**Advance System Project**

**--------**

**YOUR PROJECT NAME**

**Name:**

**Student ID:**

**Professor:**

**Introduction**

The traditional way of how library works is, a individual will go personally to the library enquire about the availability of books, search and owe the book for certain period of time and return the book on or before due date. With the growth of internet and technology advancement many useful applications/ software are designed for easy and quick access.

Librarysoft is one such application which acts as a platform for checking the availability status of the books online and checking the different category of books available in that library. The main intension of this application is to reduce human effort and time.

**Project features**

This application has registration page for both staff and student who fills all the necessary details. After registration, student can login to the application, can choose a book based on the categories displayed.

Search option would be provided to select choice of books online and can borrow the book for particular period of time. Admin or authoritative staff member can issue the book to student after verifying the student information. Admin has the right to remove the student, update the new books, and update information regarding Library Soft.

User/Student after owing the book need to return the book on or before due date, if student failed to return the book he/she would be charged some extra fee. User/student doesn’t want to continue with the Library Soft Admin can remove the user and he is no longer accessible to the application

**Technologies used:**

**Front End:** HTML, CSS, JSP, JAVA SCRIPT

**Logic:** JAVA

**Controller:** Servlets

**Database platform:** SQL

**Server:** Apache Tomcat

**Frame work:** Springs/Struts

**UML Diagrams:**

UML stands for Unified Modelling Language. Which is used in object-oriented software engineering. It's typically used in software engineering it is a rich language that can be used to model an application structure, behavior and even business processes. There are **14 UML diagram types.**

They can be divided into two main categories are structured diagrams and Behavioral diagrams. The scenarios are organized hierarchically and they capture the system functionality at various abstraction levels, including scenario groups, scenarios, and sub-scenarios. Combining scenarios or sub-scenarios can form complex scenarios. Data are also separately identified, organized, and attached to scenarios. This scenario model can be used to cross check with the UML model. The modeling the logic captured by a single use case or usage scenario, or for modeling the detailed logic of a business rule. Graphical notations used in structural things are the most widely used in UML. The main types of UML Diagrams in this project:

Use case diagram

Sequence diagram

Activity diagram

Class diagrams

Collaboration diagram

State Chart Diagram

Deployment Diagram

Component Diagram

**Use Case Diagrams:**

Actor Notation:

An actor can be defined as some internal or external entity that interacts with the system.



Actor is used in a use case diagram to describe the internal or external entities.

Use case is represented as an eclipse with a name inside it. It may contain additional responsibilities.

Login

Use case is used to capture high-level functionalities of a system.



Student use case diagram



Staff Use case diagram



Admin Use case diagram

**Use Cases (Detail Description)**

**Case 1: USER Registration**

**Primary Actor:**User

**Agenda:** The New user can be able to create the account by in the e commerce register application.

**Scope:** A User can login to the system application upon successful registration.

**Precondition:** All users should have a valid email id and Password to register in the e commerce application.

**Post Conditions:**

1. The new user and already registered can browse and all the categories the webpages.
2. Click on the user login for login.
3. Click on create register to register as new user.
4. Provide appropriate details for a new account for user.

**Failure Conditions:**

1. Entering the invalid details will registration.

**Case 2: User Login**

**Primary Actor: U**ser

**Agenda:** User can be able to login to the application using credentials.

**Scope:** A registered user can access all the allowed menus after successful Login.

**Precondition: U**ser should have successfully registered to the application before login.

**Post Conditions:**

1. A user can navigate to the all category using the navigation menu.
2. Login using appropriate emails and password.
3. View and navigate through the available access menu.
4. Logout Successfully.

**Success Conditions:**

1. User can access all navigation menu options.

2. Login using appropriate email and password.

3. View of all items based on category.

4. By clicking on add to cart button use can add items to cart.

5. By clicking place order button user can we all cart items if user already login else he need to login once.

6. User needs to provide successfully his address during placing a order.

7. Finally user needs to conform the order.

8. Logout successfully.

**Failure Conditions:**

1. Invalid user Email or Password.

**Case 4: Admin Login**

**Primary Actor:** Admin

**Agenda:**  Admin can be able to login to the application with Email-id and Password.

**Scope:**  Admin can access all the allowed menus available in the application without any restrictions.

**Precondition:**Admin should have valid login credentials and URL to access the application menu.

**Success Conditions:**

1. Admin can navigate to the navigation menu in the Admin home.
2. Login by providing appropriate Email-id and Password.
3. All the navigation view through the available menu.
4. Logout Successfully.

**Failure Conditions:**

1. Invalid user Email or Password.

**Case 5: Admin Operations**

**Primary Actor:** Admin

**Agenda: A**dmin can be able to access the application menu such as, Add Category, Add Sub Category, Add Items , Add Sub Category and Forgot Password.

**Scope:** Admin can access and can add items to category and navigation menu available in the application.

**Precondition:**  Admin should have to provide valid login emailid and Password to access the application system.

**Success Conditions:**

1. Admin user can navigate to the navigation menu in the Admin home.
2. Login by providing appropriate Email-id and Password.
3. Add Categories Details.
4. Add Sub Category Details
5. Add items to different Category.
6. Logout Successfully.

**Failure Conditions:**

1. Invalid user Email or Password.

**Entity-Relationship diagrams:**

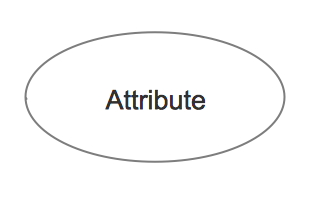
An entity–relationship model (ER model) describes composed of entity types and specifies relationships that can exist between instances of those entity types. The Semantic modeling is modeling data structures, based on the meaning of these data. Different variants of the entity-relationship diagrams are used as a tool for the semantic modeling. ER-model based diagrams have three main components: an entity, a relation and attributes. An entity is a class of similar objects, information about which should be considered in the model.

