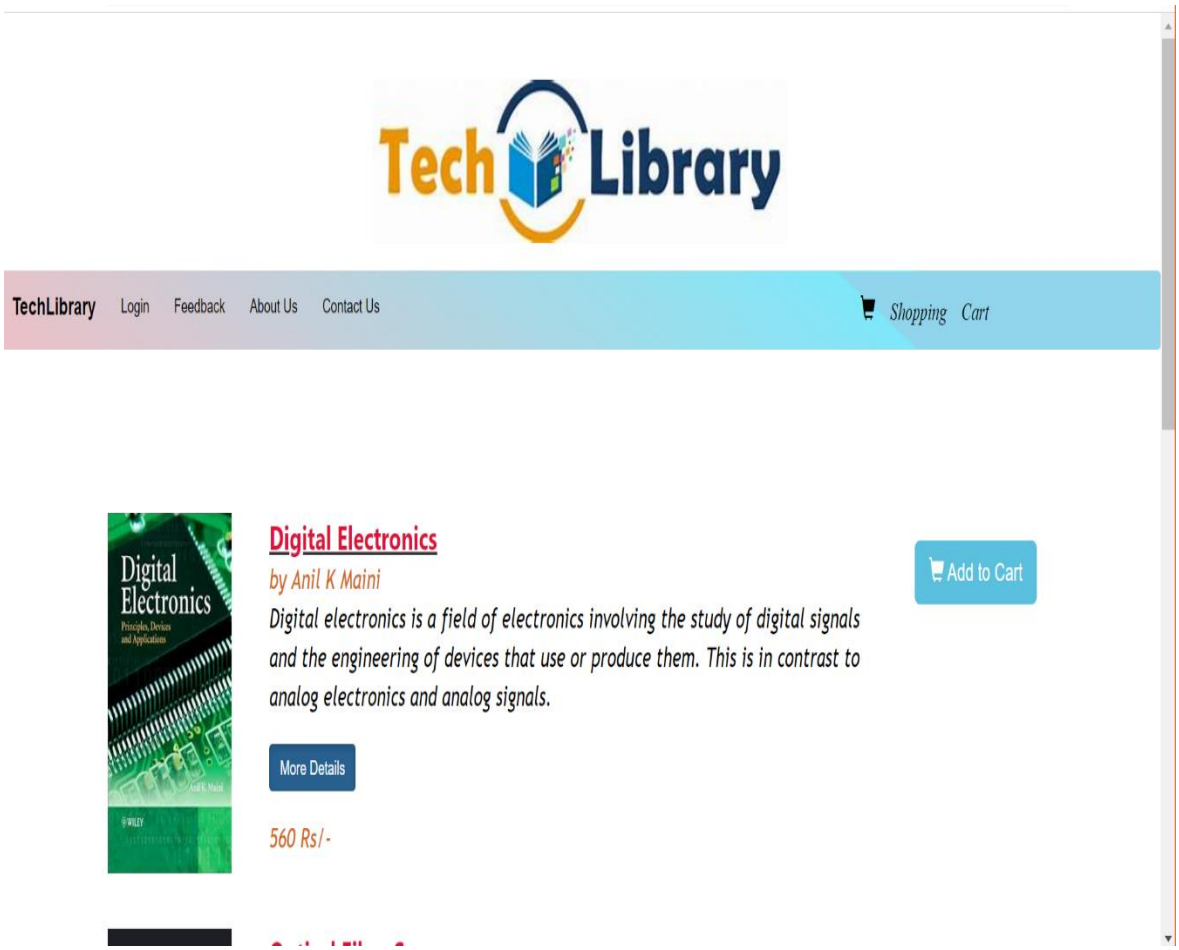
Project: Tech-Library

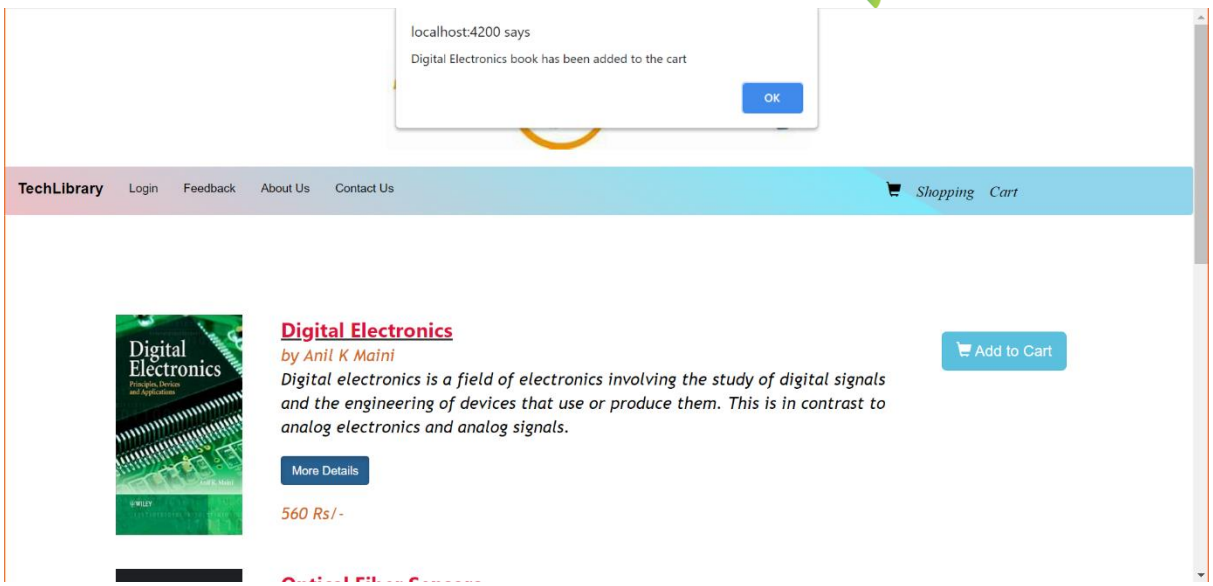
Front End Development

TechLibrary is an E-commerce application which provides customers with an opportunity to buy books. It is an online platform which deals with new and old books of all fields. It maintains the database of new books and records the books that have been purchased by customers as order history. The admin of the website purchases the books from the vendor and displays it to the customer or the buyer. The order will be placed by the customer when the login is successful. After he has placed his order, he can proceed with the payment.

