

customerRelationshipManagement

Above project uses Spring MVC along with Hibernate ORM to perform basic CRUD operations on Customer Relationship Management Portal.

The model *Customer* has 3 visible as well as 1 hidden attribute.

- They are ->
1. *Id* (hidden)
 2. *First Name*
 3. *Last Name*
 4. *Email id*

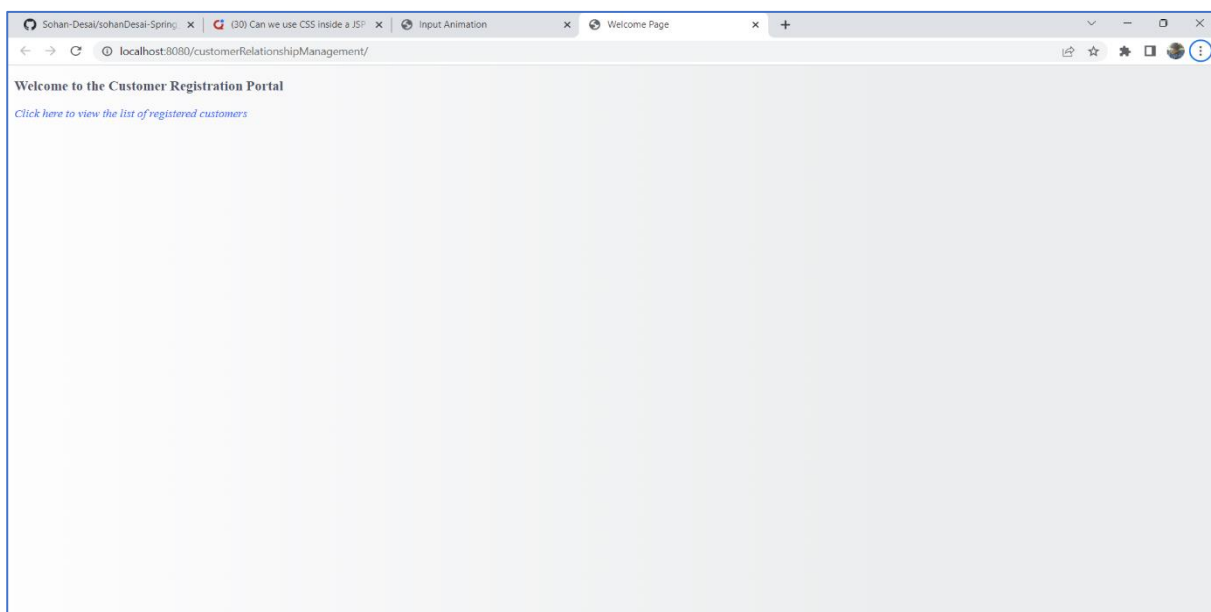
The application performs following functions ->

1. **Create** a new Customer record
2. **Read** and display existing records
3. **Update** existing records and save any changes
4. **Delete** from existing records

We shall further see each of these functionalities in greater detail

The Main page

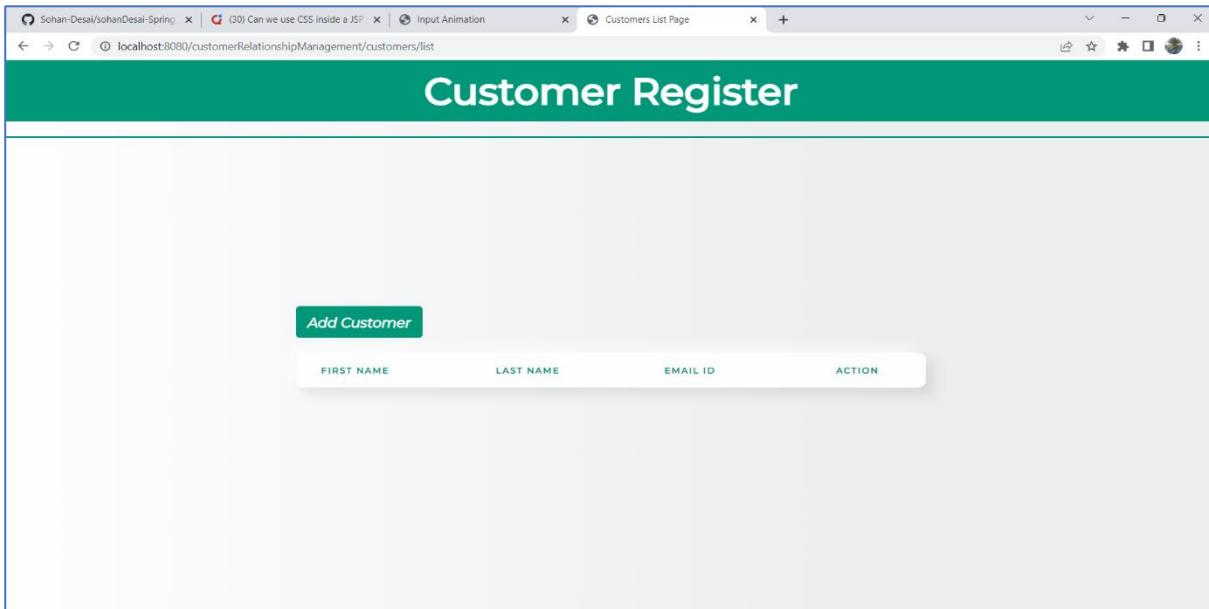
This is the default page that gets loaded when the application is booted on Tomcat server. It has a welcome message and link to **Customer** list page.



