

Assignment

JSF Web Application with MySQL and MongoDB Databases

Contents

Introduction.....	3
MySQL Database.....	3
Mongo Database	4
Marks	4
Plagiarism	4
Skeleton.....	5
Submission of the Project	5
Overview of Web App.....	5
Main Page	5
Manage Stores Page.....	6
Add Store Page	7
Show Products Details Page	8
Delete Store.....	9
Manage Products Page.....	10
Delete Product.....	10
Manage Head Offices	11
Add Head Office Page	12
Connection Problems.....	14

Introduction

Write a JSF dynamic web application that queries and updates a MySQL database and a MongoDB database.

The MySQL database is called *shops* and can be downloaded from the *Project* section of Moodle for this module (*shops.sql*).

MySQL Database

The *shops* database consists of 2 tables:

Store	
Column Name	Details
Id	A code representing the Store e.g. 1
Name	The Store Name e.g. "Tesco"
Founded	The date the store was founded e.g. "1919-07-12"

Product	
Column Name	Details
pid	A code representing the Product
sid	A code representing the Store the product is sold in
prodName	The name of the product
Price	The price of the product

Full details of the Primary and Foreign Keys as well as datatypes can be found in the database itself.

Mongo Database

The Mongo database can be downloaded from the *Project* section of Moodle for this module (*storeHeadOfficeDB.txt*) and should be imported into Mongo.

The Mongo database is called *storeHeadOfficeDB* and consists of a collection called *storeHeadOffice*, which initially has 3 documents as follows:

```
{ "_id" : 1, "location" : "Hertfordshire" }  
{ "_id" : 2, "location" : "Dublin" }  
{ "_id" : 3, "location" : "Dublin" }
```

Marks

This assignment is worth 50% of the marks for the module.

90% of the marks for this assignment will be for implementing the functionality in this document.

10% of the marks for this assignment will be for innovation – extra features etc. Any innovation must be clearly indicated in a document called *innovation.doc* stored in the root folder of your application.s

Plagiarism

Plagiarism will be dealt with in accordance with the institute's [Plagiarism policy](#).

Skeleton

A skeleton for this project (named GXXXXXXXXX.7z) should be downloaded from the *Project* section of Moodle and used as a basis for this project.

Submission of the Project

The zipped Eclipse project (named GXXXXXXXXX.zip where GXXXXXXXXX is your student number) should be uploaded to the *Project* section of Moodle no later than **Sunday December 15th 2019 at 11:00pm.**

Overview of Web App

The web app should consist of the following pages:

Main Page

The *Main* page consists of 3 links:

- One to the *Manage Stores* page
- One to the *Manage Products* page
- One to the *Manage Head offices* page

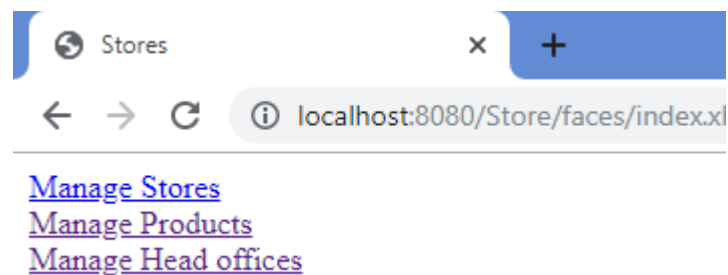


Figure 1 Main Page

Manage Stores Page

The *Manage Stores* page:

- Shows details of all Stores
- Has an *Add Store* button
- Has *Show Products* and *Delete* actions for each Store
- Has a link back to the *Home* page.