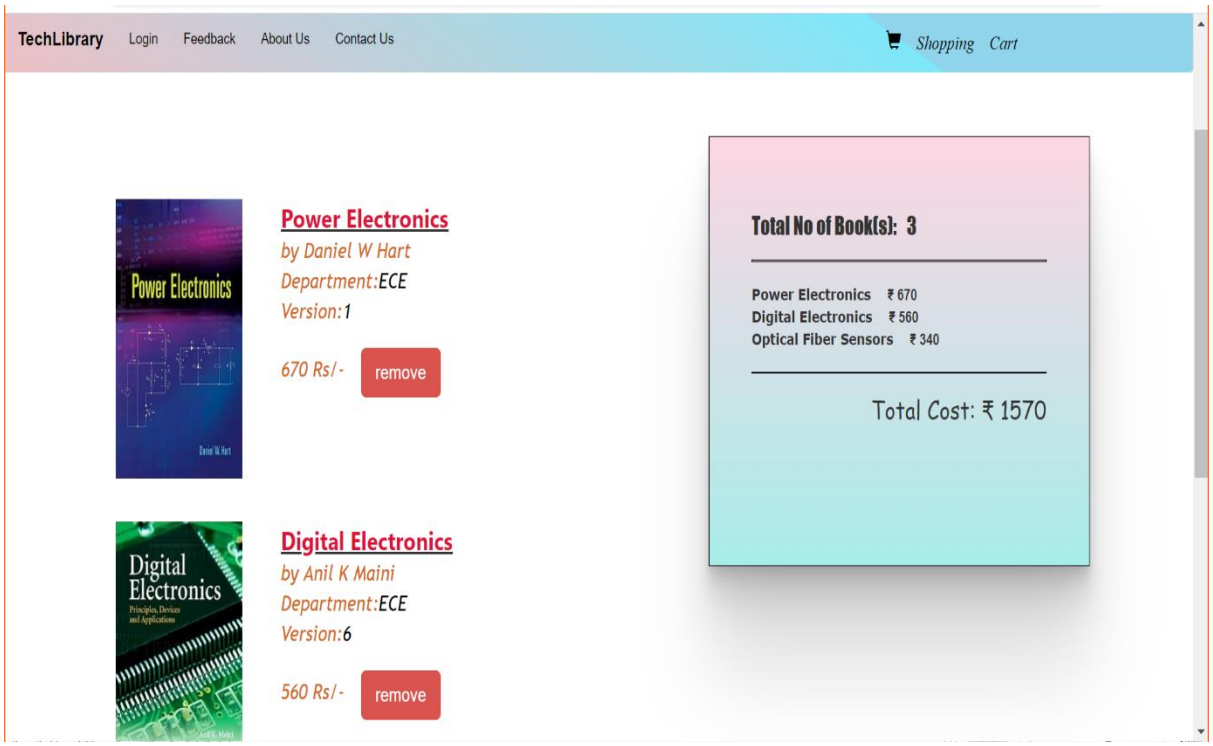
Once the application is loaded the first screen appears as below:

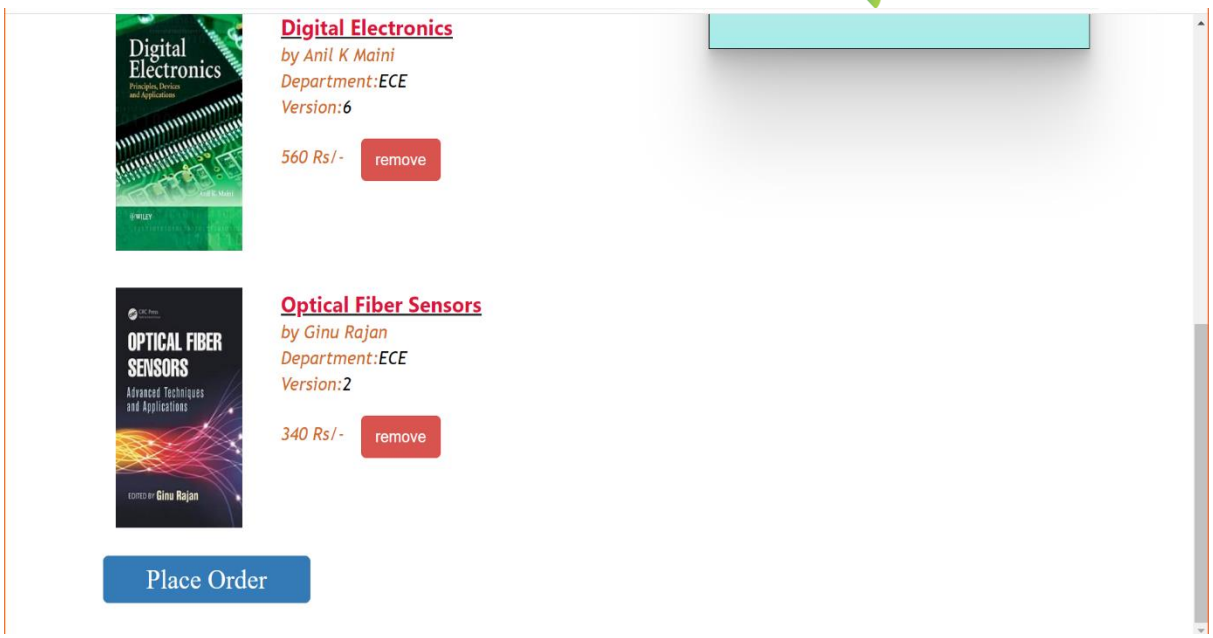


This is the home page. All the book products from “Book” entity will be displayed in this page. To view this page, the customer need not login.

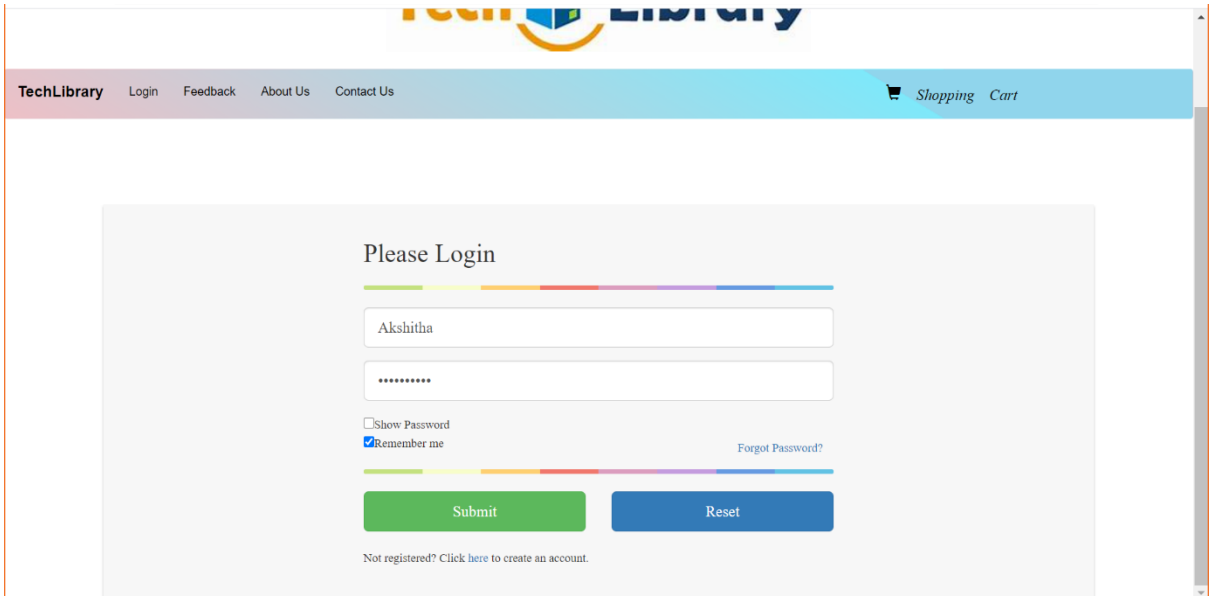


If the customer is willing to buy any book product from the application, then the respective book should be added to the cart. When the book is added to cart, then alert box will be displayed to confirm that it is being added to the cart.