Customer List page

Upon clicking on afore mentioned link, we arrive at Customer list page. This page contains a header, a table and an “**Add Customer**” button.

There are no records to be shown yet.



The table has 4 columns. First three of which are customer attributes whereas the fourth column is an action column. This column will display the action buttons viz. **Update** and **Delete**.

Clicking on **Add Customer** button takes us to customer form page.

Customer Form page

A screenshot of a web browser showing a page titled "Customer Register". The page has a green header bar with the title. Below the header, there is a light gray area containing a form titled "CUSTOMER DETAILS FORM". The form has three input fields: "First Name" with a placeholder "Eg: John", "Last Name" with a placeholder "Eg: Smith", and "Email id" with a placeholder "Eg: john@xyz.com". Below the input fields is a green button labeled "Submit". Below the button is a link labeled "Back to view list of customers".

This page has a header row displaying text “CUSTOMER DETAILS FORM”. Below it are the 3 fields along with their labels followed by the **Submit** button

Customer Register

CUSTOMER DETAILS FORM

First Name

Harshit

Last Name

Choudhary

Email id

harshit@greatlearning.in

Submit

[Back to view list of customers](#)

The submit button and text fields have Neumorphic design. The **Submit** button changes its appearance when cursor hovers over it.

Customer Register

CUSTOMER DETAILS FORM

First Name

Harshit

Last Name

Choudhary

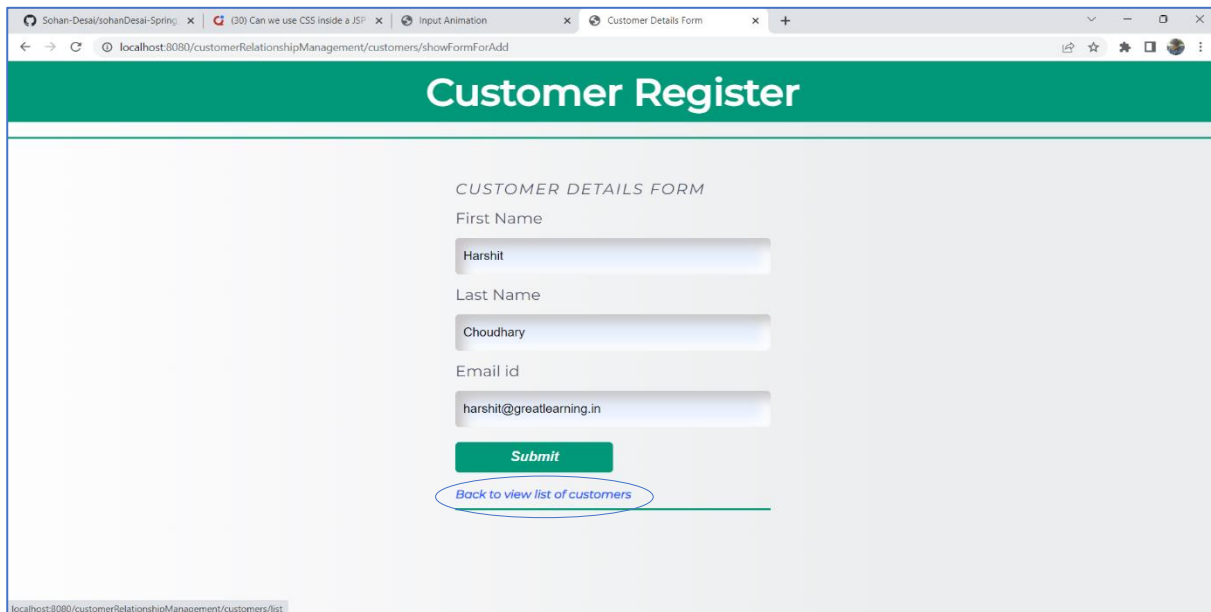
Email id

harshit@greatlearning.in

Submit

[Back to view list of customers](#)

At the very end is provided a link to go back to customer list page.



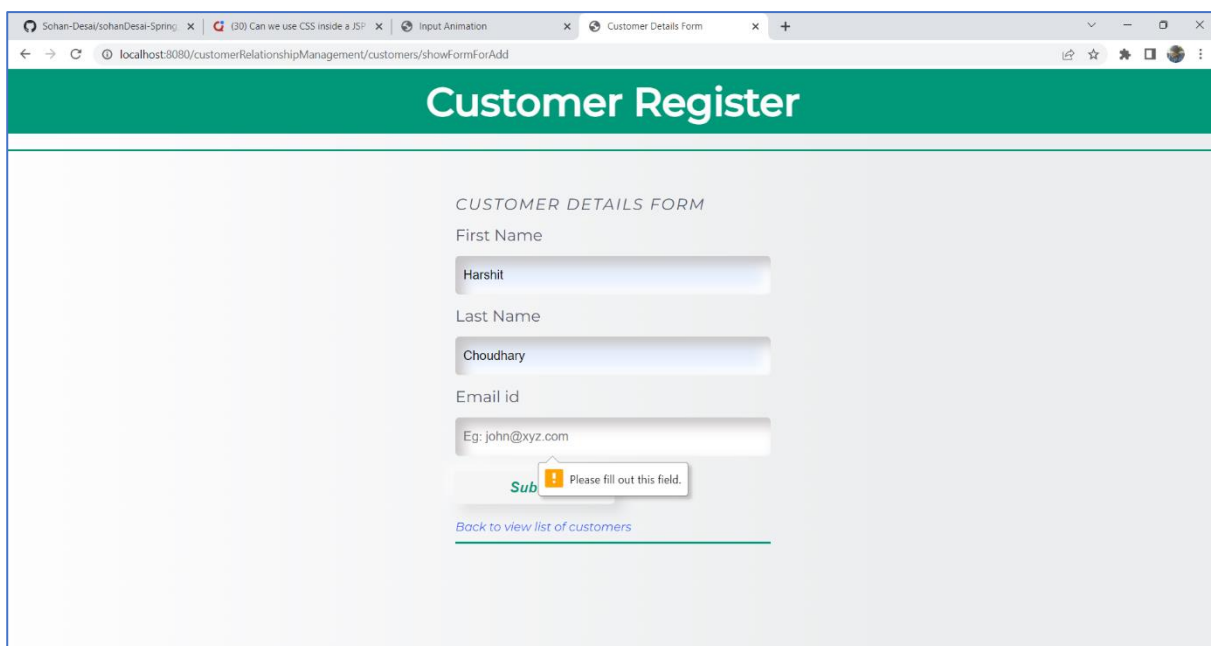
The screenshot shows a web browser window with the URL `localhost:8080/customerRelationshipManagement/customers/showFormForAdd`. The page has a green header with the text "Customer Register". Below the header, the form is titled "CUSTOMER DETAILS FORM". It contains three input fields: "First Name" with the value "Harshit", "Last Name" with the value "Choudhary", and "Email id" with the value "harshit@greatlearning.in". A green "Submit" button is located below the email field. Below the "Submit" button, there is a blue link that says "Back to view list of customers". The browser's address bar shows the URL `localhost:8080/customerRelationshipManagement/customers/list`.