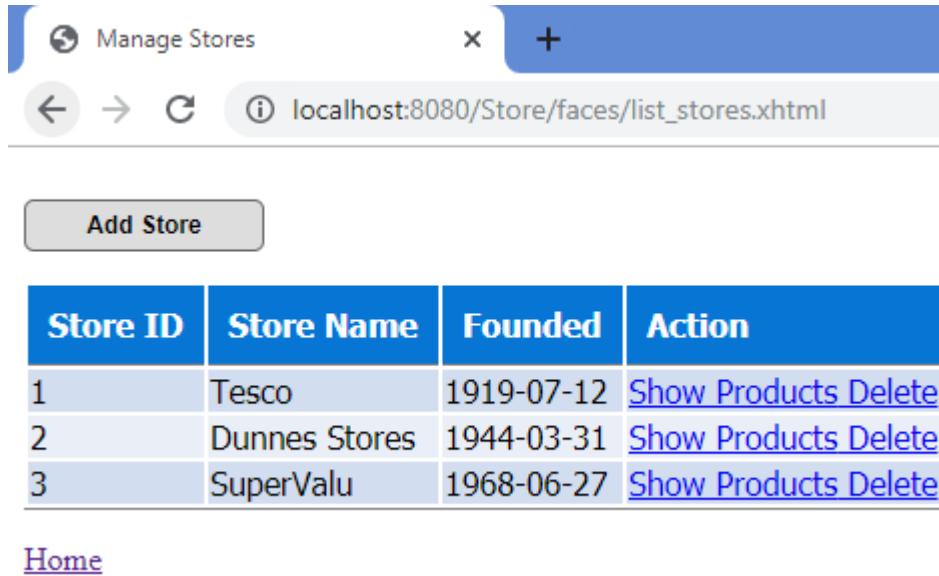
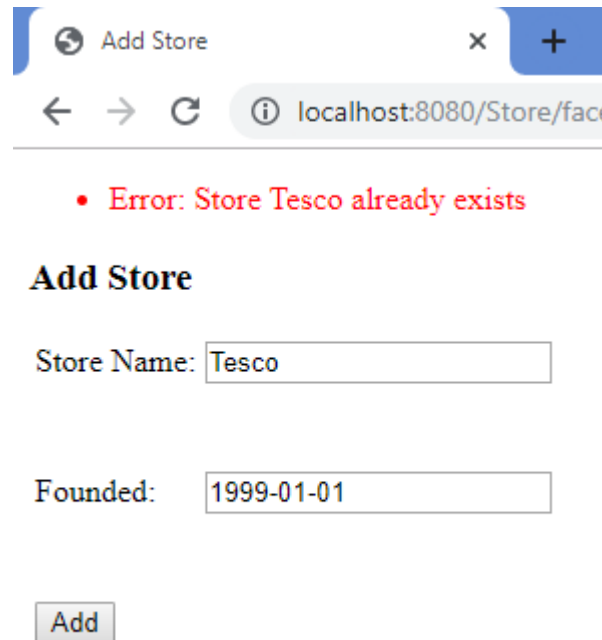


Figure 2 Manage Stores Page

Add Store Page

The *Add Store Page* allows the user to enter a *Store Name* and *Founded* date. If an existing *Store Name* is entered an error should be displayed as shown. (You can assume the date entered will always be in the correct format).

