MVC/DB Coffee Shop Locator

In this assignment you are to create a coffee shop locator application using a `Model-View-Controller` approach backed by a `MySQL` database.

Database

You will be using a new data source in this assignment. Your data source should be parsed and inserted into your MySQL database. You can write a custom Java parser to do this, or simply use phpMyAdmin. Regardless of your method, you should have a table named `starbucks` consisting of all data present in the data source provided.

Search Controller

You are to implement a `Search` controller that is mapped to the following URL:

http://cs3.calstatela.edu:8080/cs320stuXX/MVC/Search

As usual, substitute _XX with your CS3 account number._

- Your controller is responsible for querying your `MySQL` database for `Coffee Shops` that match your User's search criteria.
- Once you data has been queried, you should forward it to a `Search View` JSP page.
- Your controller will support the 3 types of searches listed below.

```
NOTE: Your Search Controller should support both POST and GET
requests.
```

Search by Zip Code

- When searching by `zip code`, your application should display all Starbucks locations that exist in the zip code provided.
    - Note: *Partial* zip code matches are **OK**.

- Your application should expect the following **parameters** when searching by zip code:
    - `type=zip`
        - The `type` parameter should be set to the literal value, `zip`.

    - `zip=<submitted value>`
        - The `zip` parameter should be assigned the zip code (or partial) being searched. That is to say, `<submitted value>` should be replaced with the actual search string submitted by the User.

Search by City Name

- When searching by `City Name`, your application should display all Starbucks locations that exist in the city specified.
    - Note: *Partial* city names, and names that *sound* similar to the actual city can be considered matches when searching.

- Your application should expect the following **parameters** when searching by city name:
    - `type=city`
        - The `type` parameter should be set to the literal value, `city`.

    - `city=<submitted value>`
        - The `city` parameter should be assigned the full (or partial) city name being searched. That is to say, `<submitted value>` should be replaced with the actual search string submitted by the User.

Search by Latitude, Longitude, and Radius

- When searching by `latitude`, `longitude`, and `radius`, your application should display all Starbucks locations that are within `radius` miles of the `latitude` and `longitude` specified.
- Your application should expect the following **parameters** when searching by location:
    - `type=location`
        - The `type` parameter should be set to the literal value, `location`.

    - `lat=<submitted value>`
        - The `lat` parameter should be assigned the latitude submitted by the

User.

- `lon=<submitted value>`
  - The `lon` parameter should be assigned the longitude submitted by the User.

- `radius=<submitted value>`
  - The `radius` parameter should be assigned the radius submitted by the User.
    -

Search View

Your `Search.jsp` View should be located at:

`/WEB-INF/hw5/Search.jsp`

- You should display the **Search Form(s)**, along with any and all **Search Results**.
- Search results should be displayed as an `Unordered List` with an `ID` **results-list**.
- Each search result should be a `List Item` with a `class` **result-item**.

Grading

- Database
  - 10 PTS - Upload a `starbucks.sql` file containing your table structure(s) and data.

- MVC Architecture
  - Search Controller
    - 20PTS - Query MySQL Database
    - Search by Zip Code
      - 20PTS - Partial Match
      - 20PTS - Full Match

    - Search by City Name

        - 20PTS - Partial Match
        - 20PTS - Phonetic Match

    - Search by Location
        - 20PTS - Perform distance calculation in MySQL Query

- Search View
    - 20PTS - Unordered List of Search Results
    - 20PTS - List Items for each Result Item

Total: 170PTS