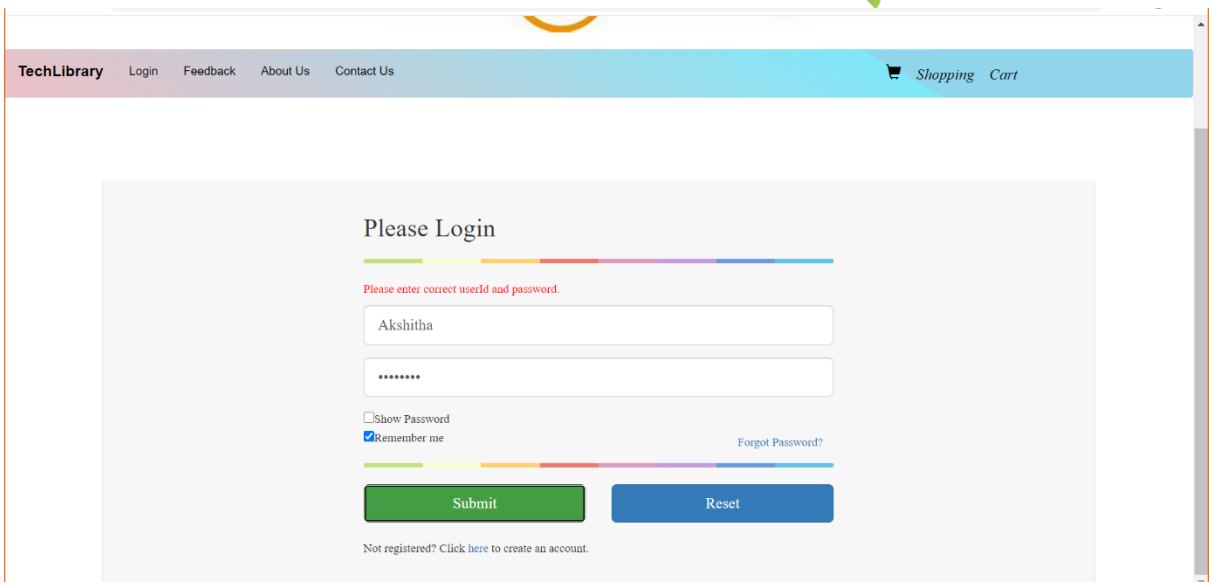
This is the shopping cart page. These are the books a customer has added to order. Later if they are not willing to buy certain book, then they can "remove" that book from the shopping cart by clicking on remove button. They can proceed to the payment by clicking on "place order" button.



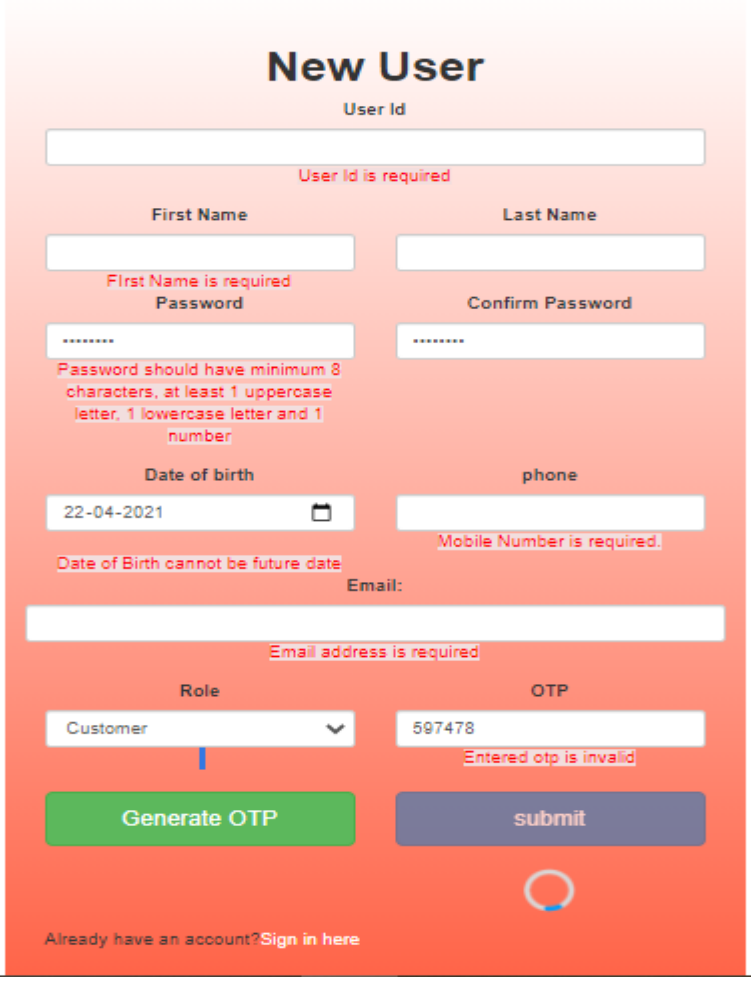
If the customer hasn't logged in while placing his order, he/she will be redirected to the login page. If he already has an account then he will login, otherwise he needs to create an account in sign up page.

There is a "show password" check box to toggle between text and hidden password. The "remember me" check box is used to save the login credentials of use rid and password.

After the customer has logged in, the "login" field in the navigation bar will change to "logout".



When the login credentials are incorrect, the warning message will be displayed.