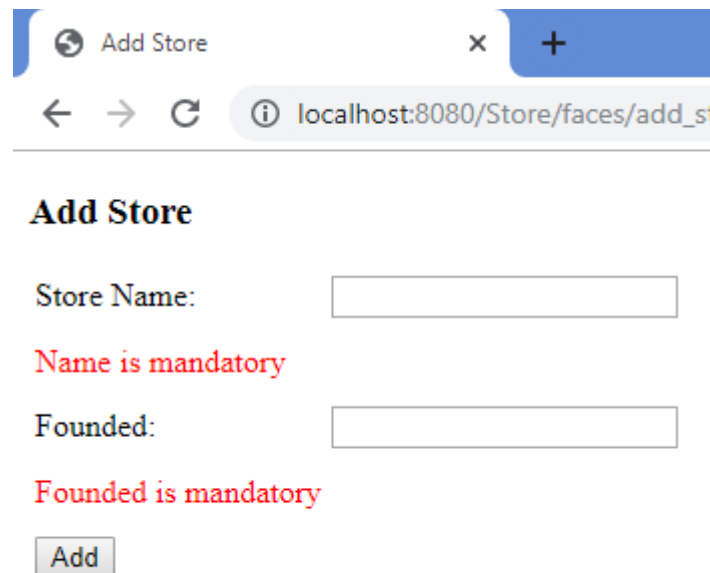


The screenshot shows a web browser window with the title 'Add Store'. The address bar displays 'localhost:8080/Store/faces'. Below the browser window, a red error message is shown: 'Error: Store Tesco already exists'. The main content area is titled 'Add Store' and contains two input fields: 'Store Name' with the value 'Tesco' and 'Founded' with the value '1999-01-01'. At the bottom of the form is an 'Add' button.

[Home Stores](#)

Figure 3 Store Name already exists

If *Store Name* or *Founded* are not specified appropriate error messages are shown:



The screenshot shows the 'Add Store' page with empty input fields for 'Store Name' and 'Founded'. Below the 'Store Name' field, a red error message is displayed: 'Name is mandatory'. Below the 'Founded' field, another red error message is displayed: 'Founded is mandatory'. An 'Add' button is located at the bottom of the form.

[Home Stores](#)

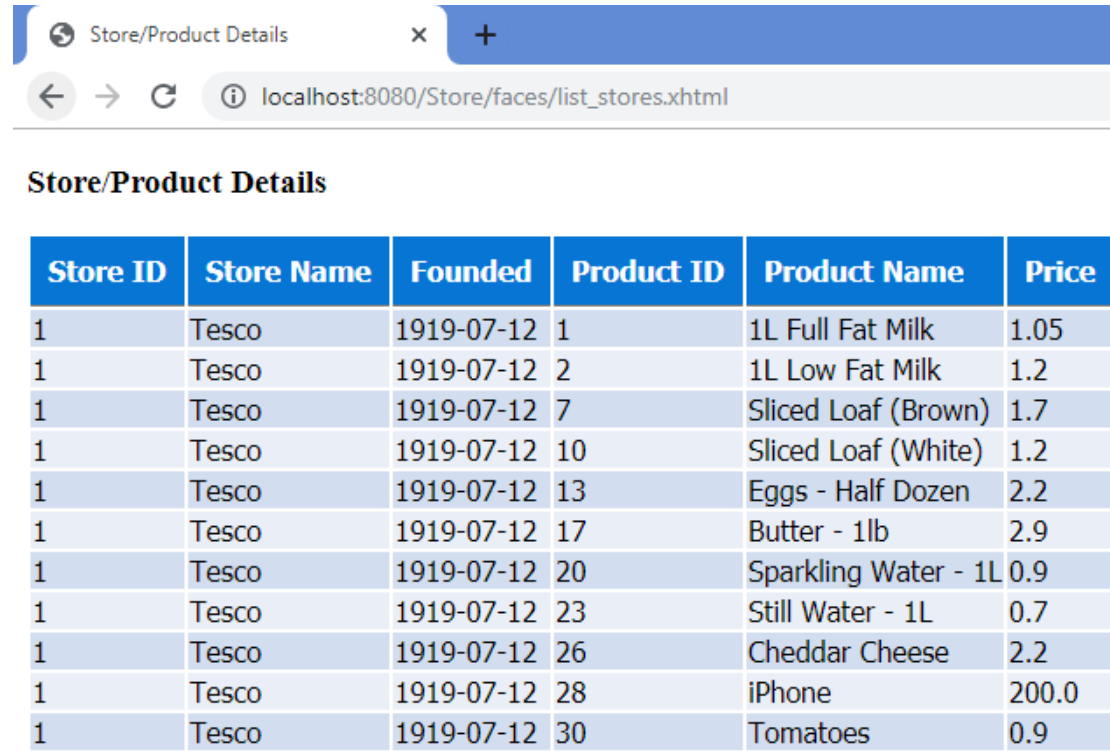
Figure 4 Data not entered

When a Store has been added successfully, the user is returned to the *Manage Stores Page*.

