A Project Report on

Identity Management

By
Shivakumar PUTTASWAMY

Table of Contents

1. Subject description

Identity management describes the management of individual identities, their authentication, authorization, roles and privileges within or across system and enterprise boundaries with the goal of increasing security and productivity while decreasing cost, downtime, and repetitive tasks.

2. Subject analysis

2.1. Major features

The IAM project will provide the following features:

- Authenticate existing users.
- Create new identity.
- Search an identity.
- Update an identity.
- Delete an identity

2.2. Application Feasibility

- IAM project is a prototype of an identity management system.
- Web based application which provides simple and user friendly interface.
- GUI is designed and developed by using open source software's.
- Cost of building this application is very less.
- Performance is very high.

2.3. Technology Used

The main components used for the creation of project are as follows:

- Core Java
- Maven
- Derby database
- Spring and Hibernate Framework
- Servlet
- Java Server Pages

2.4. Data description

- The data which the project creates and uses are the identities.
- There are 2 types of identities or Data Models:
  - Normal Identities ○ This Data Model provides structure for DAO.
    - Consists of the following metadata
      USERNAME
      PASSWORD

2.5. Expected results

- User friendly application/tool created for Identity management system.
  - The application is capable of following:-
    - Authenticating the user.
    - Creating a new identity,
    - Updating an existing identity
    - Deleting an identity.
- Derby database is created and used to store data.
- The application communicates with the database and returns with the results in quick time.

2.6. Algorithms study

Algorithms study needs to be done for following two parts of IAM Application:-

- One for Front End i.e. Graphical user interface.
  - Study of Web application in java.
  - Study of Windows Builder in java.
- Other for Back end i.e. Database.
  - Study of storing data using files.
  - Study of Hibernate with JDBC functionality of Java

2.7. Scope of the application

2.7.1. Scope:

- Create an simple IAM Web based application.
  The IAM project will provide the following features:
  - Authenticate existing users.
  - Create new identity.
  - Search identity.
  - Update identity
  - Delete Identity

2.7.2. Limitations:

- Identities have very less amount of information or less attributes stored in the data base.
- The password stored is not encrypted in the database.
- Users cannot change their password or username.

2.7.3. Evolution:

- Data of identities stored is limited to 7 (including username and password) fields. Attributes of identities can be increased.

3. Conception

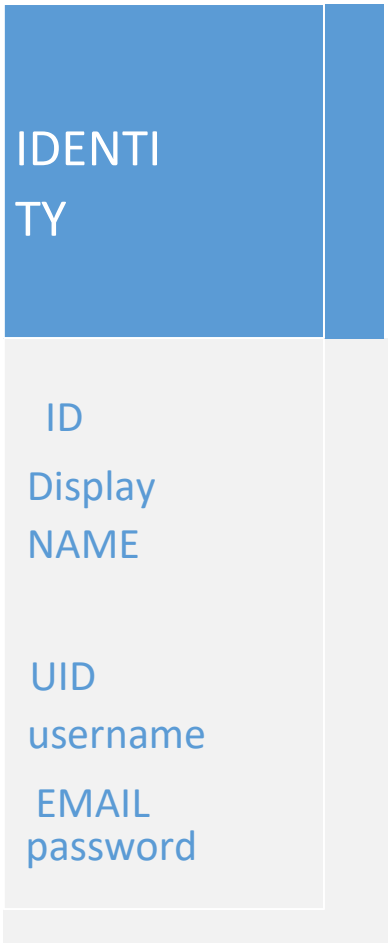
3.1. Chosen algorithm

- Graphical User Interface or Front End :-
- **JSP** - **Java server pages** is chosen for GUI because of following reasons:
    - Free and easily available.
    - Easy to use and create simple user friendly web pages.
- **Hibernate for Back End or Data Storage :-**