On clicking the **Submit** button, the newly entered records are displayed on customer list page.

#Note->

The email field is a required field which has to be populated before submitting the details.

*A warning message displays when we click on **Submit** button without entering data as per convention in the email field.*



The screenshot shows the same web browser window as before, but the "Email id" field now contains the placeholder text "Eg: john@xyz.com". The "Submit" button is now disabled and has a red border. A red warning message box is displayed above the button, saying "Please fill out this field." Below the button, there is a blue link that says "Back to view list of customers". The browser's address bar shows the URL `localhost:8080/customerRelationshipManagement/customers/showFormForAdd`.

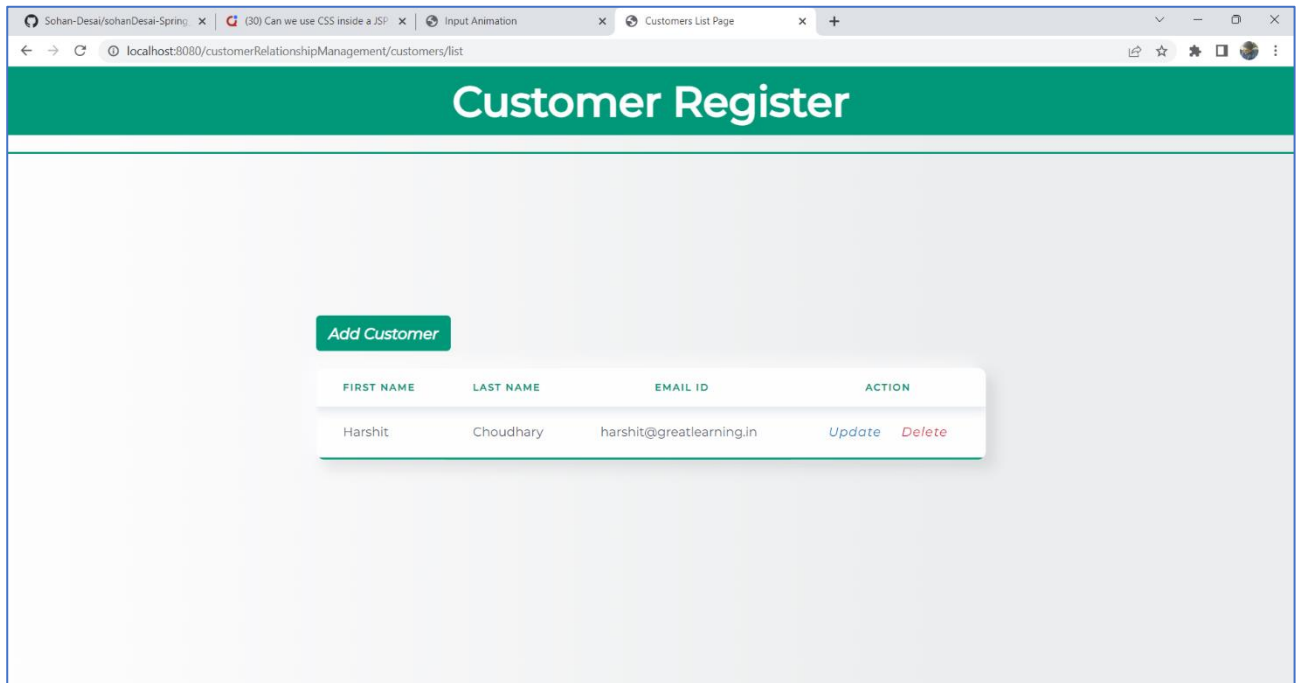
Email id is a required field

The email field checks whether the string contains @ inside it flanked on both sides by strings of characters.