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It is usually drawn in a graphical form as boxes (entities) that are connected by lines.An ER model is typically implemented as a database.

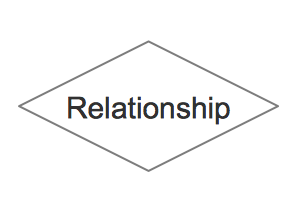
**Entity**: An entity is represented by a rectangle which contains the entity’s name.



**Attribute:** In the Chen notation, each attribute is represented by an oval containing an attribute's name



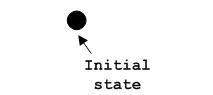
**Relationship:** A relationship where the entity is existence independent of another entity. A strong relationship is represented by a single rhombus.In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database, a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.



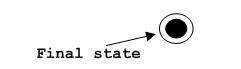
**Activity Diagram:**

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

**Initial node** The filled in circle is the starting point of the diagram.The initial state is defined to show the start of a process. This notation is used in almost all diagrams.The usage of Initial State Notation is to show the starting point of a process.



**Activity final node** The filled circle with a border is the ending point. An activity diagram can have zero or more activity final nodes. This notation is also used in almost all diagrams to describe the end.The usage of Final State Notation is to show the termination point of a process.



**Activity diagram:**

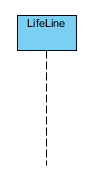


**Sequence Diagram:**

The sequence diagram shows how objects communicate with each other in terms of a sequence of messages. Interaction diagrams, a subset of behavior diagrams, emphasize the flow of control and data among the things in the system being modeled.

**Lifeline:**

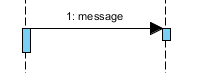
A lifeline represents an individual participant in the Interaction.



**Send message:**

A message defines a particular communication between Lifelines of an Interaction.

Send message is a kind of message that represents the start of execution.



**Return Message:**

A message defines a particular communication between Lifelines of an Interaction.

Return message is a kind of message that represents the pass of information back to the caller of a corresponded former message.

https://www.visual-paradigm.com/VPGallery/img/diagrams/Sequence/ReturnMessage.png

**Self Message:**

A message defines a particular communication between Lifelines of an Interaction.

Self message is a kind of message that represents the invocation of message of the same lifeline.





Sequence Diagram

**Class Diagram**

It describes the structure of a system by showing the system's classes, their attributes, and the relationships among the classes.

****

Class Diagram

**Collaboration Diagram**

It is a specific type of interaction diagram, where the focus is on timing constraints.



Collaboration Diagram

**State Chart Diagram**

It is standardized notation to describe many systems, from computer programs to business processes.

****

Student - State Chart Diagram



Staff – State Chart Diagram



Admin – State Chart Diagram

**Deployment Diagram**

It serves to model the hardware used in system implementations, and the execution environments and artifacts deployed on the hardware.



Deployment Diagram

**Component Diagram**

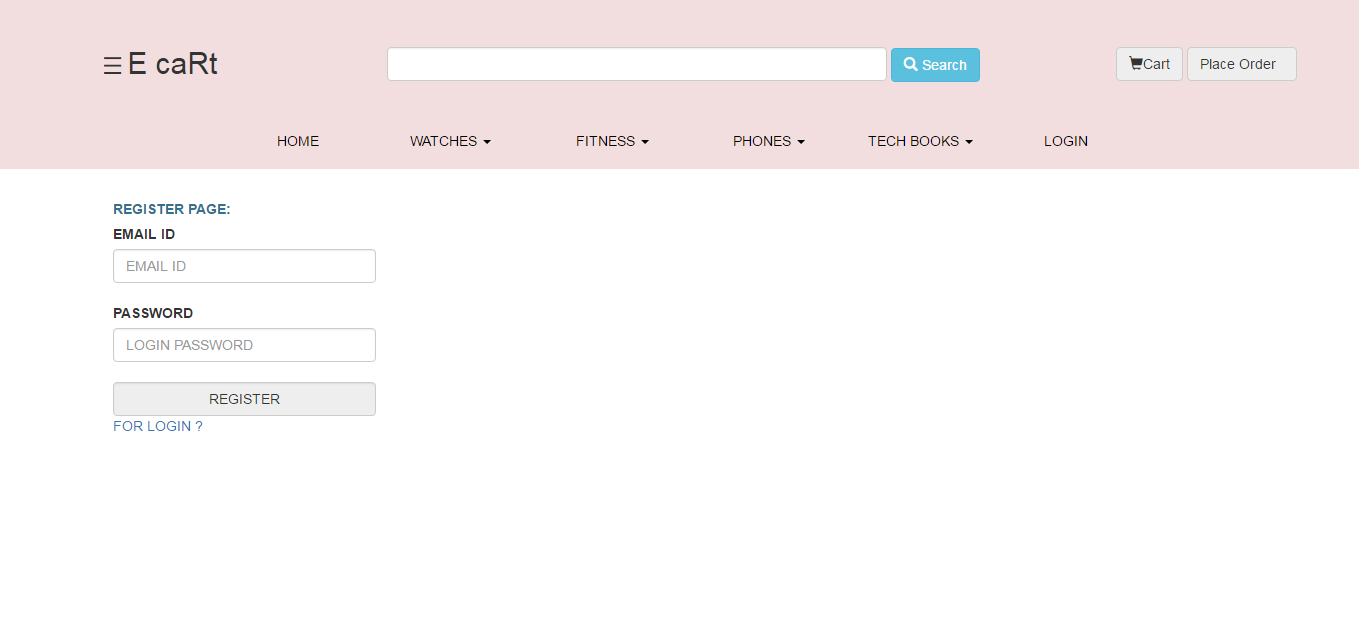
It depicts how a software system is split up into components and shows the dependencies among these components.