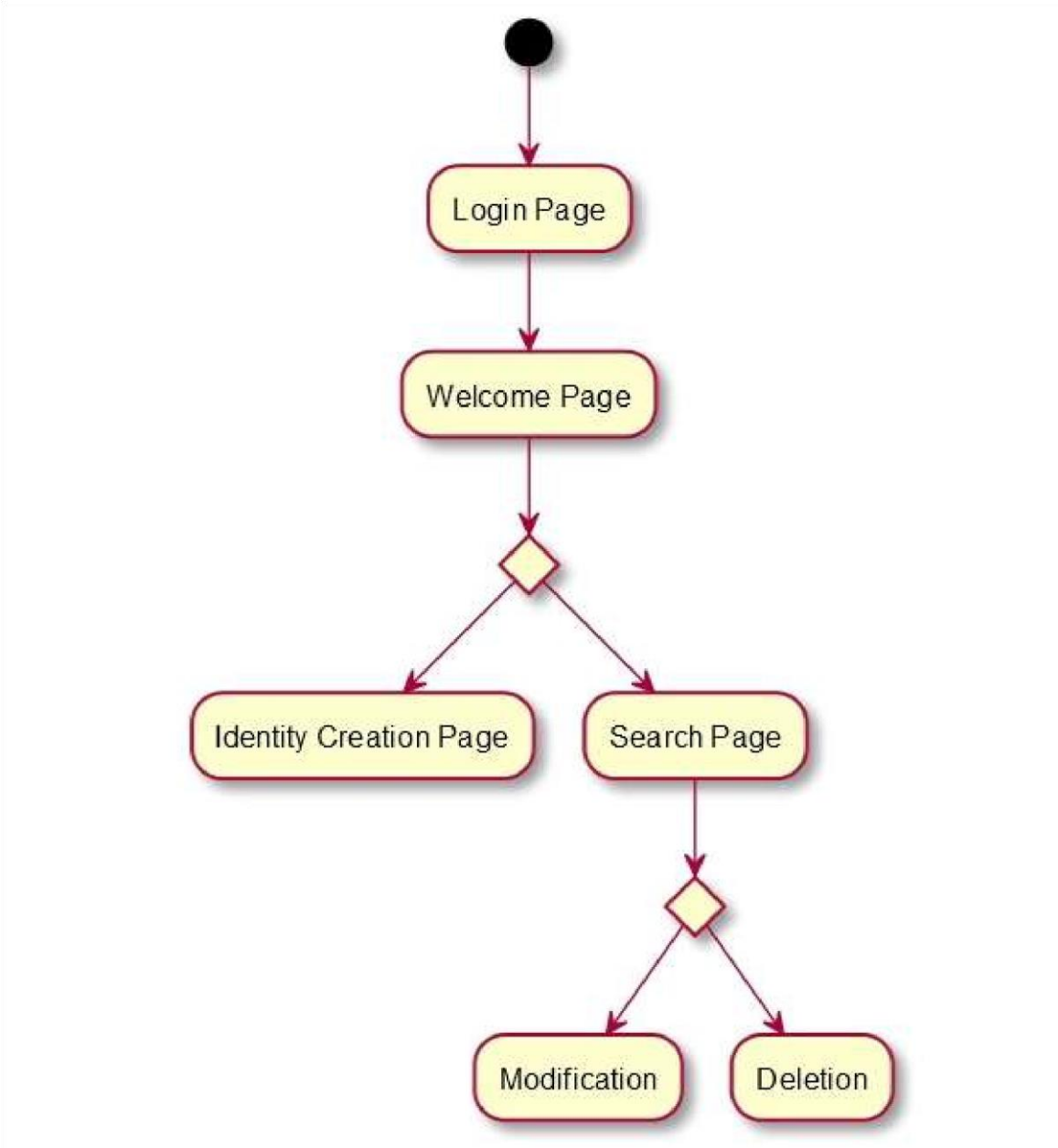
  Hibernate with Derby database is chosen to store IAM application related data because of following reasons:
    - It is Free.
    - Easy to use.
    - Easy to configure.
    - Scalable, faster than file system.