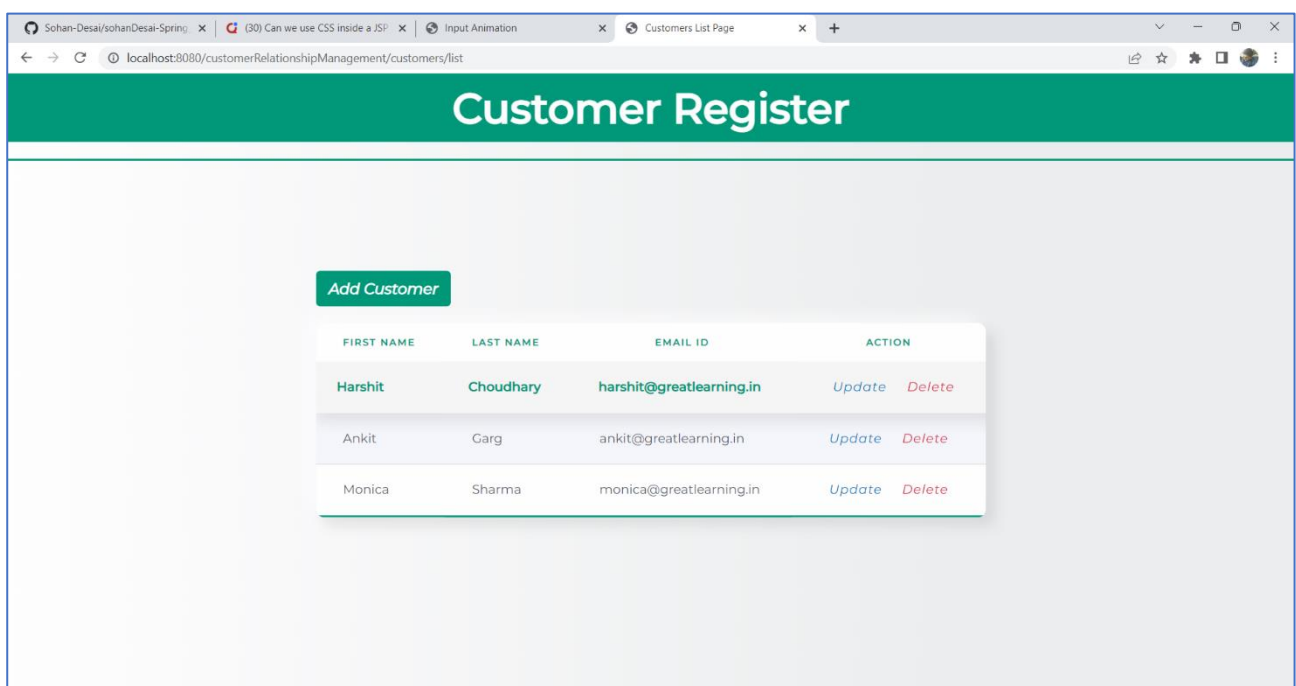
*User can **Submit** details only upon entering email id as per convention.*

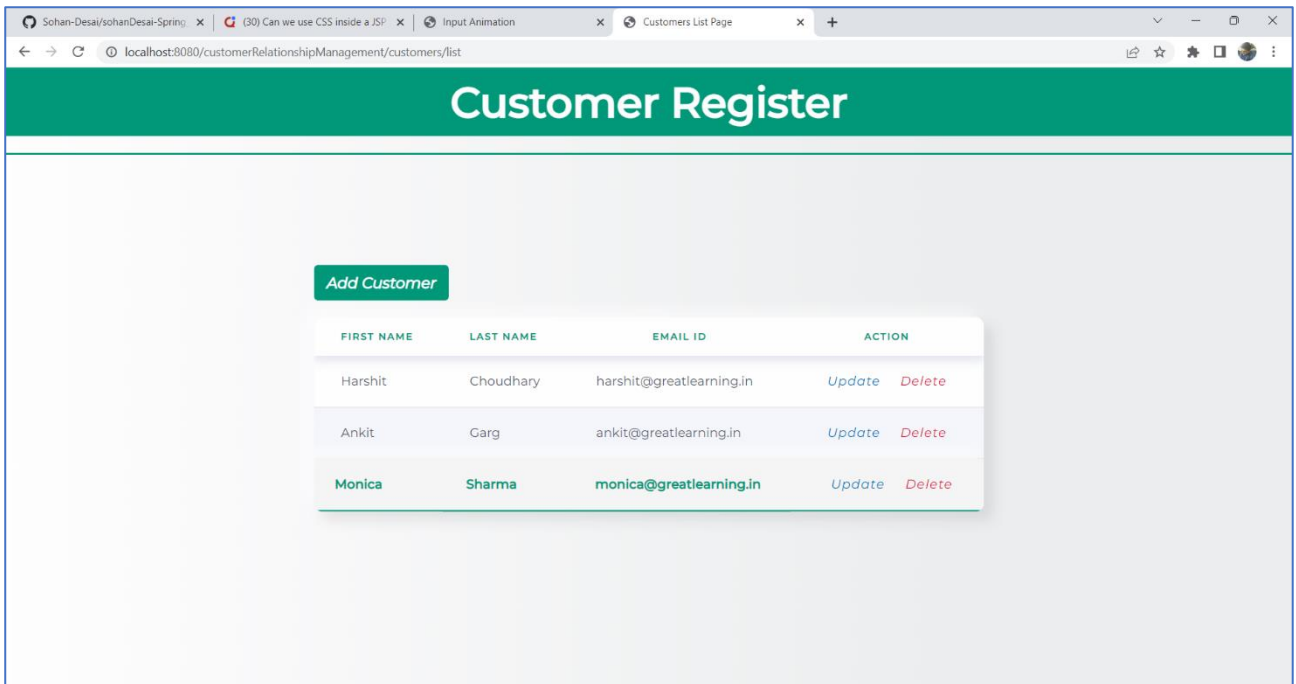
Customer table

The entered records get displayed on customer list page.