Show Products Details Page

When the *Show Products* action is pressed on the *Manage Stores* page, the user is brought to the *Store/Product Details* page.

This page shows details of all Stores and the Products in those stores as shown.



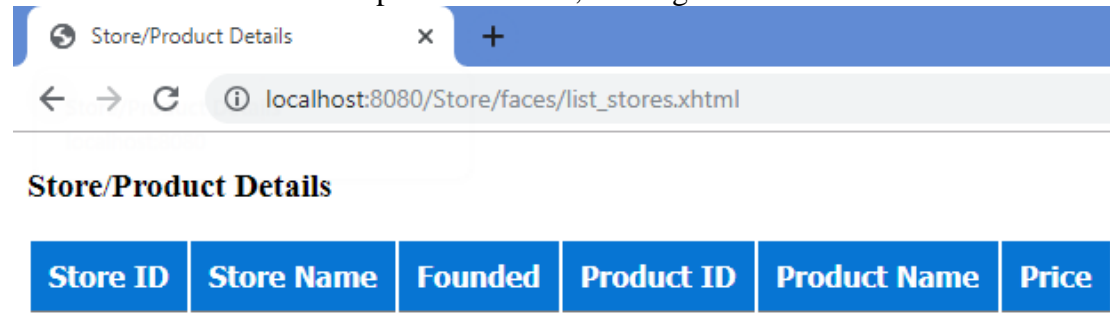
The screenshot shows a web browser window with the title 'Store/Product Details'. The address bar displays 'localhost:8080/Store/faces/list_stores.xhtml'. Below the browser window, the page title 'Store/Product Details' is followed by a table with 6 columns: Store ID, Store Name, Founded, Product ID, Product Name, and Price. The table contains 11 rows of data, all for the store 'Tesco'.

Store ID	Store Name	Founded	Product ID	Product Name	Price
1	Tesco	1919-07-12	1	1L Full Fat Milk	1.05
1	Tesco	1919-07-12	2	1L Low Fat Milk	1.2
1	Tesco	1919-07-12	7	Sliced Loaf (Brown)	1.7
1	Tesco	1919-07-12	10	Sliced Loaf (White)	1.2
1	Tesco	1919-07-12	13	Eggs - Half Dozen	2.2
1	Tesco	1919-07-12	17	Butter - 1lb	2.9
1	Tesco	1919-07-12	20	Sparkling Water - 1L	0.9
1	Tesco	1919-07-12	23	Still Water - 1L	0.7
1	Tesco	1919-07-12	26	Cheddar Cheese	2.2
1	Tesco	1919-07-12	28	iPhone	200.0
1	Tesco	1919-07-12	30	Tomatoes	0.9

[Home Stores](#)

Figure 5 All Products in the Tesco store

If there are no Products in a particular Store, nothing should be show:



The screenshot shows a web browser window with the title 'Store/Product Details'. The address bar displays 'localhost:8080/Store/faces/list_stores.xhtml'. Below the browser window, the page title 'Store/Product Details' is followed by a table with 6 columns: Store ID, Store Name, Founded, Product ID, Product Name, and Price. The table is empty.

Store ID	Store Name	Founded	Product ID	Product Name	Price
----------	------------	---------	------------	--------------	-------