3.2. Data structures

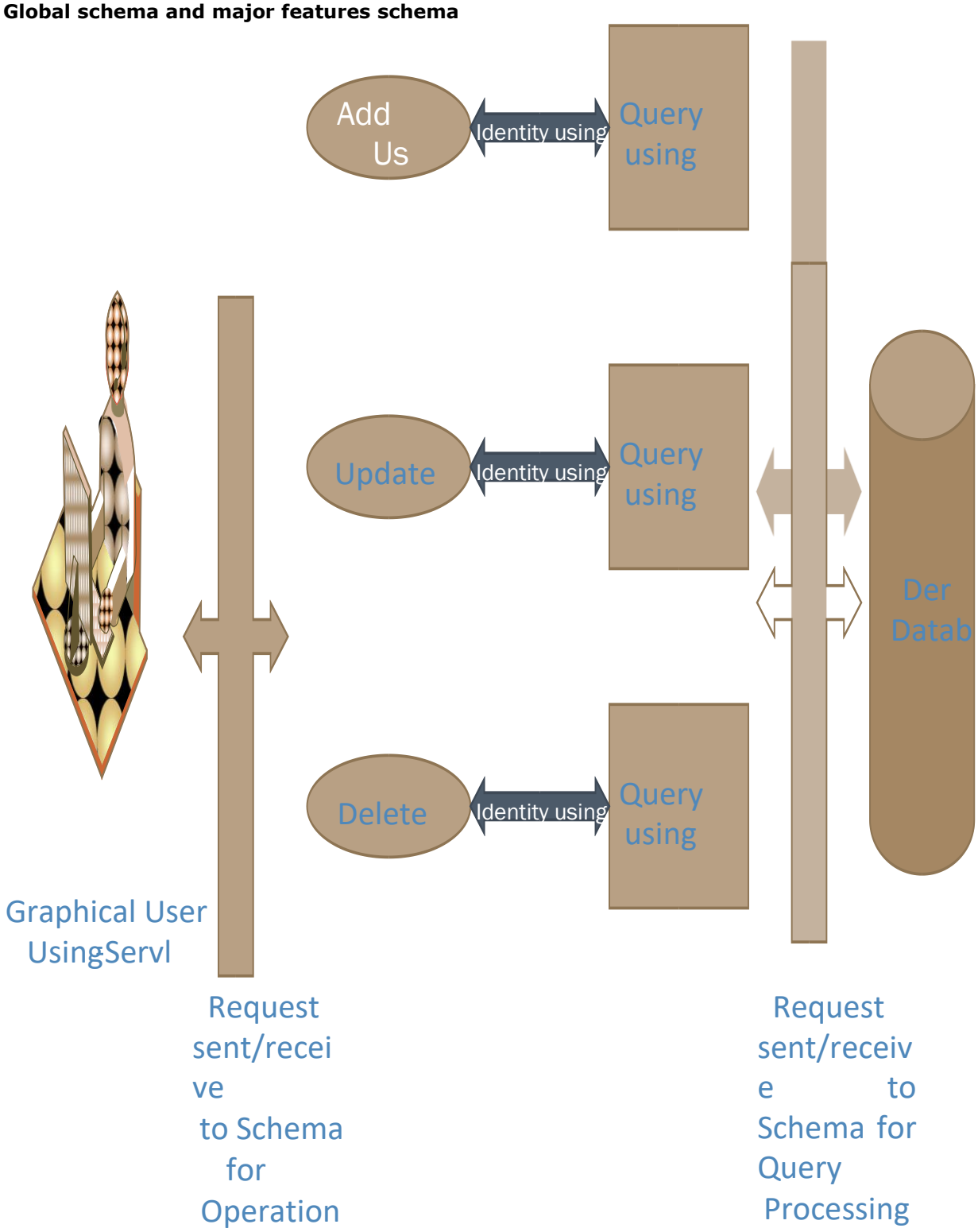
Class Diagram



3.3. Global application flow



3.4. Global schema and major features schema



4. Console operations description

4.1. Authenticate Users

- Users are authenticated on the **Login page** (login.jsp).
- User can use username **test** and password **test** any time to login into the system. Once the **user** is created from **ceate page**, he can use the new user name and password.
- Users can use login name and password to enter into the system.
- Users are redirected to the **Index page** (index.jsp).