User can further add more records.

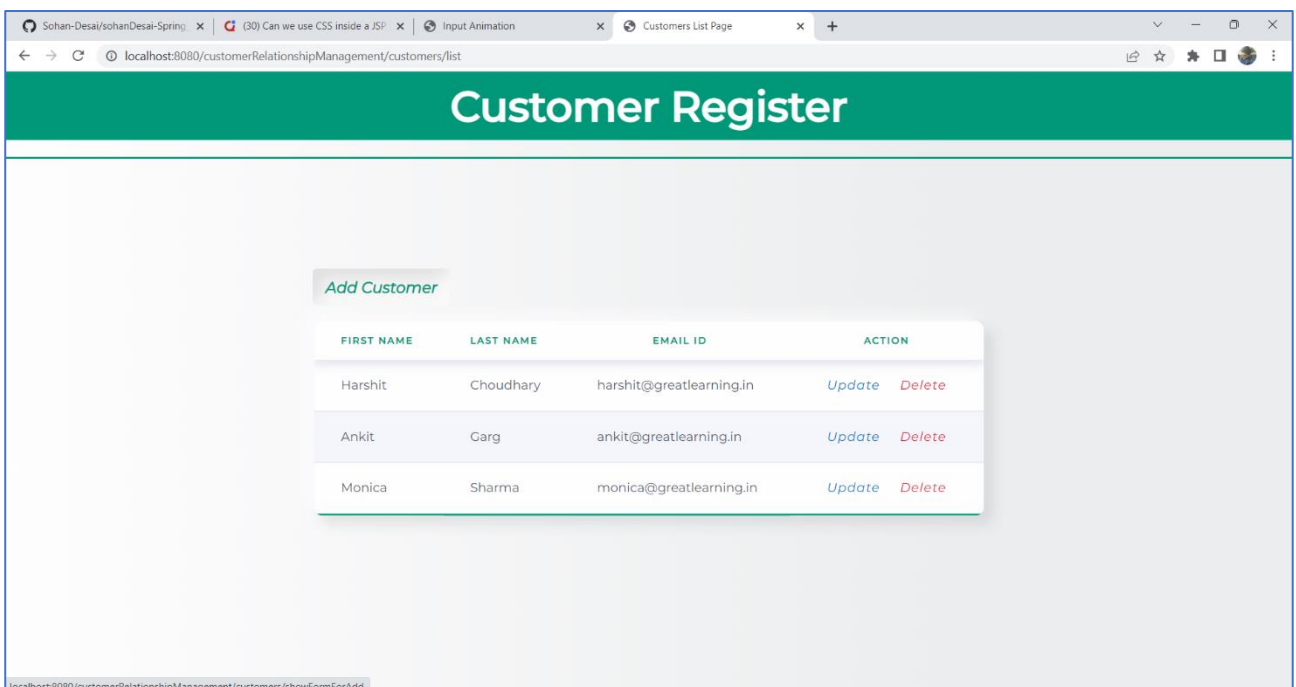




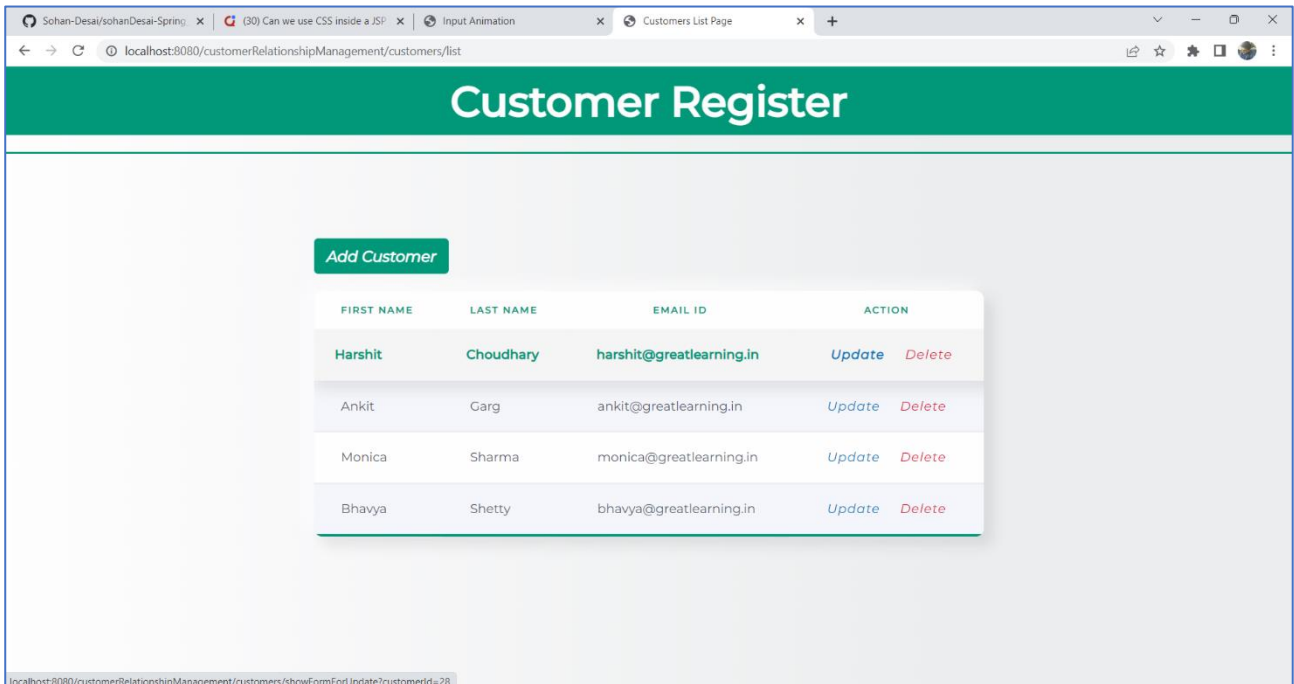
Both the **Add Customer** and table body rows change appearance when cursor hovers on them.