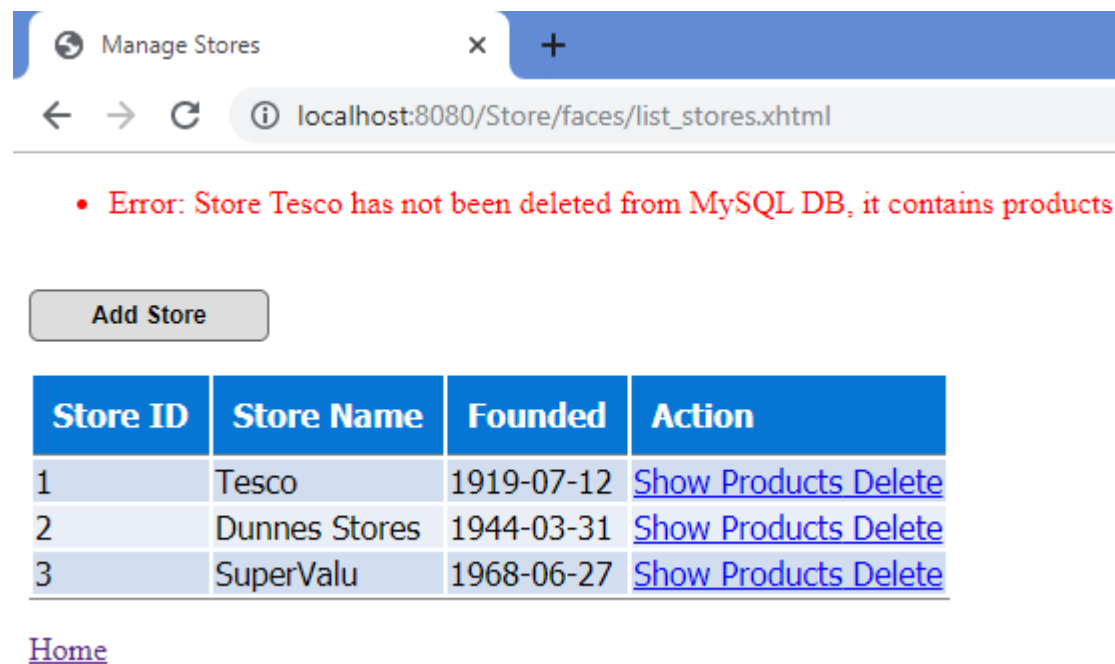
[Home Stores](#)

Figure 6 There are no products in this store

Delete Store

When the *Delete* action is pressed on the *Manage Stores* page, if the particular Store to be deleted has no Products, it is deleted from the database and the user remains on the *Manage Stores* page which is updated to show the new list of courses.

If the particular Store to be deleted has Products, an error message is shown and the Store is not deleted from the database.



The screenshot shows a web browser window with the title 'Manage Stores'. The address bar displays 'localhost:8080/Store/faces/list_stores.xhtml'. Below the browser window, a red error message is displayed: 'Error: Store Tesco has not been deleted from MySQL DB, it contains products'. Below the error message is a button labeled 'Add Store'. Below the button is a table with four columns: 'Store ID', 'Store Name', 'Founded', and 'Action'. The table contains three rows of data. The first row is for 'Tesco' (Store ID 1, Founded 1919-07-12) with links for 'Show Products' and 'Delete'. The second row is for 'Dunnes Stores' (Store ID 2, Founded 1944-03-31) with links for 'Show Products' and 'Delete'. The third row is for 'SuperValu' (Store ID 3, Founded 1968-06-27) with links for 'Show Products' and 'Delete'. Below the table is a link labeled 'Home'.

• Error: Store Tesco has not been deleted from MySQL DB, it contains products

Add Store

Store ID	Store Name	Founded	Action
1	Tesco	1919-07-12	Show Products Delete
2	Dunnes Stores	1944-03-31	Show Products Delete
3	SuperValu	1968-06-27	Show Products Delete

[Home](#)

Figure 7 Cannot delete Stores that have Products

Manage Products Page

The *Manage Products* page:

- Shows details of all Products
- Has a *Delete* action for each Product
- Has a link back to the *Home* page.