4.1.1. Technical Description:

DAO Method: **(**Spring and hibernate framework is used **)**

The method void `readAllUsers` (Identity user) of interface `IdentityDAOInterface` is used to check if user exist or not.

4.2. Create Identity & Users of IAM system

- Users can go to the Create Identity page (**create.jsp**) to create an identity by clicking on **create identity** button on **Index page**.
- User enters the required information such as **displayname, uid, email,birthdate,username and password** to create an identity and click the **save identity** button to save the data into the database.

4.2.1. Technical Description:

DAO Method: **(**Spring and hibernate framework is used **)**

- The method save(Identity identity) of interface **IdentityDAOInterface** is used to create identity.

4.3. Search Identity (Spring and hibernate framework is used **)**

- Users can go to the **Search Identity** web page (search.jsp) to search an identity by clicking on **search identity** link on **Index page.**
- User can search specific identity by giving **display name and email** of identity using **search** button.

4.3.1. Technical Description:

DAO Methods (Spring and hibernate framework is used)

- The method search(Identity identity) of interface **IdentityDAOInterface** is used to search an identity.

4.4. Update Identity

- Users can go to the **Update Identity page** (**update**.jsp) to update an identity by clicking on **search identity** button on **Index page.**
- **User cannot update the USERNAME & PASSWORD of the any Identity**.

- User can update identities using **update** button.
- Once the update button is selected new **update page(update.jsp)** will be opened where user can update the required details.

4.4.1. Technical Description:

DAO Methods (Spring and hibernate framework is used)

- The method **update**(Identity identity) of interface **IdentityDAOInterface** is used to update an identity.

4.5. Delete Identity

- Users can **Delete Identity** on **Search Identity** web page (search.jsp)**.**
- User can delete identities using **delete** button.

4.5.1. Technical Description:

DAO Methods (Spring and hibernate framework is used)

- The method **delete**(int identityId) of interface **IdentityDAOInterface** is used to delete an identity.

5. Configuration instructions

5.1. Database Connection Details:-

```
<bean id="dataSource"
        class="org.springframework.jdbc.datasource.DriverManagerDataSource">

        <property name="driverClassName"
value="org.apache.derby.jdbc.ClientDriver" />
        <property name="url"
value="jdbc:derby://localhost:1527/HibernateTest2;create=true" />
        <property name="username" value="APP" />
```