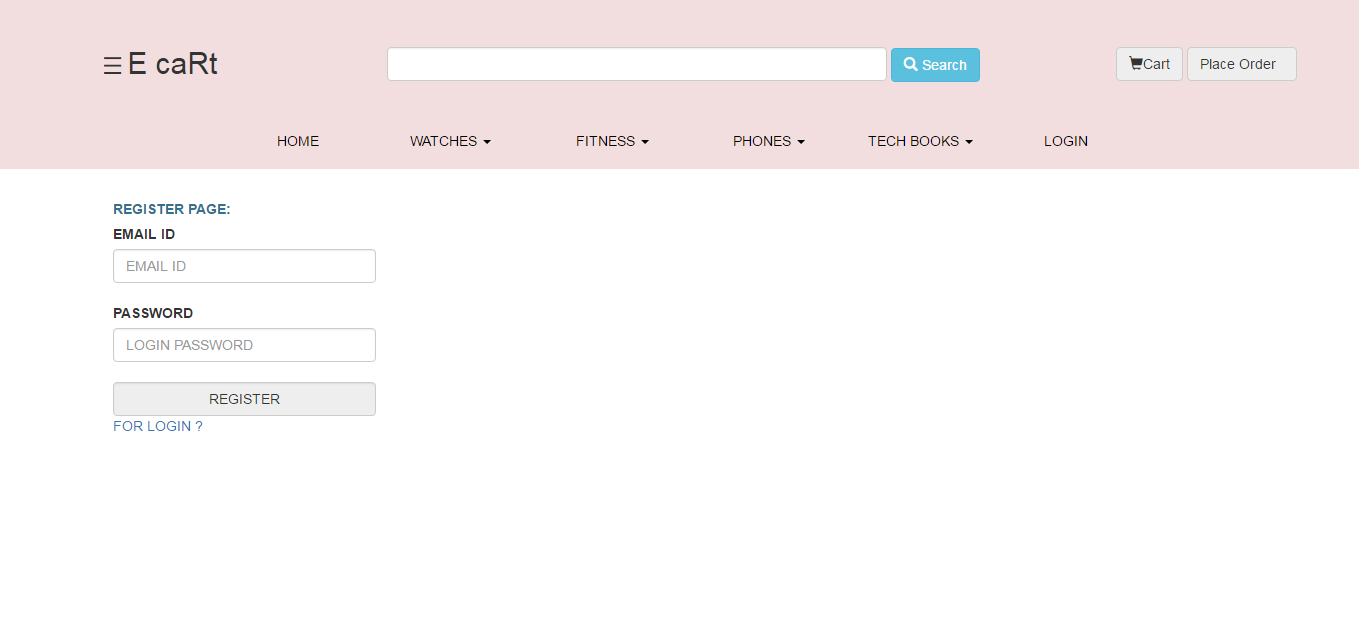
Component Diagram

**Website Screenshots:**

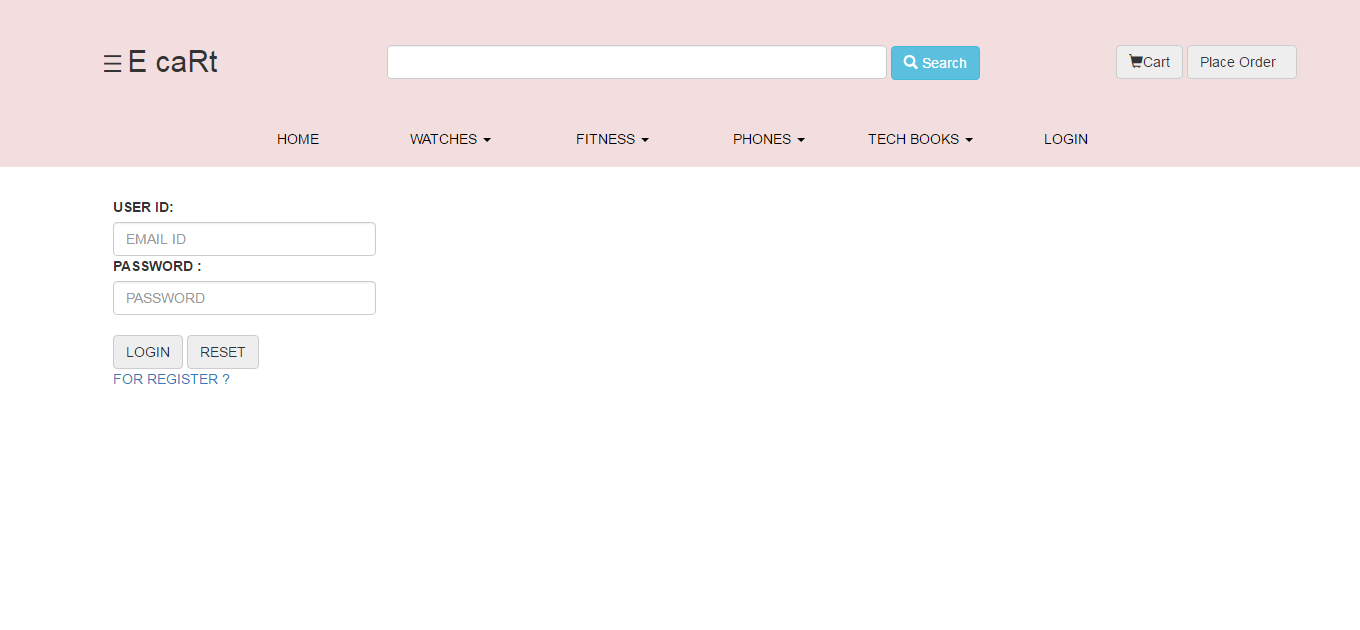
Home page:



Register Page:



Login page:



Items view:

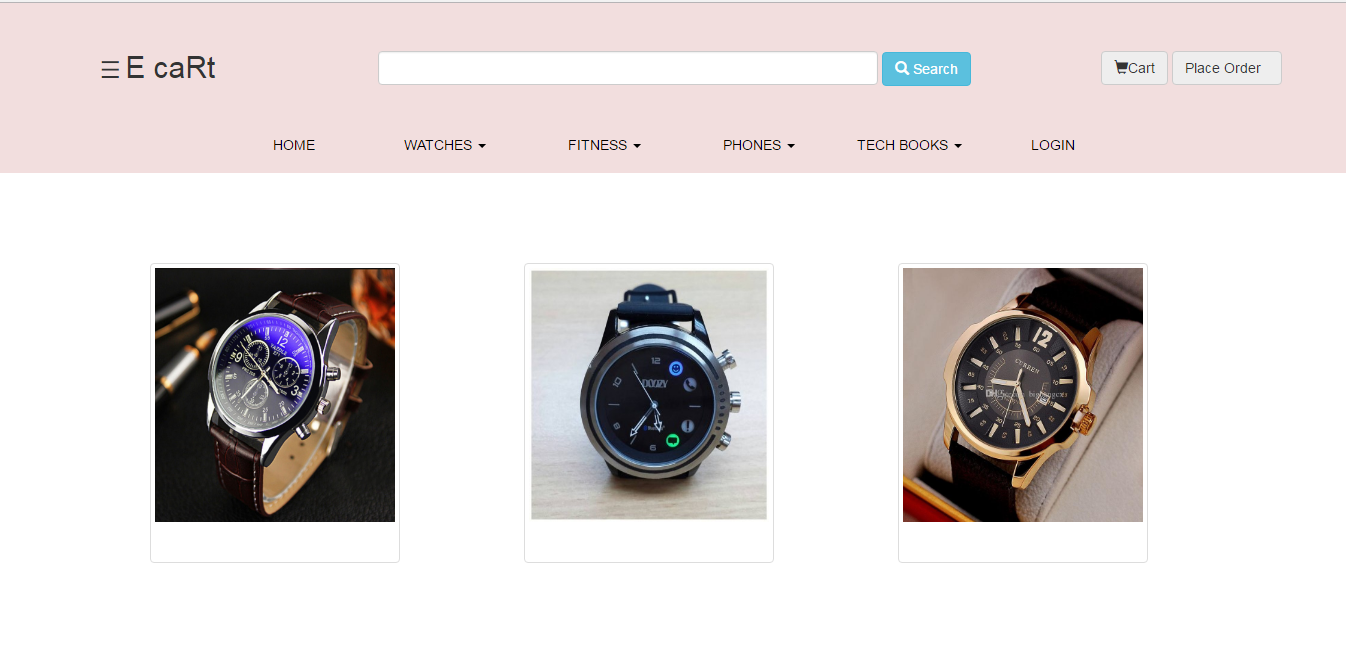
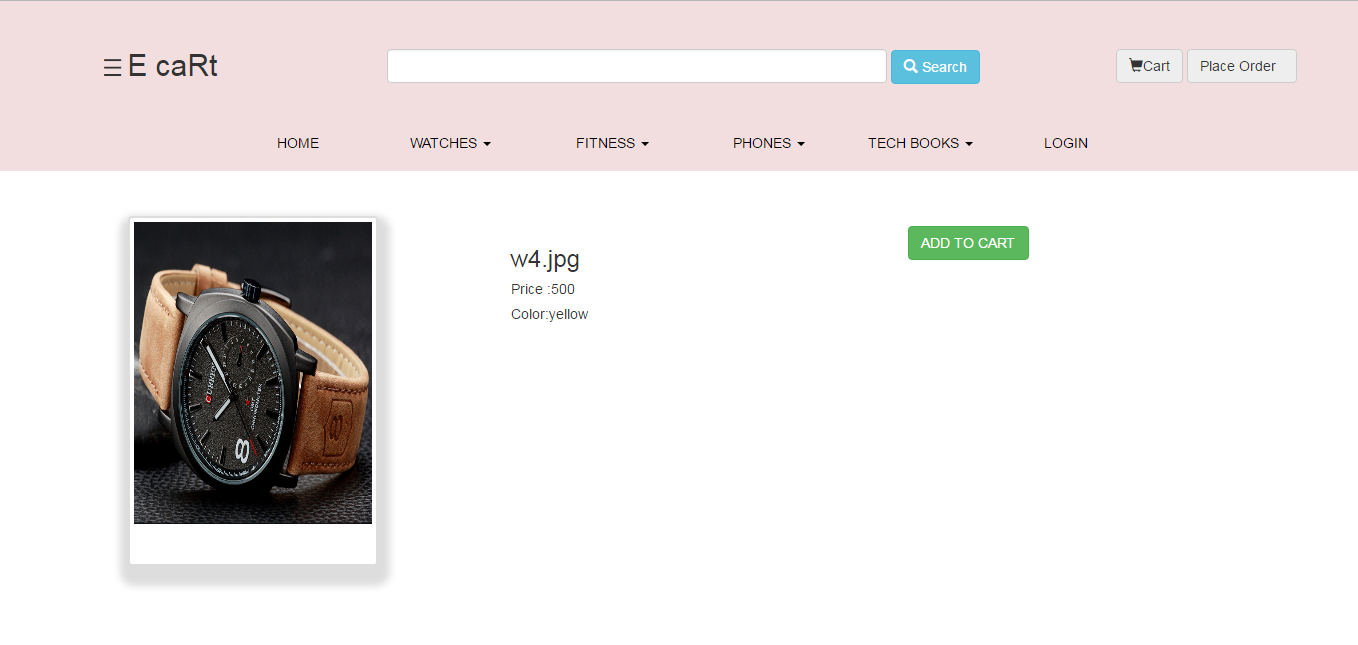