Store ID	Product ID	Name	Price	Action
1	1	1L Full Fat Milk	1.05	Delete
1	2	1L Low Fat Milk	1.2	Delete
2	3	1L Full Fat Milk	1.05	Delete
2	4	1L Low Fat Milk	1.1	Delete
3	5	1L Full Fat Milk	1.15	Delete
3	6	1L Low Fat Milk	1.2	Delete
1	7	Sliced Loaf (Brown)	1.7	Delete
2	8	Sliced Loaf (Brown)	1.65	Delete
2	9	Sliced Loaf (Brown)	1.7	Delete
1	10	Sliced Loaf (White)	1.2	Delete
2	11	Sliced Loaf (White)	1.2	Delete
3	12	Sliced Loaf (White)	1.4	Delete
1	13	Eggs - Half Dozen	2.2	Delete
2	14	Eggs - Half Dozen	2.1	Delete
3	15	Granola	4.1	Delete
2	16	Granola	3.5	Delete
1	17	Butter - 1lb	2.9	Delete
2	18	Butter - 1lb	2.9	Delete

Figure 8 Manage Products Page

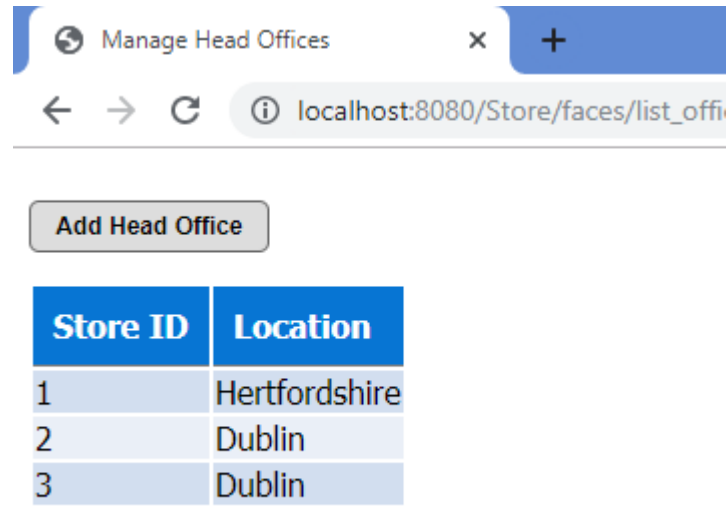
Delete Product

When the *Delete* action is pressed on the *Manage Products* page, the product is deleted from the database and the user remains on the *Manage Products* page which is updated to show the new list of products.

Manage Head Offices

The *Manage Head Offices* page:

- Shows the Store ID and Location of the Store's head office – from the *storeHeadOffice* collection in the *storeHeadOfficeDB* database in MongoDB
- Has an *Add Head Office* button
- Has a link back to the *Home* page.



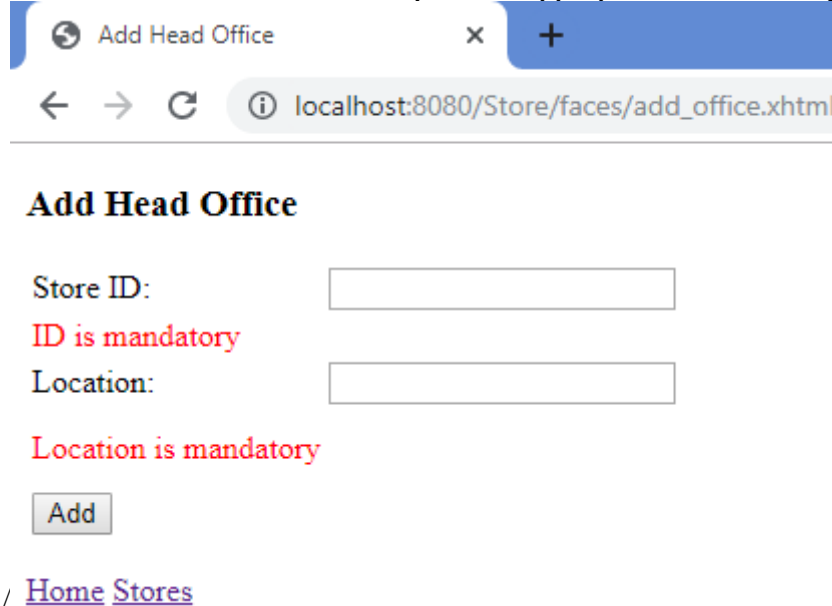
[Home](#)

Figure 9 List Head Offices

Add Head Office Page

The *Add Head Office Page* allows the user to enter a *Store ID* and *Location*.