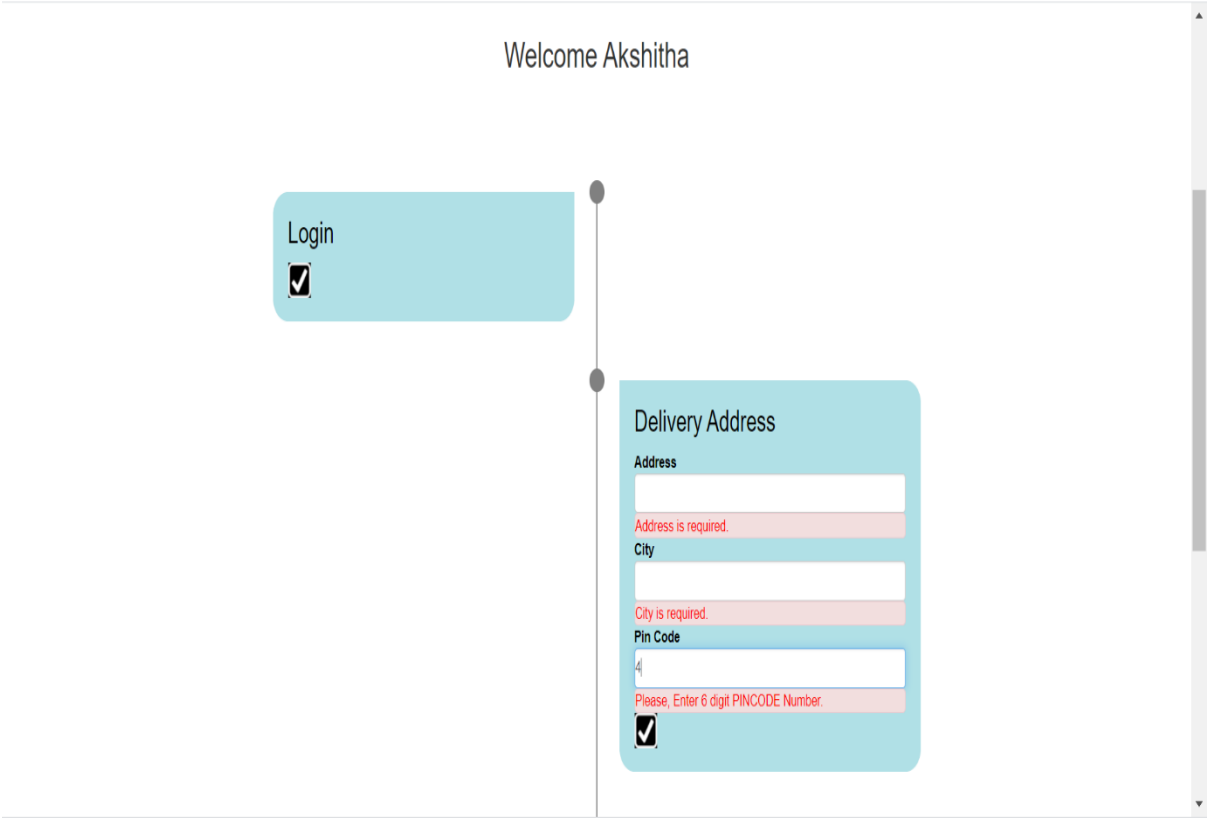
The above is the validation for the sign up page. All the fields in the sign up form is required to be filled with some instructions mention below the textbox.

The “userId” is required as the customer has to login using the username every time he/she logins in the application. The “firstname” is a required data to be filled. The “lastname” is an optional data. The “password” must have a minimum of 8 characters, should contain 1 uppercase, 1 lowercase and 1 number. The “confirm password” checks if both the passwords entered by the customer matches. The “Date of Birth” should not contain a future date. If the “Date of Birth” contains future date the warning message is displayed. “Phone” must contain 10 digits. The “email” must contain @ symbol, after a “.” it should contain atleast two characters. The “role” of the person logging in can be either a customer or an admin.

This is the Sign up page for a customer or an admin to create an account. The password is encrypted using “SHA-256” (Secured Hash algorithm). The encrypted password will be stored in the database.

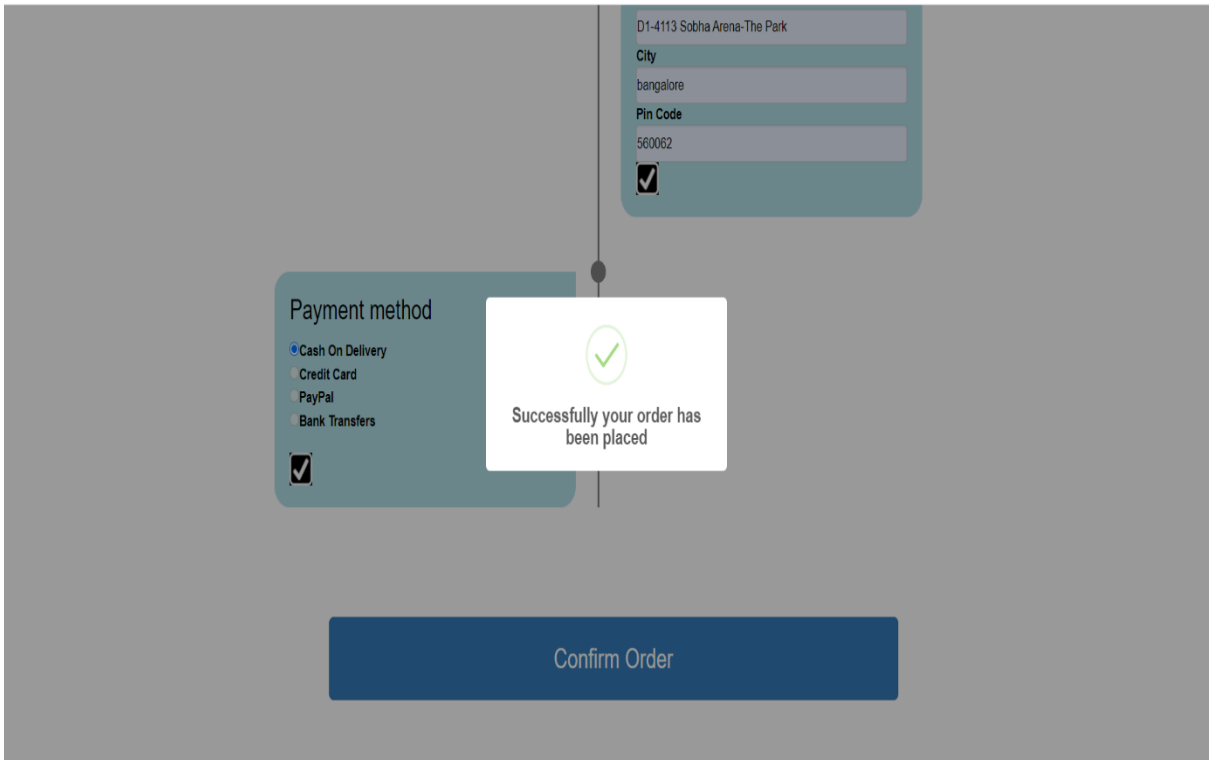
When the “Generate OTP” button is clicked, the OTP message will be sent to the email address mentioned in the above text box. When the entered OTP is invalid the message “Entered OTP is invalid” will be displayed.