6. Commented Screenshots

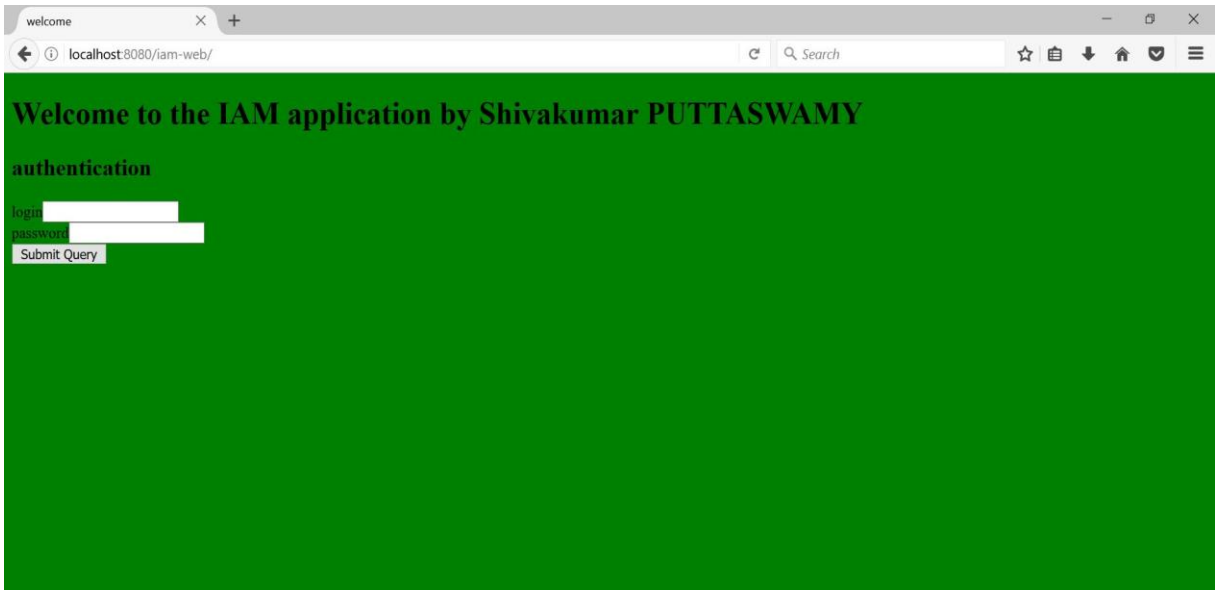


Fig 6.1 Welcome screen

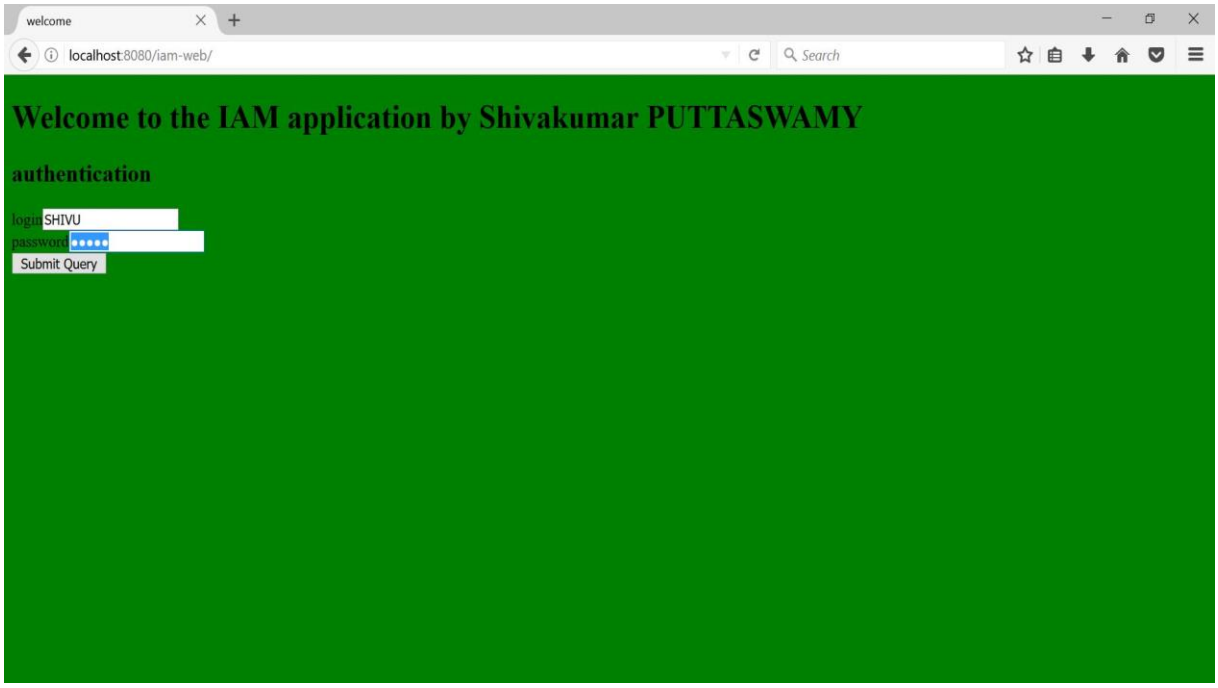


Fig 6.2 Welcome screen with user credentials