While the rows get slightly enlarged (by 2%) and the inline text changes in colour, size and thickness.

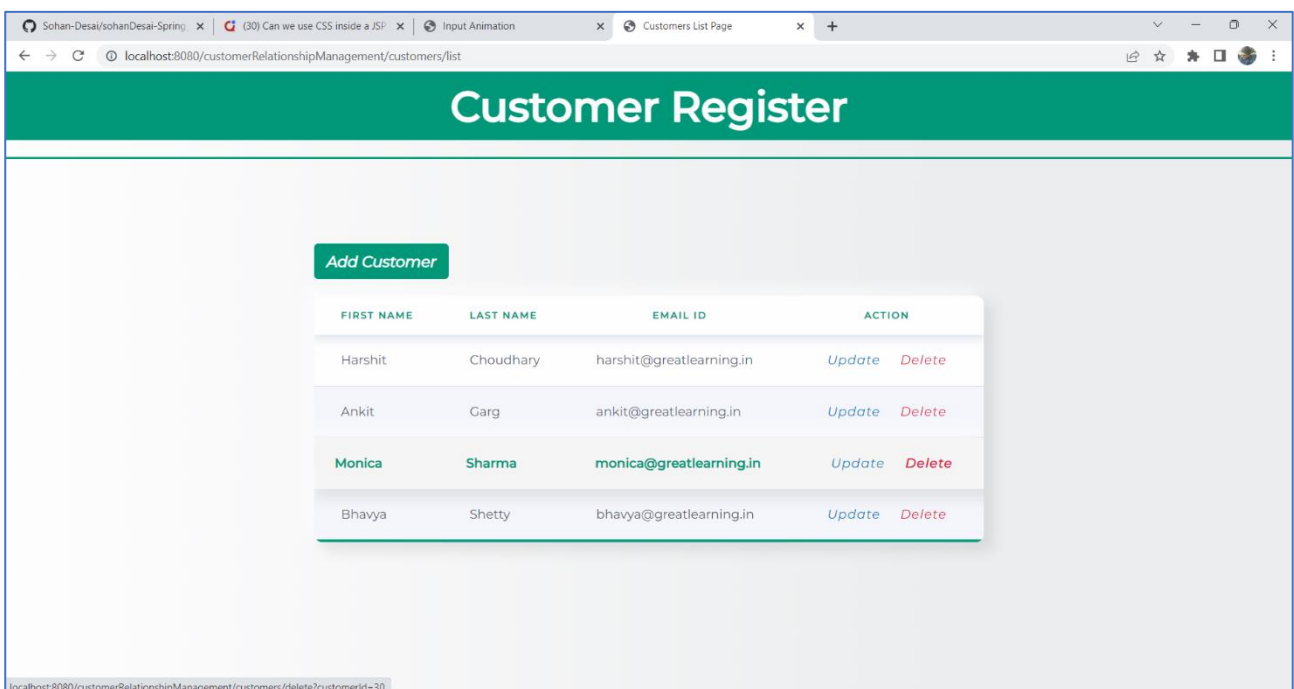
Whereas the **Add Customer** button takes appearance of an inset button.



Similarly, the **Update** and **Delete** hyperlinks increase in font-weight when the cursor hovers on them.



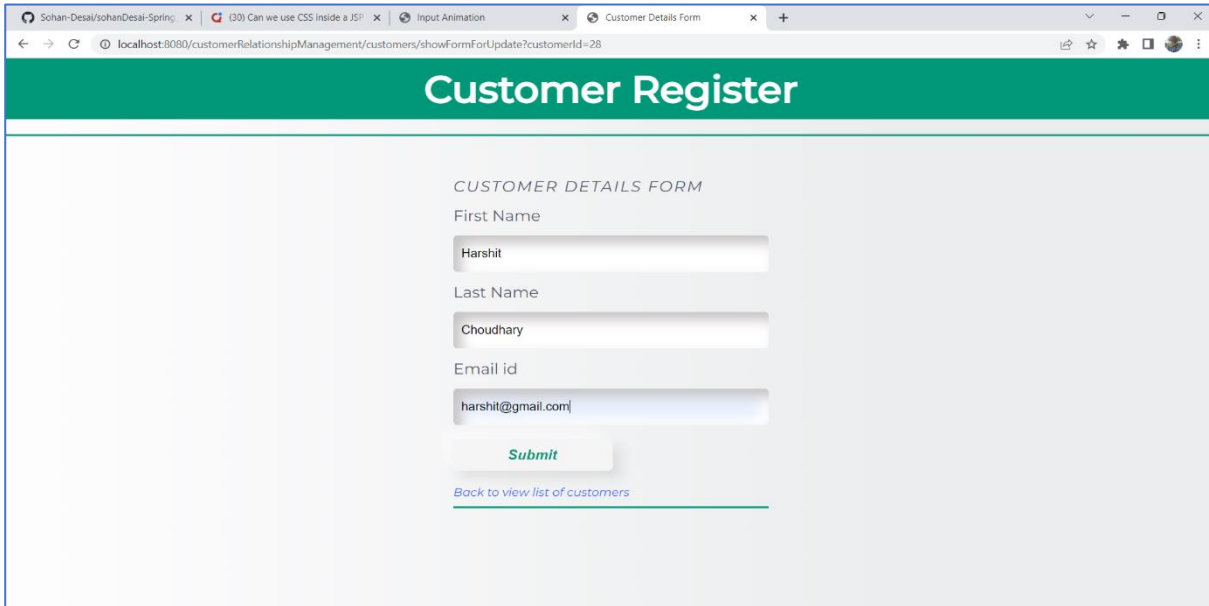
Update link hover



Delete link hover

On clicking the **Update** link, the “Customer details” form gets loaded, pre-populated with existing data.