After entering all the details in the required format, then the account will be created and the customer will be directed to the home page.

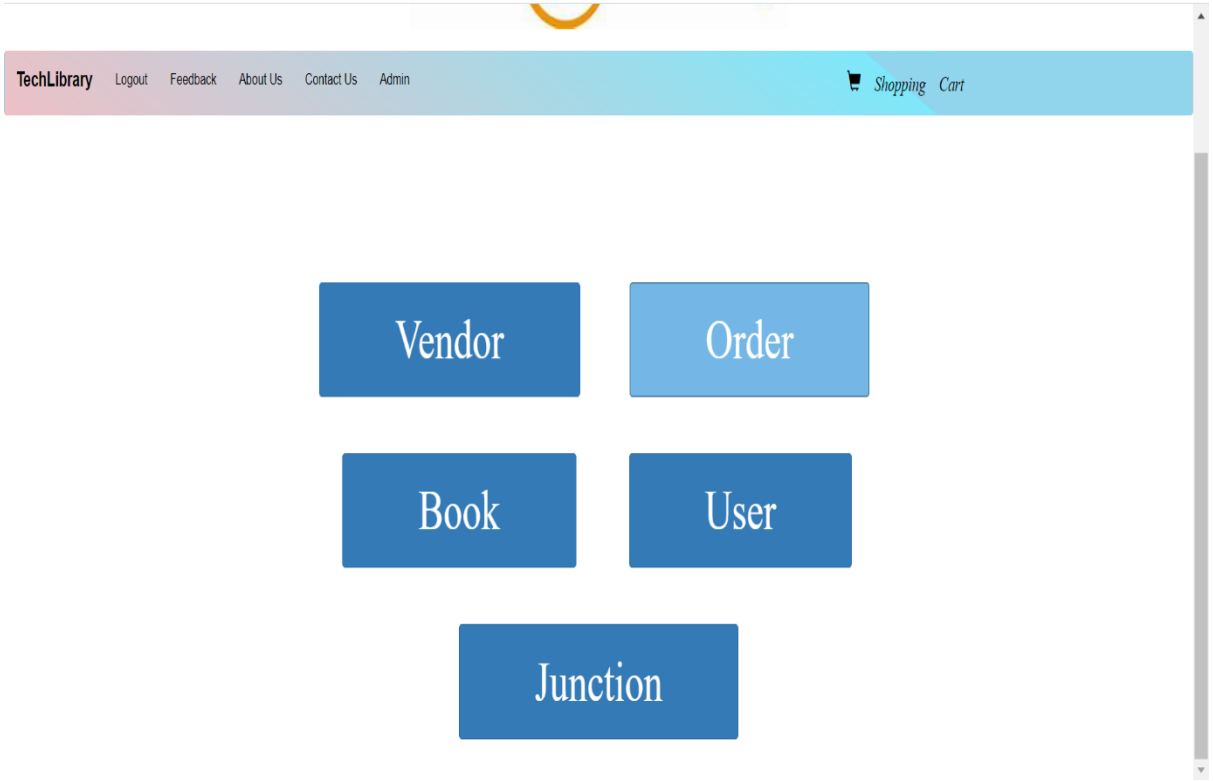


This is the payment page where login credentials is compulsory to order any book. After the login, the delivery address is required to order. The pincode has to be 6 digits and cannot be left blank.

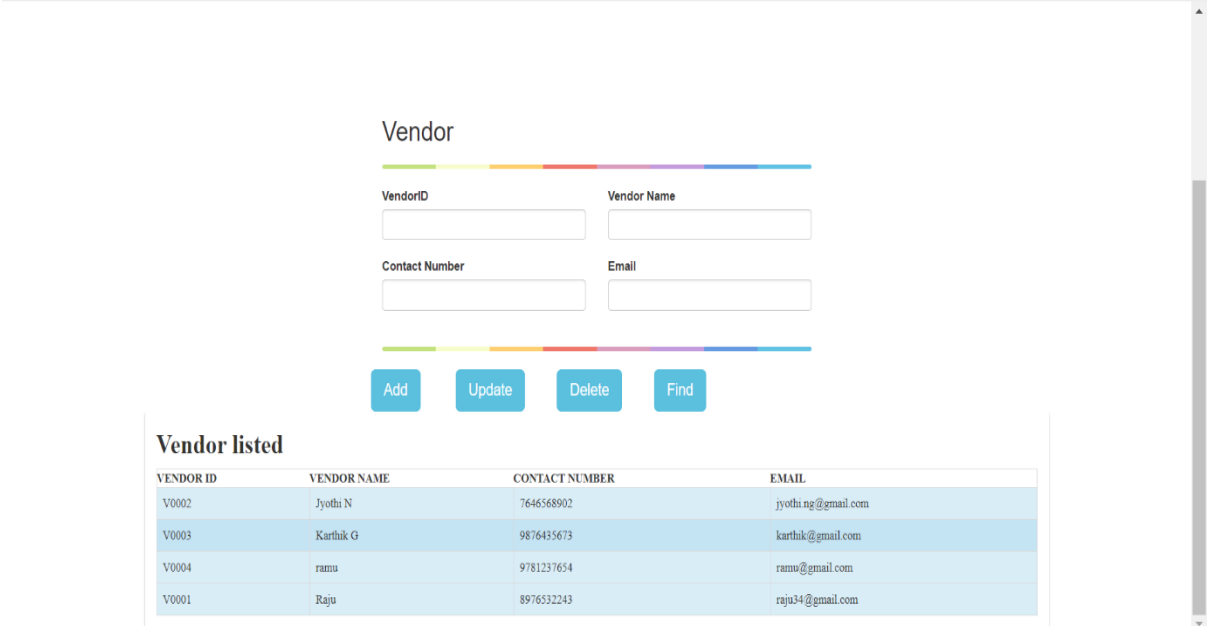
The payment method “Cash on Delivery” is selected by default. After the payment method, the customer needs to click on “confirm order” button to place the order.



Once the customer has entered the valid "address", the confirm order button is enabled and the dialog box appears confirming that the order has been placed successfully.