like user name and password

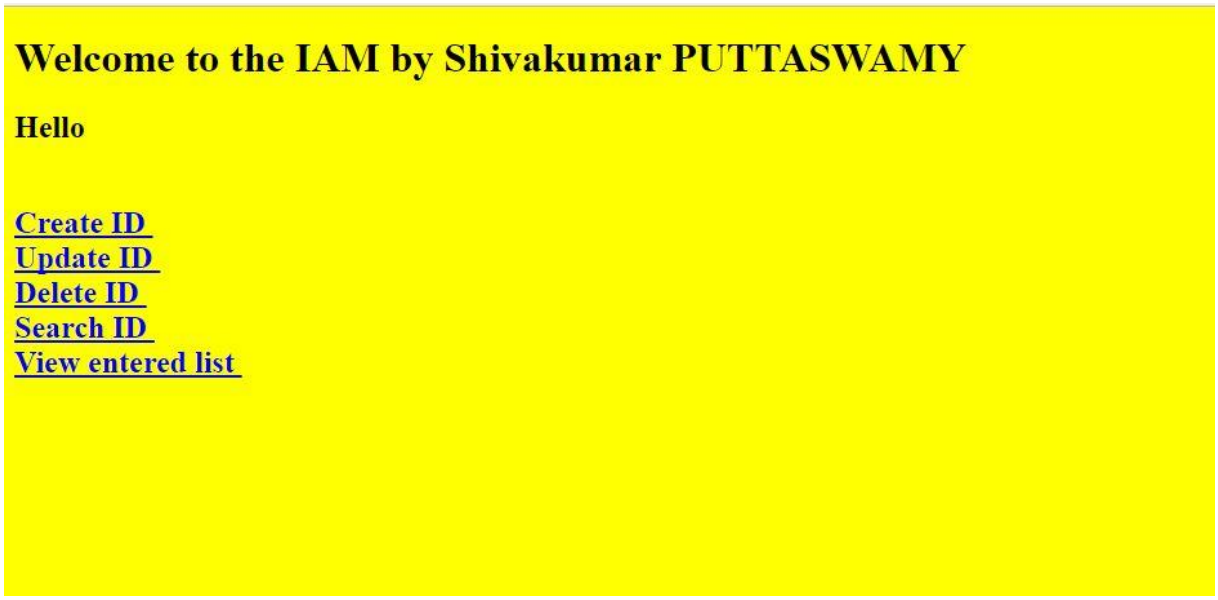


Fig 6.3 Once user authorised he can Create ID , Update ID , Delete ID , Search ID and View entered list .