Image details views



Address form:

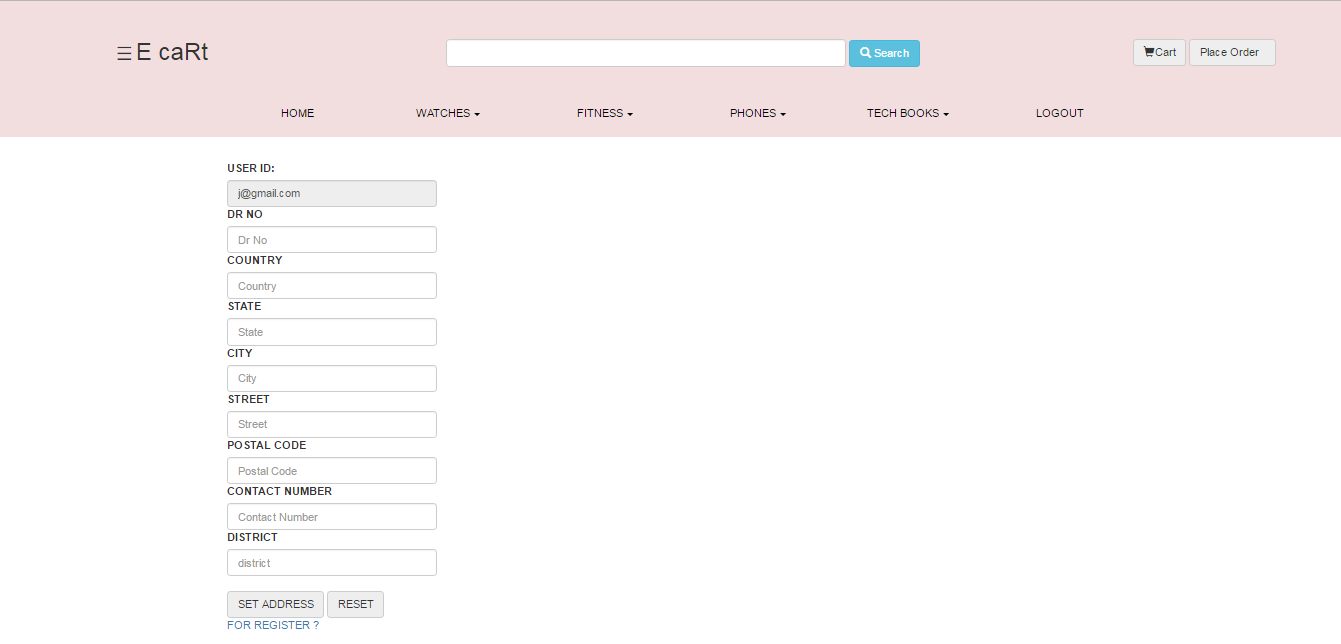
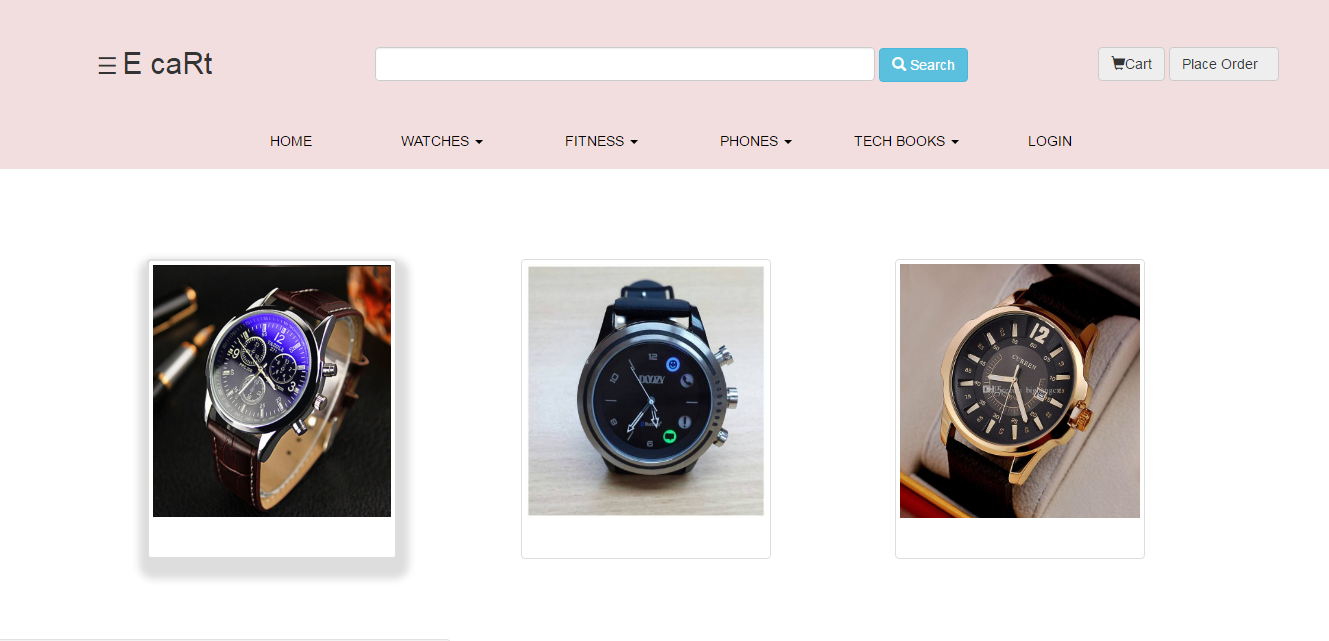
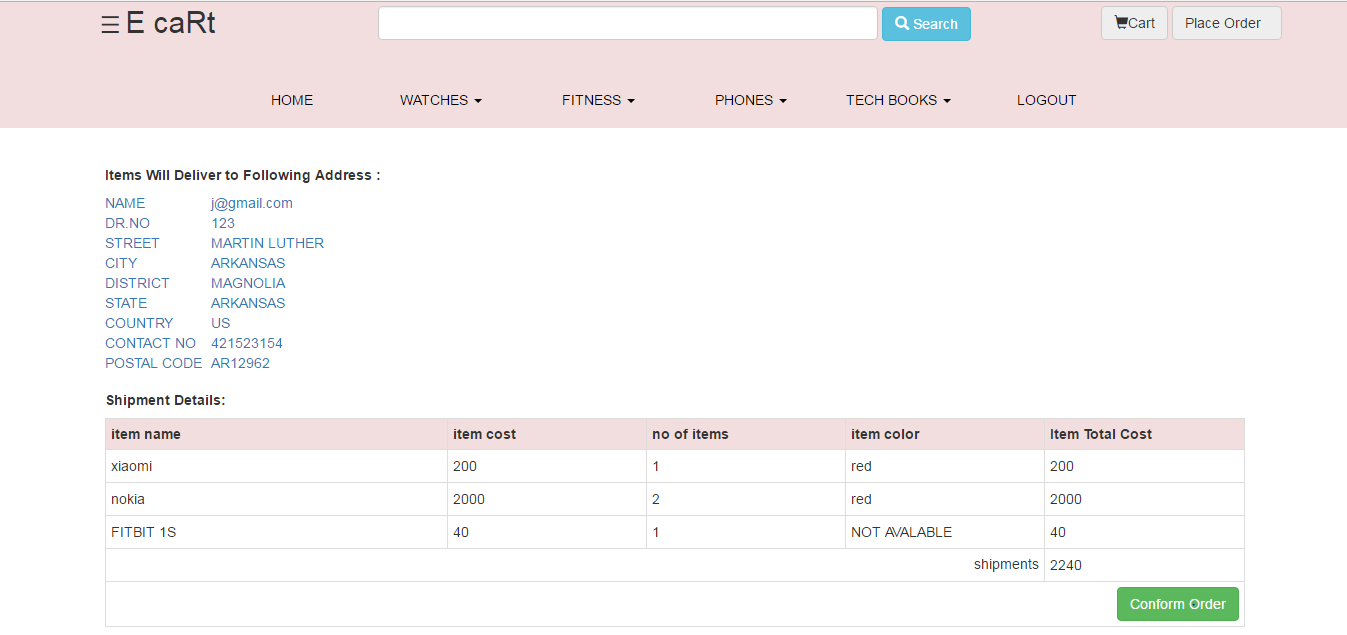