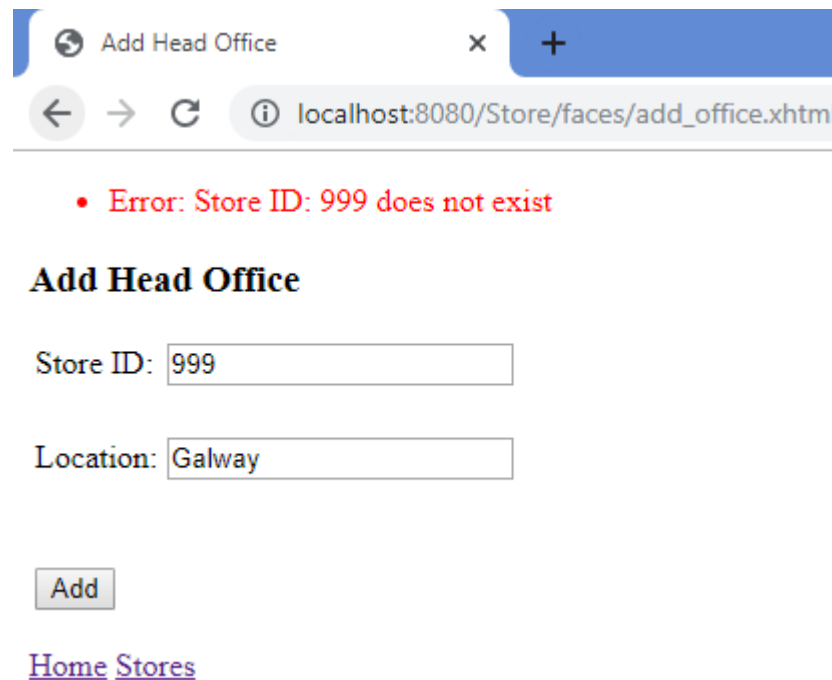
If *Store ID* or *Location* are not specified appropriate error messages are shown:



The screenshot shows a web browser window with the title 'Add Head Office'. The address bar displays 'localhost:8080/Store/faces/add_office.xhtml'. The page content includes the heading 'Add Head Office' and two input fields. The first field is labeled 'Store ID:' and has a red error message 'ID is mandatory' below it. The second field is labeled 'Location:' and has a red error message 'Location is mandatory' below it. An 'Add' button is located at the bottom of the form. A link labeled 'Home Stores' is visible at the bottom of the page.

Figure 10 Add Head Office - Mandatory fields

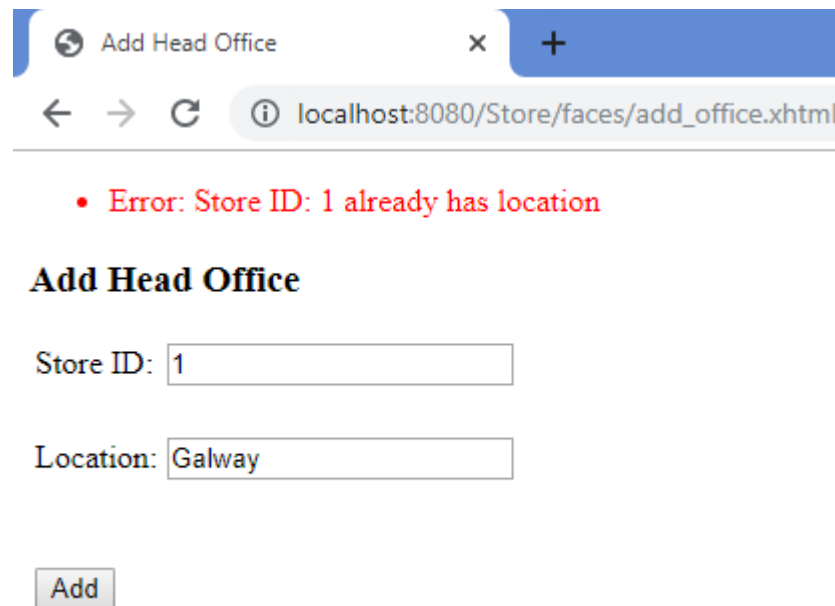
If a non-existent *Store ID* is entered, i.e. one that does not exist in the MySQL database, an appropriate error message is shown:



The screenshot shows the same web browser window as Figure 10. The 'Store ID' field now contains the value '999'. A red error message 'Error: Store ID: 999 does not exist' is displayed above the 'Location' field. The 'Location' field contains the value 'Galway'. The 'Add' button and the 'Home Stores' link are still present.

Figure 11 Add Head Office - No such Store

If a *Store ID* is specified, i.e. a document with that `_id` exists in the MongoDB, an appropriate error message is shown:



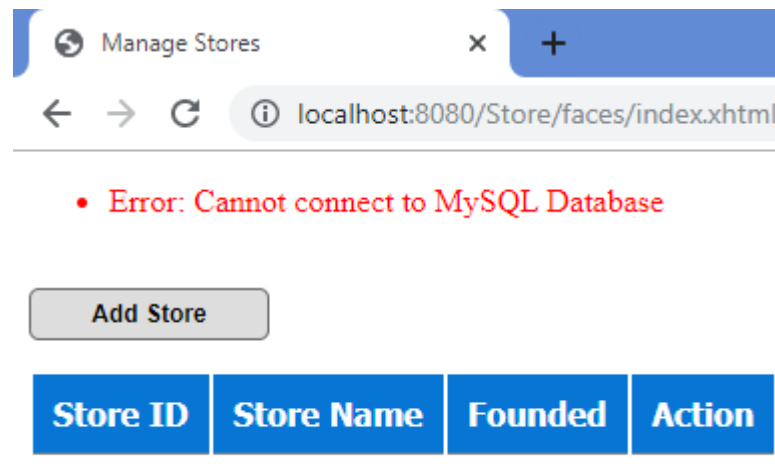
The screenshot shows a web browser window with a single tab titled "Add Head Office". The address bar displays "localhost:8080/Store/faces/add_office.xhtml". Below the address bar, a red error message is shown: "• Error: Store ID: 1 already has location". The main content area is titled "Add Head Office" and contains two input fields: "Store ID:" with the value "1" and "Location:" with the value "Galway". Below these fields is a button labeled "Add".

[Home Stores](#)

Figure 12 Add Head Office - Head Office already exists

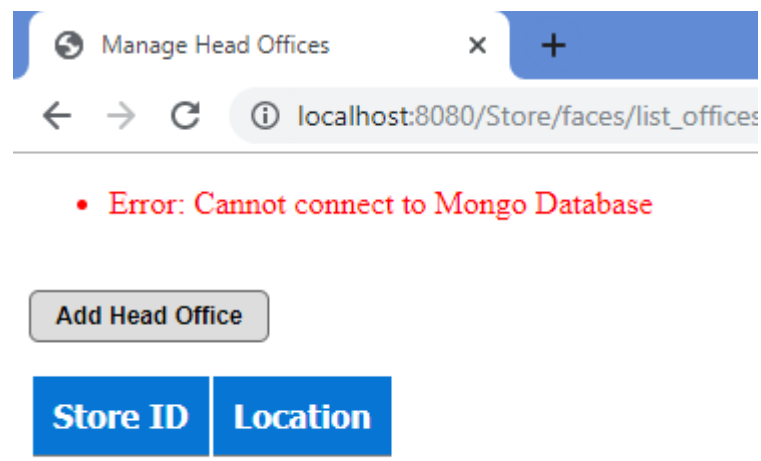
Connection Problems

If the MySQL or Mongo databases are uncontactable at any time, an appropriate error message must be shown:



[Home](#)

Figure 13 MySQL database uncontactable while trying to Manage Stores



[Home](#)

Figure 14 Mongo database uncontactable while trying to Manage Head Offices