

## Readme

---

### ❖ Lines of Code

- **Java Files:**
  - Number of Files: - 25
  - Line of code: - 3156
- **JSP Files:**
  - Number of Files: - 18
  - Line of code: 3521
- **Js Files:**
  - Number of Files: - 4
  - Line of code: - 528
- **CSS Files:**
  - Number of Files: - 2
  - Line of code: - 770

**Total Line of code: - 7975**

**Note:** - We used js files to do ajax call, auto complete API.

### ❖ Deployment Steps: -

#### ➤ Needed Jar files

- mysql-connector-java-5.1.44-bin.jar (for JDBC driver that connect java to MySQL server)
- mongo-java-driver-3.2.2.jar (to connect mongo DB database from java)
- servlet-api.jar (for java servlets)

Put all these required jar files into the class path folder what is set into the environment variable.

➤ **Compilation**

- I have already compiled the project, so you can find all the compiled .class file in different packaged folders.
  - **CSP584\_Project\_Pandya\_Rushit/Source Code\Pocket\WEB-INF\classes**
- I implemented MVC architecture, so you can find all the view related classes in to view folder, beans into beans folder and controllers into controller's folder etc.

➤ **Setup and Run the Project:**

1. Copy the below source code folder and put it into the webapps directory of Tomcat.
  - a. **CSP584\_Project\_Pandya\_Rushit/Source Code/Pocket**
2. Start the Tomcat server by running startup.sh file.
3. Start the MySQL server.
4. Now to load database into MySQL open the MySQL workbench and import the .sql file present on below path.
  - a. **CSP584\_Project\_Pandya\_Rushit/Create\_DB.sql**
5. Start the mongo DB server
6. Open the mongo DB client and create the database and collection from below .txt file.
  - a. **CSP584\_Project\_Pandya\_Rushit/mongoDBDatabase.txt**
7. Open the browser and type the URL "**localhost:8080/Pocket**" to run the web application.