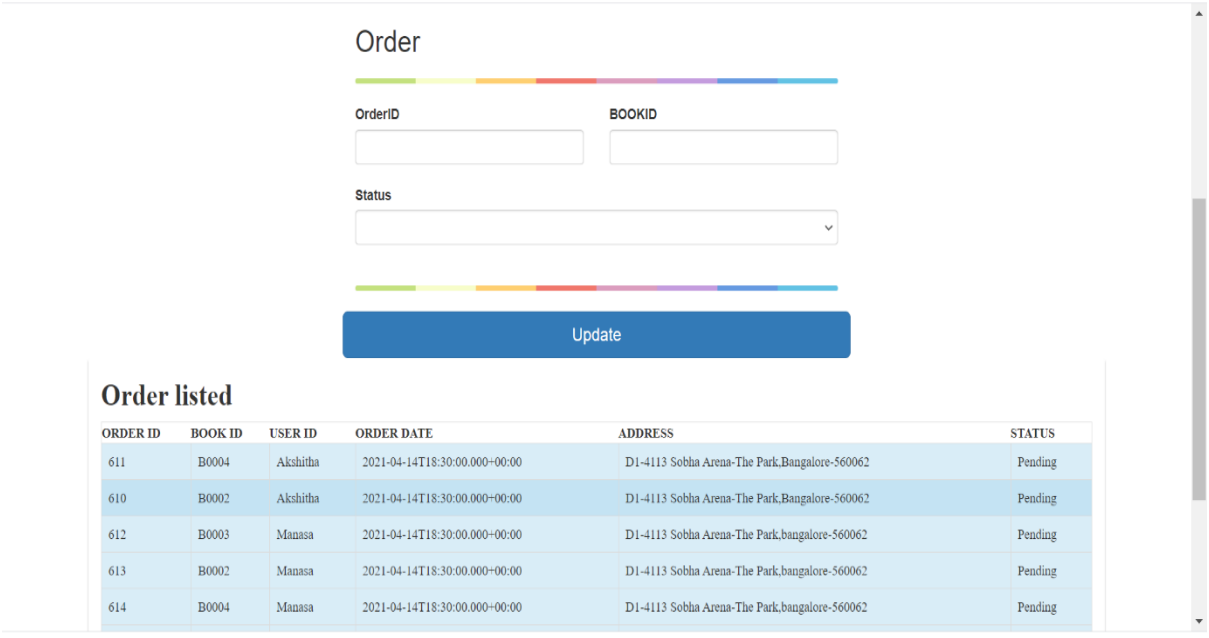


Only when the user is logged in as "admin" the admin field in the navigation bar will be displayed. For "customers" the admin field will not be visible.



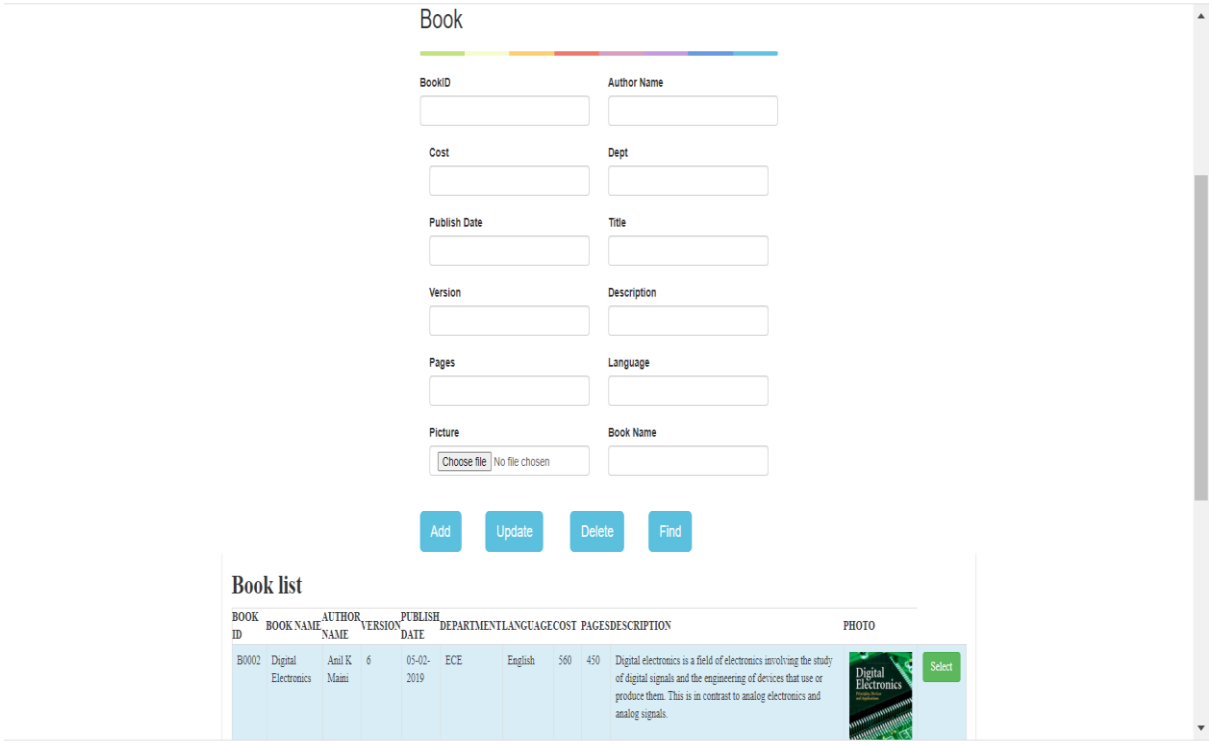
The Vendor entity is filled by the admin. The admin can perform CRUD (add, update, delete, find by vendorId) operations on the vendor table. The vendor table contains the list of vendors, who sells the books in the web application.

Here, VendorId is the primary key and all the other attributes will contain the information about the vendor.