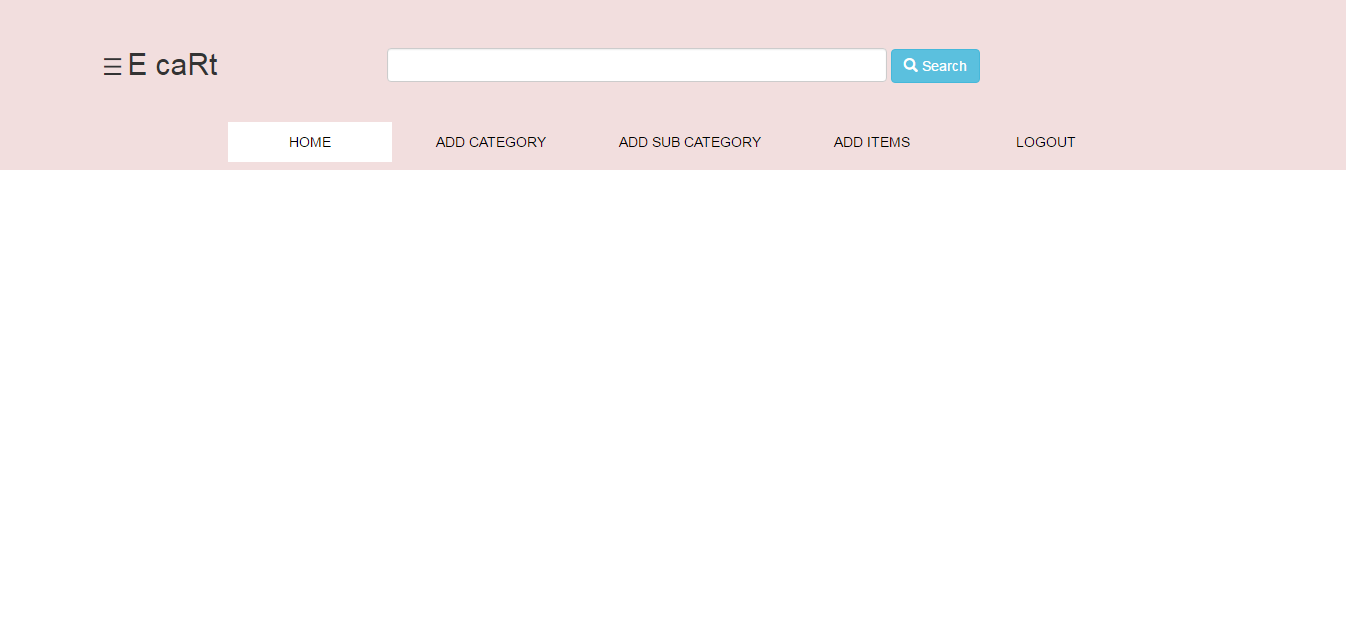
Image views :