User can make necessary changes and click on **Submit** button.

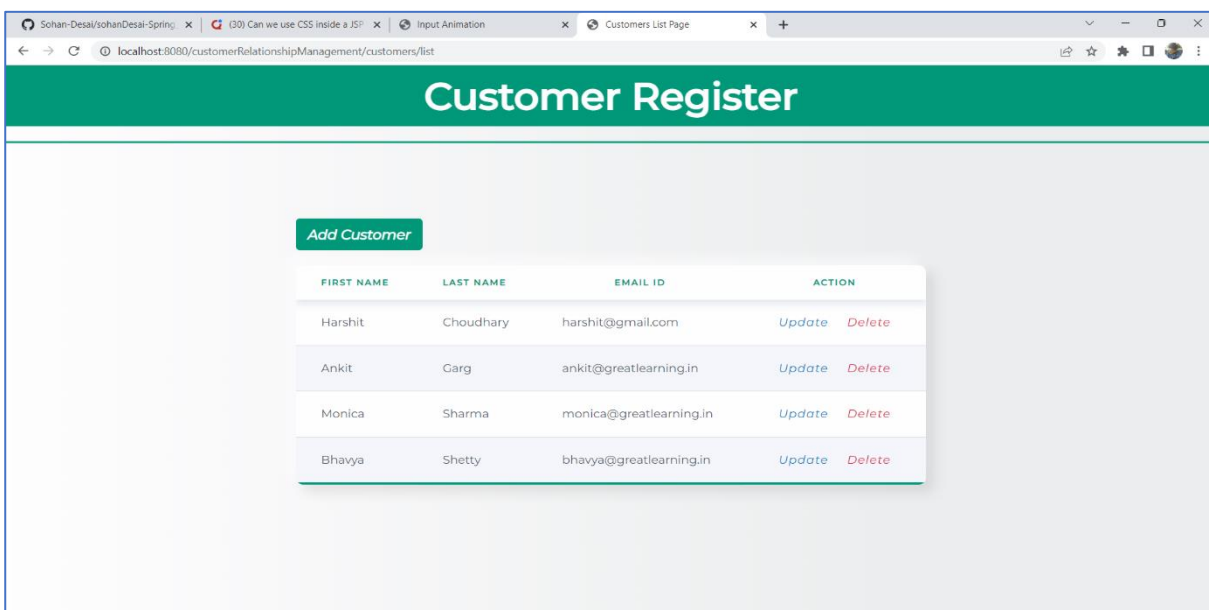


The screenshot shows a web browser window with the URL `localhost:8080/customerRelationshipManagement/customers/showFormForUpdate?customerId=28`. The page has a green header with the text "Customer Register". Below the header, the form is titled "CUSTOMER DETAILS FORM". It contains three input fields: "First Name" with the value "Harshit", "Last Name" with the value "Choudhary", and "Email id" with the value "harshit@gmail.com". Below the input fields is a green "Submit" button and a link that says "Back to view list of customers".

After clicking on **Submit** button. The updated record gets displayed on customer list page.

Eg: Email id harshit@greatlearning.in is updated to harshit@gmail.com

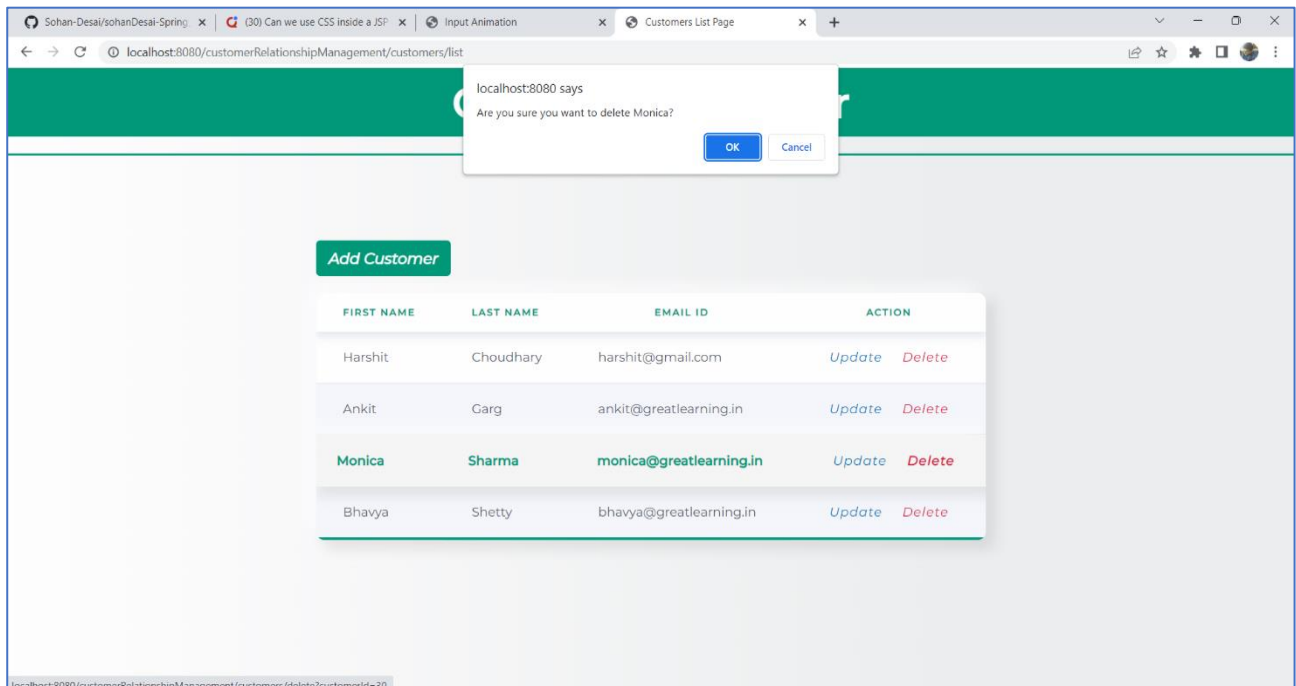
Here we see the record has been updated!