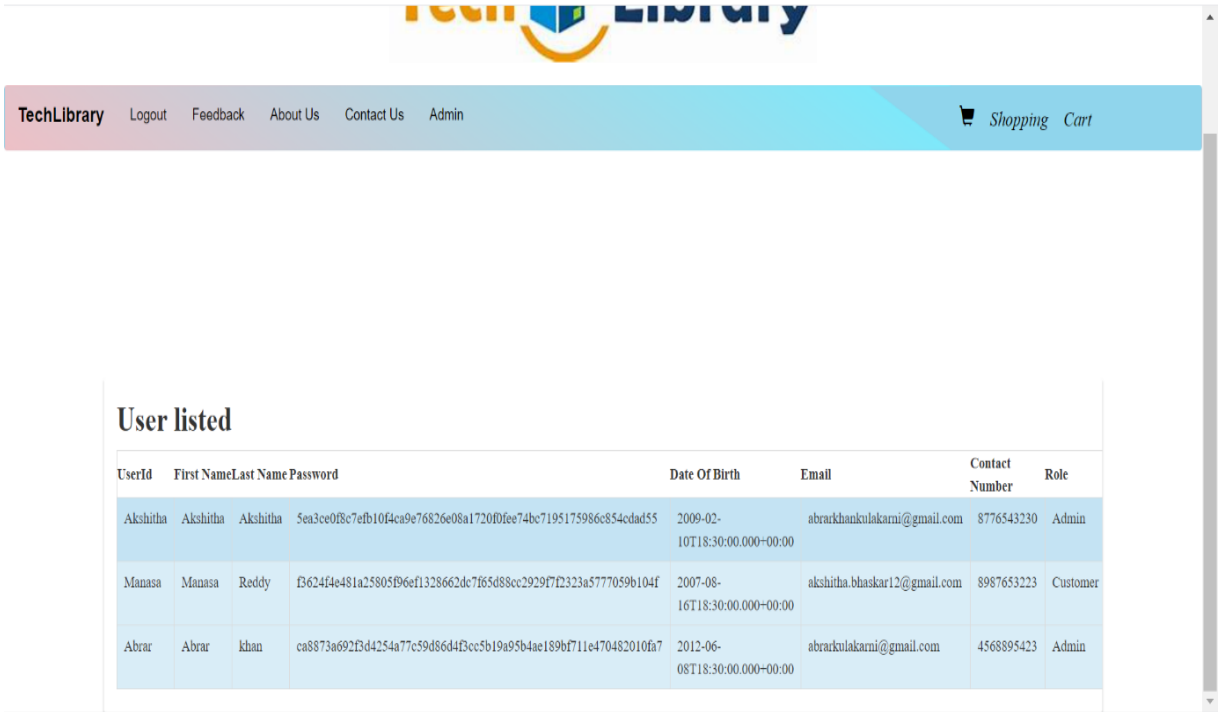


After the customer orders the books, then the bookId for the book which they have ordered and the userId of the customer along with the date on which they had ordered will be added to the “orders” table. OrderId will be auto-generated and will be the primary key.

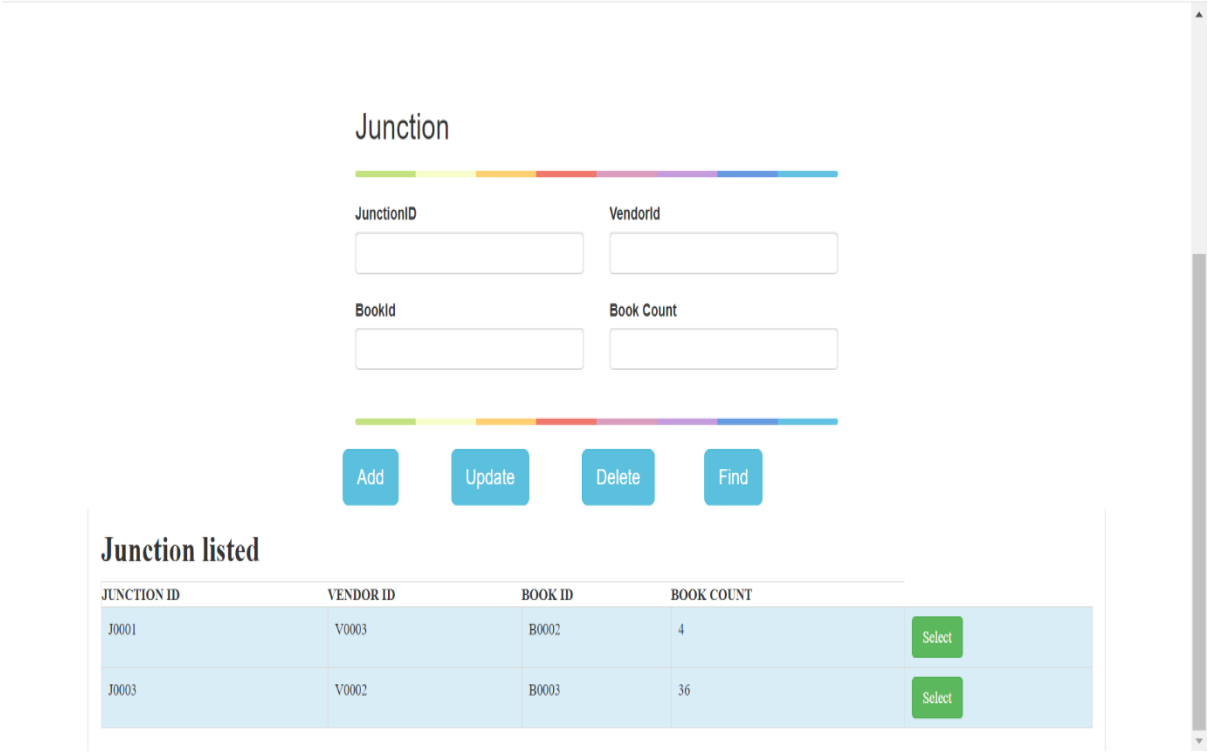
The admin will have authority to change the status to “pending, shipped and delivered” based on the orderId and bookId.



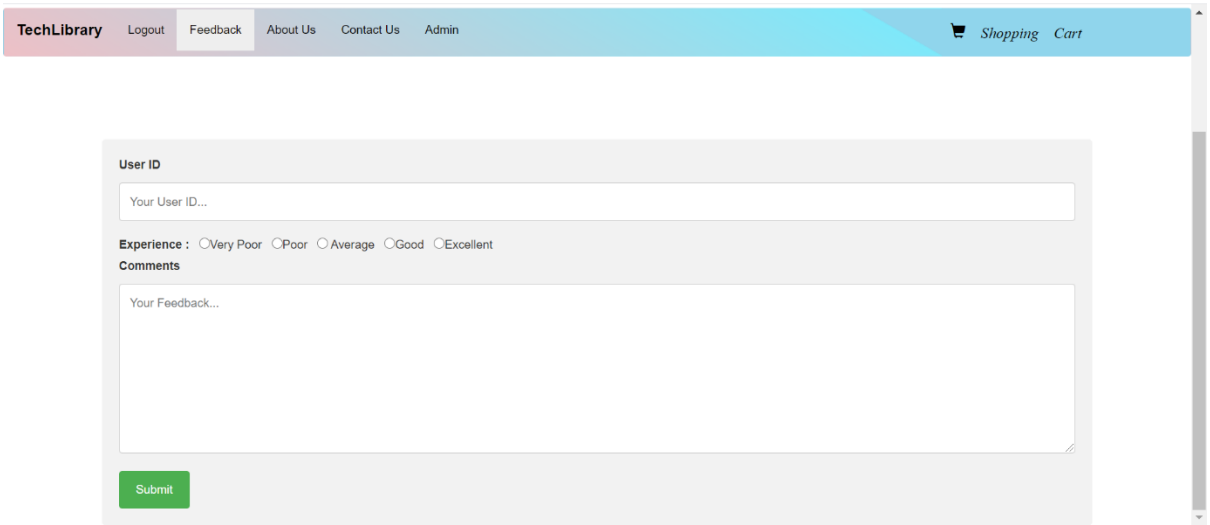
The books are stored in the database. The admin can again perform CRUD