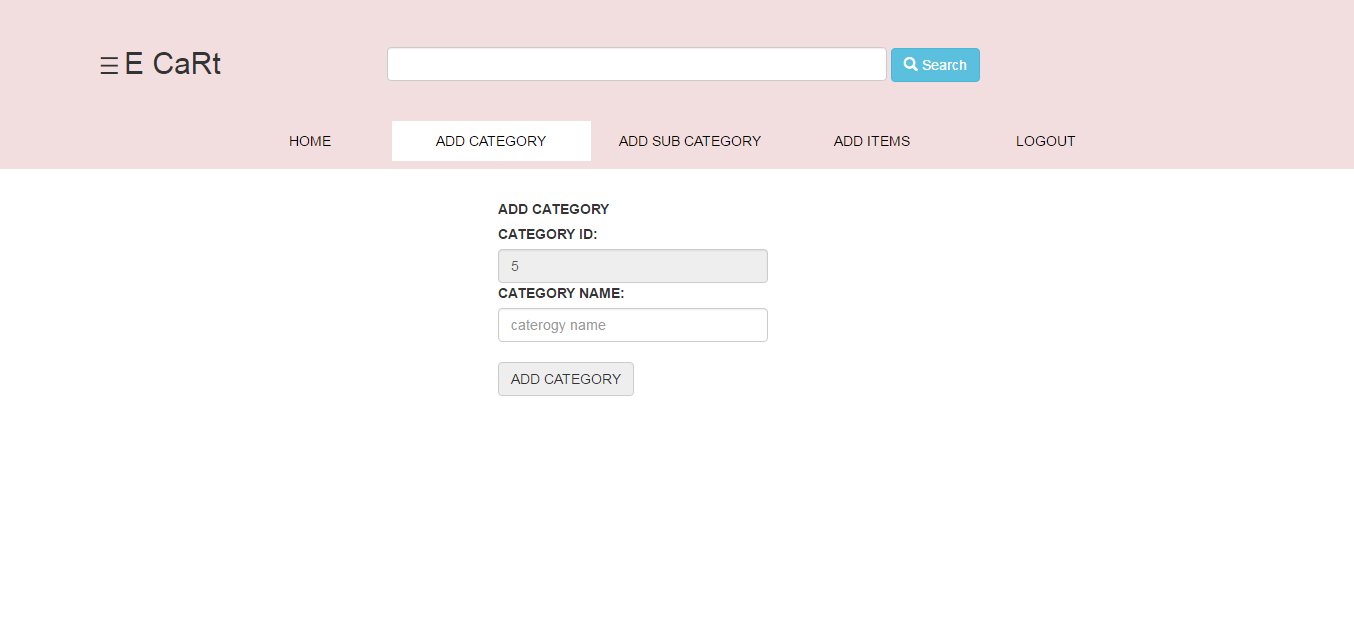
View Order Details:



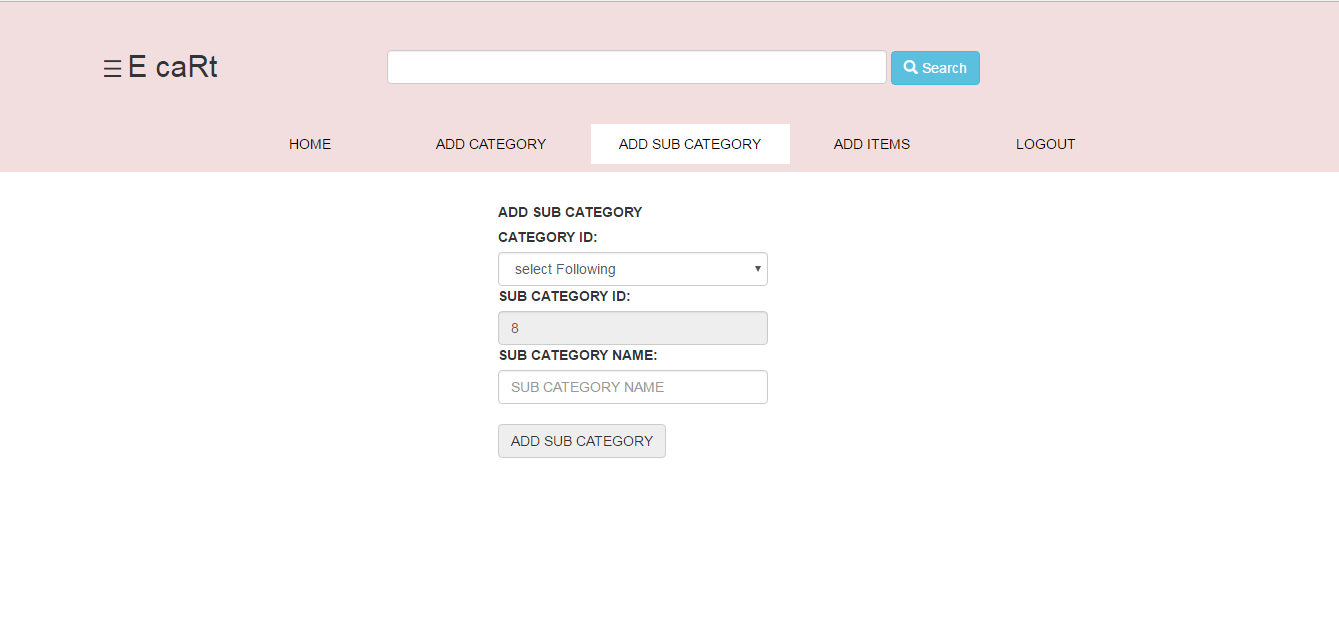
Admin Menu:



