a stand-alone object in Hibernate's persistent mechanism which can be manipulated independently of other objects. In contrast, a component is subordinate to an entity and can be manipulated only with respect to that entity.

For example, an Album object may represent an entity; but the Tracks object associated with the Album objects would represent a component of the Album entity, if it is assumed that Tracks can only be saved or retrieved from the database through the Album object. Unlike J2EE Hibernate can switch databases.

Advantages of Hibernate Framework

- **Open Source and Lightweight:** Hibernate framework is open source under the LGPL license and lightweight.
- **Fast Performance:** The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first level cache and second level cache. First level cache is enabled by default.
- **Database Independent Query:** HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries. So, you don't need to write database specific queries. Before Hibernate, if database is changed for the project, we need to change the SQL query as well that leads to the maintenance problem.
- **Automatic Table Creation:** Hibernate framework provides the facility to create the tables of the database automatically. So, there is no need to create tables in the database manually.

- **Simplifies Complex Join:** Fetching data from multiple tables is easy in hibernate framework.
- **Provides Query Statistics and Database Status:** Hibernate supports Query cache and provide statistics about query and database status.