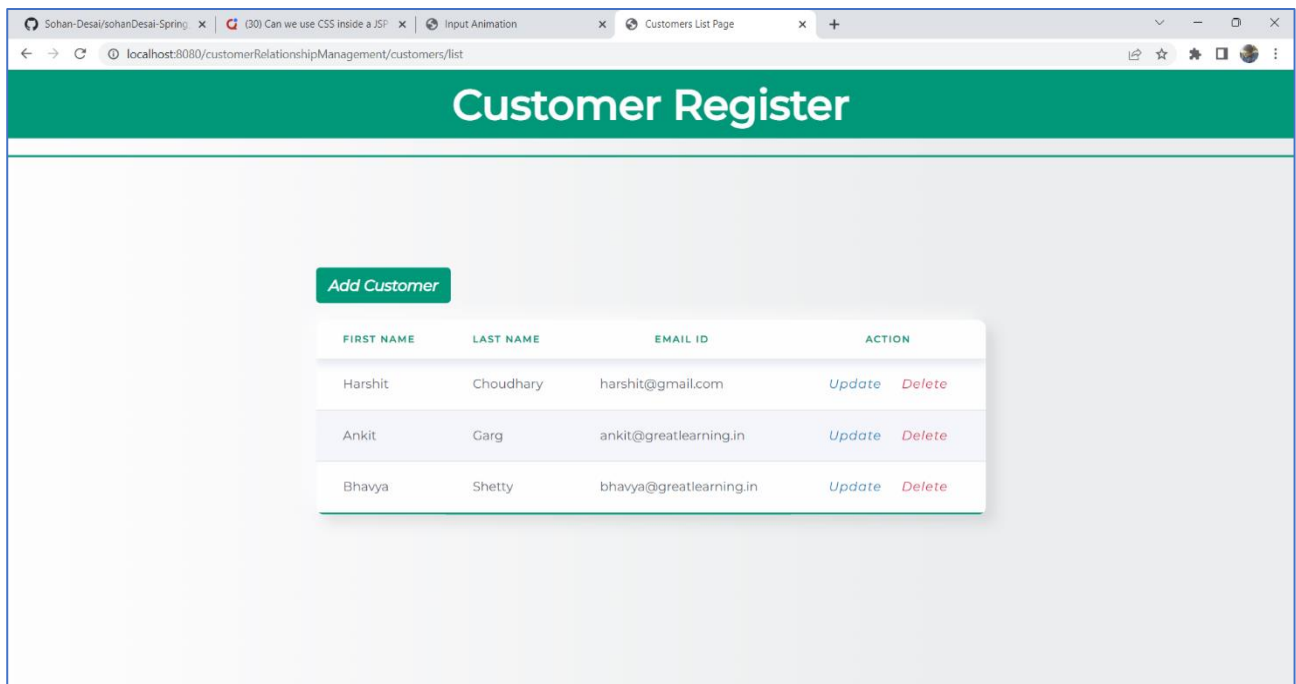
The screenshot shows a web browser window with the URL `localhost:8080/customerRelationshipManagement/customers/list`. The page has a green header with the text "Customer Register". Below the header, there is a green "Add Customer" button. Below the button is a table with the following data:

FIRST NAME	LAST NAME	EMAIL ID	ACTION
Harshit	Choudhary	harshit@gmail.com	Update Delete
Ankit	Garg	ankit@greatlearning.in	Update Delete
Monica	Sharma	monica@greatlearning.in	Update Delete
Bhavya	Shetty	bhavya@greatlearning.in	Update Delete

Finally, on clicking the **Delete** link, a message pops up on the page asking to confirm whether to delete the concerned record from database.



On confirmation, the concerned record gets deleted while remaining records are displayed on the page.



Monica's records are deleted

Thus, we have successfully checked all the functionalities of our Spring MVC project for Customer Relationship Management Portal.

Thank you!

#Note->

Please note that there is a minor jitter in the header row whenever the cursor hovers over last row of table. I however couldn't deduce the exact reason for that bug.

Also you will find a video recording of this project on my google drive. Link to which I've shared below.

https://drive.google.com/drive/folders/14Ji3uxexF9MAJ0v2_7xj6zHzQv73_2KH?usp=sharing

Thanks again!