

MP1 READ ME

Aim: To explore JSP/JSTL presentation layer technologies, JDBC database interaction, and to implement the Model - View - Controller (MVC) pattern.

JDBC interaction will be C(reate), R(read), U(pdate) and D(elete)

Project Summary:

1. JSP/JSTL

- The main purpose of the first part of MP1 was to explore the JSP/JSTL technology along with JDBC interaction
- The given World schema consisted of three databases:
 - Country
 - City
 - CountryLanguage
- This projects enables the user to do the basic CRUD operations on these databases
- User can access every database and navigate through different JSPs in order to view all the records, update/delete certain record or create a new record

2. MVC

- The main purpose of the second part of MP1 was to implement the MVC framework
- The Model-View-Controller is basically separation of the three main layers of operation:
 - Data Presentation Layer: views
 - Business Logic Layer: controllers
 - Data Access Layer: models
- The views will contain all the JSPs having the HTML/CSS content, that would be seen by the user
- The models can access the database using JDBC objects and SQL
- The controllers are used to drive the operations based on user's input. It contains the logic that renders the specific view to user containing relevant data fetched from the database
- The aim was to let user perform the basic CRUD operations on the mentioned three databases
- The user can access every database and navigate through various JSPs in order to create, view, update or delete any particular record

Design & Development Insights:

1. JSP/JSTL

- In order to explore JSP/JSTL technologies I created all the relevant JSPs under the Web Pages Folder
- There are 4 JSPs corresponding to Create, Show, Update and Delete operations for every database
- There are 3 JSPs (inclusive of index.jsp) for navigation purpose
- Every JSP performs a particular operation and passes the control to other JSP as a result of post form action
- JSTL such as core (for comparison or other logic implementation), sql (for SQL operations) and fmt (Formatter for display) are used in every JSP
- When this project is run, the user is presented with a index.jsp which has two options to choose from , either show all or create new record
- If the user selects show all option, he would be asked to select the database he wants to work on
- Then he would be navigated to the specific database showAll JSP
- He can update and delete the records
- If the user selects update option for say first record, he would be navigated to a update form JSP where he will get to see all the data entries for that record and he is allowed to change any of that data and submit the form in order to reflect the changes in the database
- He also has an option to go back to the Home page or Show All page if no update is needed
- If the user wants to delete a record, he can do so by clicking the delete link against the particular record
- This will navigate him to a delete JSP where he can see which record did he delete as the record ID would be displayed and then the user can go to the HOME page
- The user can click on new record from the Home page which will navigate him to database selection page after which he can view the create form JSP and enter new data values and save the changes by hitting submit
- This will navigate him back to the show all JSP

2. MVC

- Segregation of the code is done by creating different packages such as controllers, models or services
- All the view files (JSPs) are kept in the Web Pages folder. It also contains the CSS and other extra files (images or index.jsp)
- The controllers package contains the servlets for each of the CRUD

operations for each database

- The models package contains the java classes having database properties analogous to the columns in the tables in order to create relevant objects and store their property values
- The service package contains the data access operations in order to create, update, read or delete records from the three databases
- The connection is established using the JDBC connection pool
- All the SQL queries are written in this java file
- Whenever the user runs this project, the index page will be displayed
- It shows various options to show records or create new records for all the three databases viz. Country, City and Country Language
- If the user selects to show records for any of the tables, he would be navigated to the JSP which will display all the records from the chosen database
- Every displayed record will have an option of update or delete at the end of each row
- If the user selects update option for say first record, he would be navigated to a update form JSP where he will get to see all the data entries for that record and he is allowed to change any of that data and submit the form in order to reflect the changes in the database
- He also has an option to go back to the Home page if no update is needed
- If the user wants to delete a record, he can do so by clicking the delete link against the particular record
- This will navigate him to a delete JSP where he can see which record did he delete as the record ID would be displayed and then the user can go to the HOME page
- The user can click on new record from the Home page which will navigate him to the create form JSP and he can enter new data values and save the changes by hitting submit
- This will navigate him back to the show all JSP

Requirements:

1. OS: Mac OS 10.8
2. Installation: MySQL WorkBench, NetBeans and JDK 7 for macosx
3. Connection Pools: tgajareMp1JspPool **or** tgajareMp1MvcPool
4. Connection Resources: jdbc/tgajareMp1Jsp **or** jdbc/tgajareMp1Mvc
5. Web Contexts: tgajaremp1jsp **or** tgajaremp1mvc
6. MySQL user: itmd4515 with password as itmd4515

Screen Captures and Expected Results:

1. JSP/JSTL

2. MVC