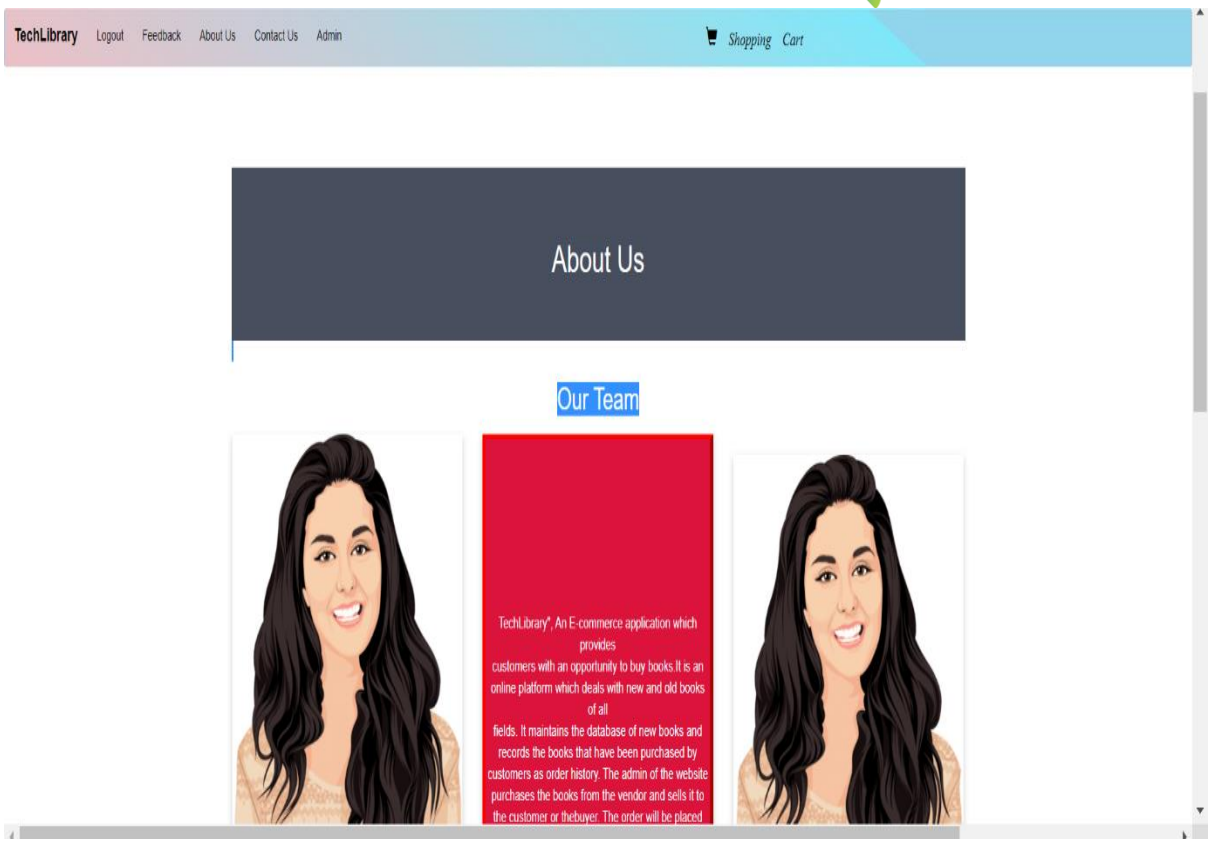
All the users who have an account will be displayed in this page.



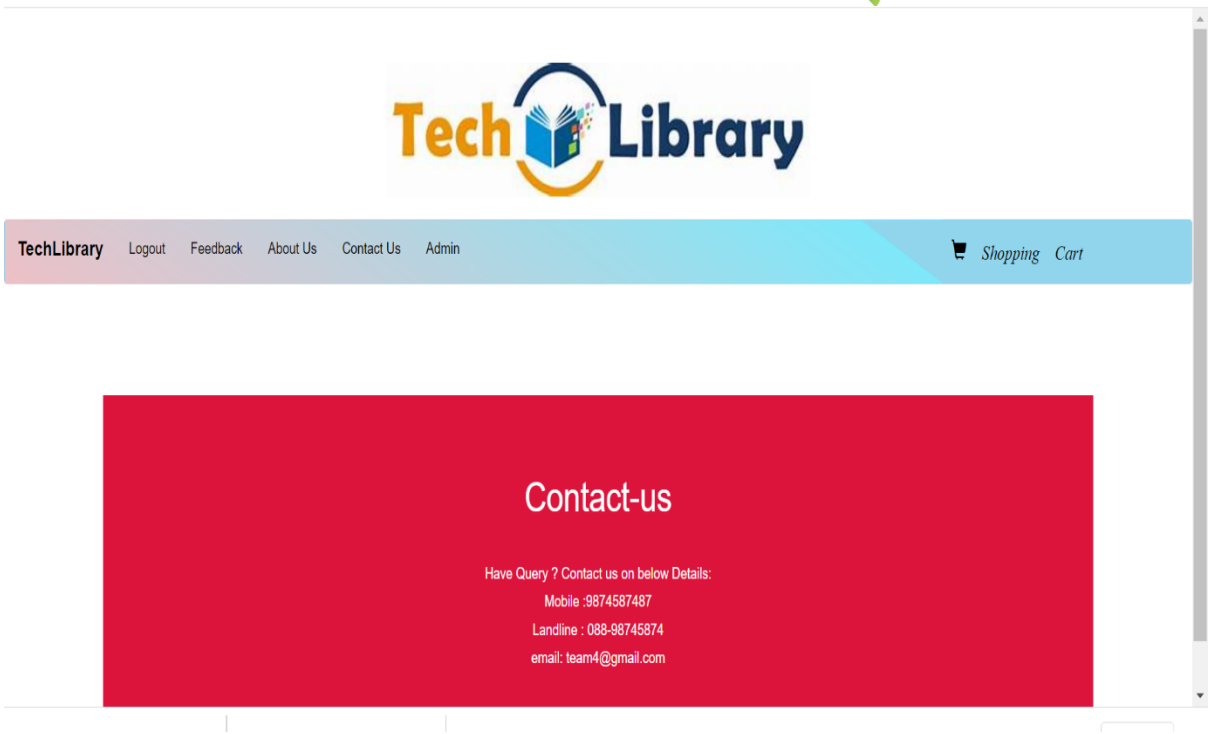
The junction table is created because each vendor may supply many books and bookId column cannot be a multivalued. Hence, junction table is created where junctionId is primary key and vendor Id is the foreign key of vendor entity and bookId is the foreign key of book entity. The admin can perform CRUD (add, update, delete, find by junctionId) operations on the junction table.



The "feedback" is given by the customer and the details entered will be stored in the "feedback" entity in the database.



The above page is the “about-us” page which has information about our project.



The above page is the “contact-us” page.