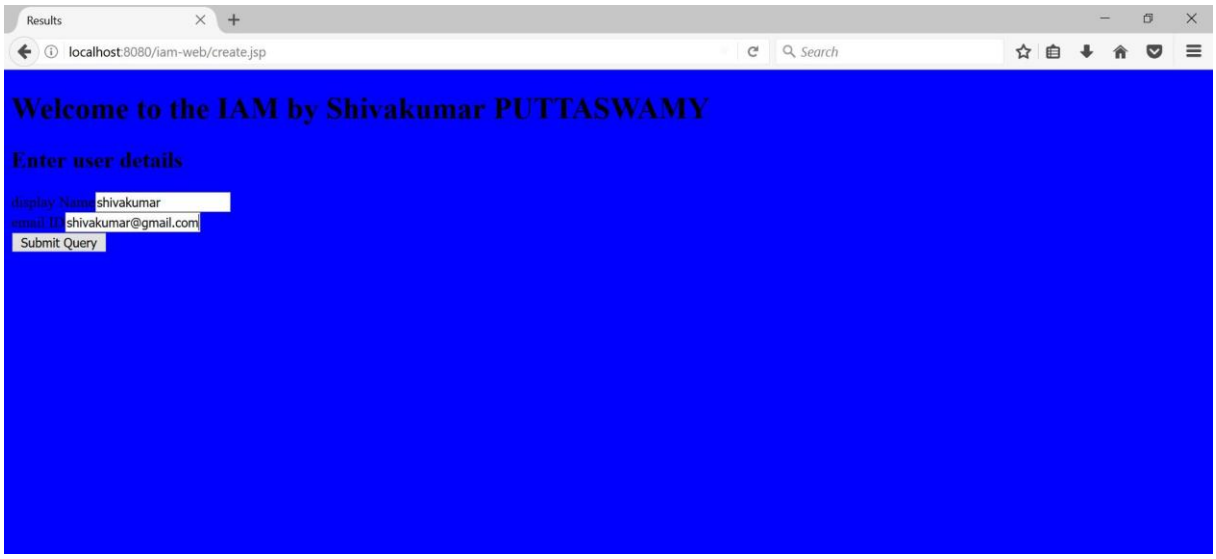


Fig 6.4 User can enter details like display name and email ID

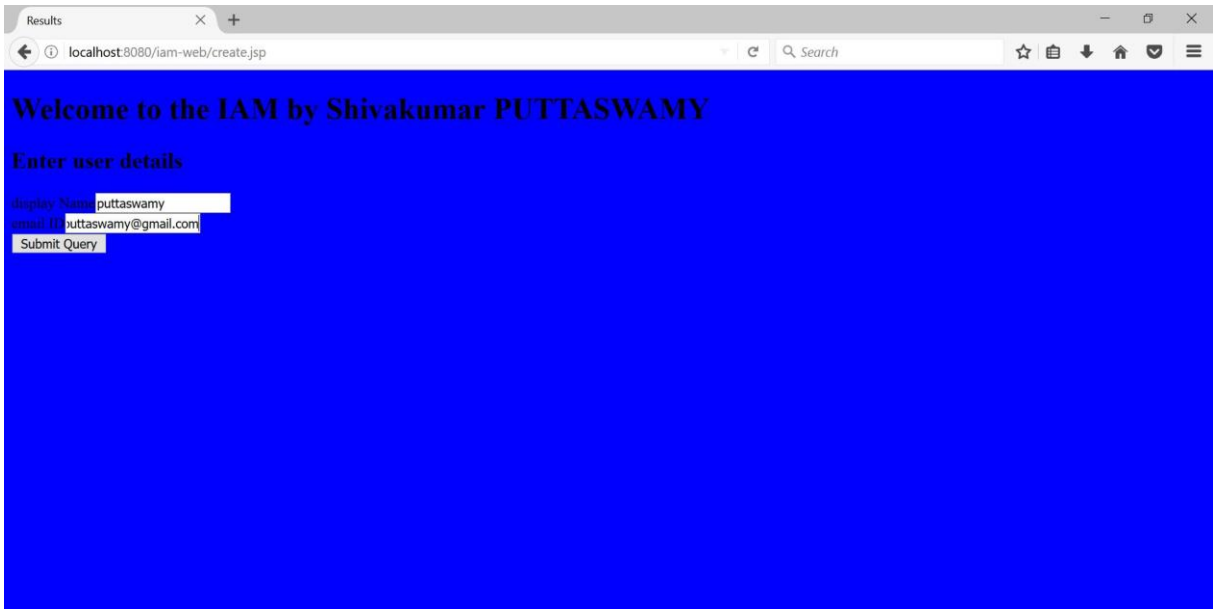


Fig 6.5 User can enter details like display name and email ID