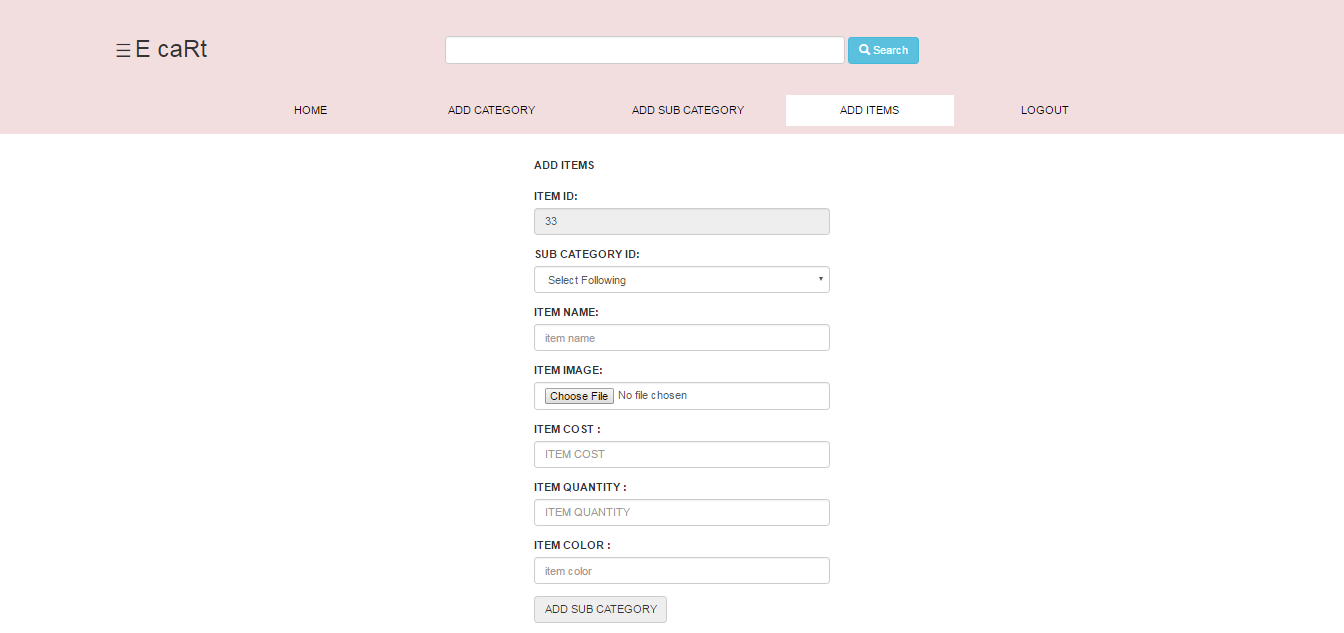
**Add Category:**

****

**Add Sub Category Names:**

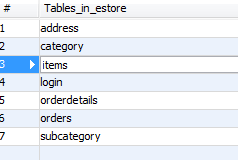
****

**Add Items:**

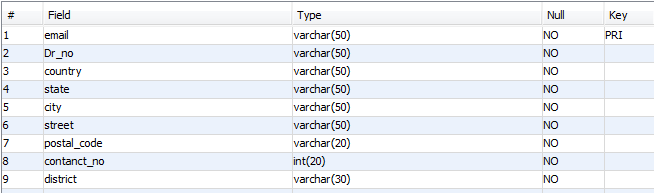
****

**MYSQL Screen Shots:**

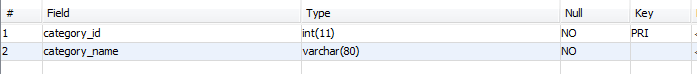
The SQL Server consisting of following table:



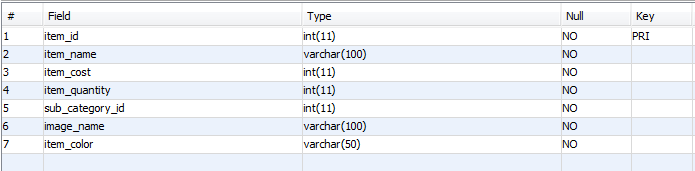
**Address Table:**

****

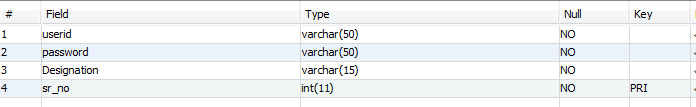
**Category Table:**

****

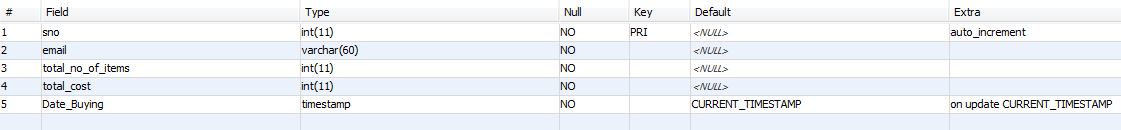
**Items Table:**

****

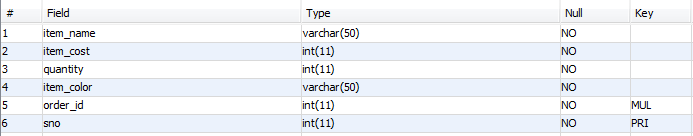
**Login Table:**



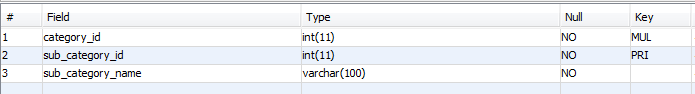
**Orders Table**



**Order Details:**

****

**Sub Category Details:**

****

Conclusion:

We hope this shopping Cart will become a successful interface for business to open up small, online boutiques. Its stability, customizability, and user-friendly interface make it an ideal choice for facilitating online transactions and creating a satisfying customer experience.

Reference:

1) http://www.w3schools.com/html/default.asp

2) http://www.w3schools.com/css/default.asp

3) <http://www.w3schools.com/bootstrap/default.asp>

4) <http://www.w3schools.com/sql/default.asp>

5) http://www.w3schools.com/php/default.asp