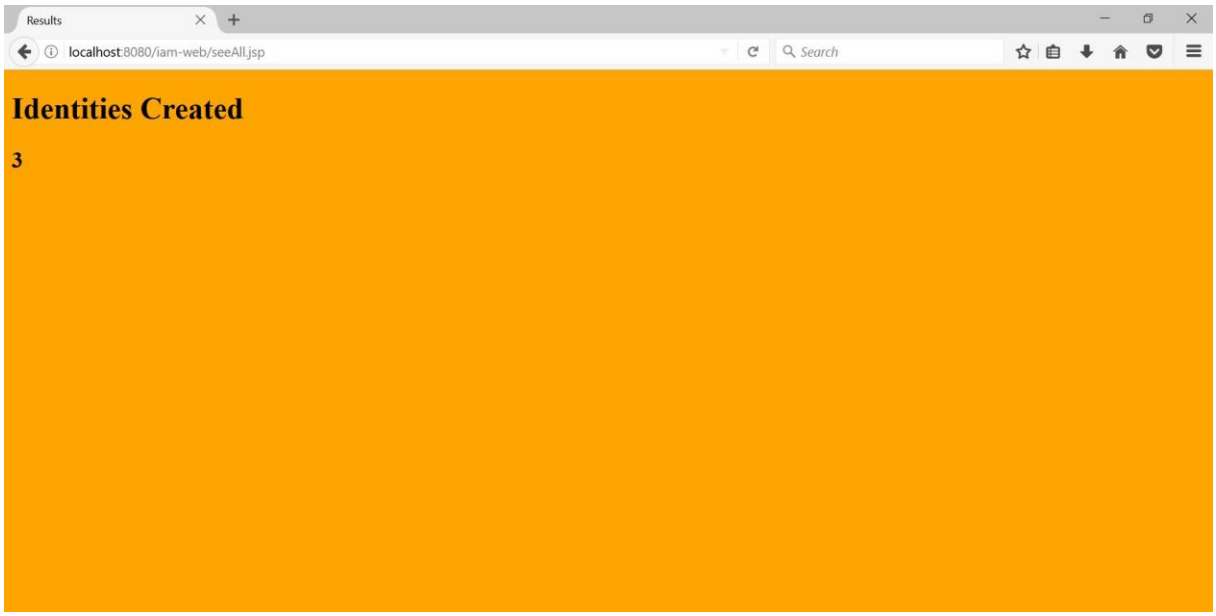


Fig 6.6 User can find out how many identities are created.

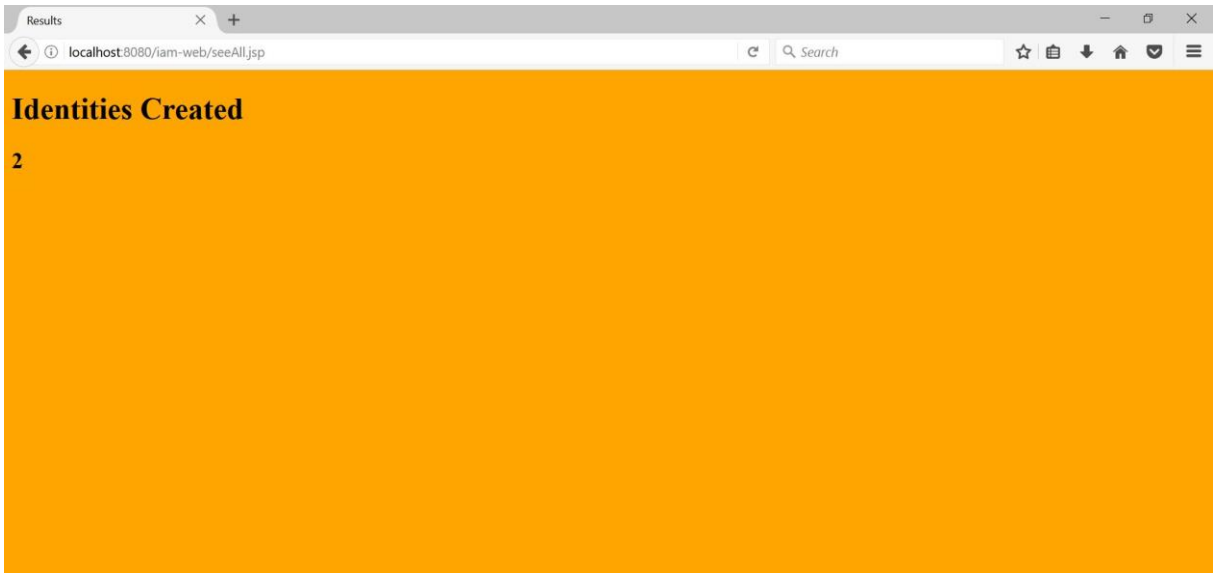


Fig 6.6.1 User can view how many identities are created.

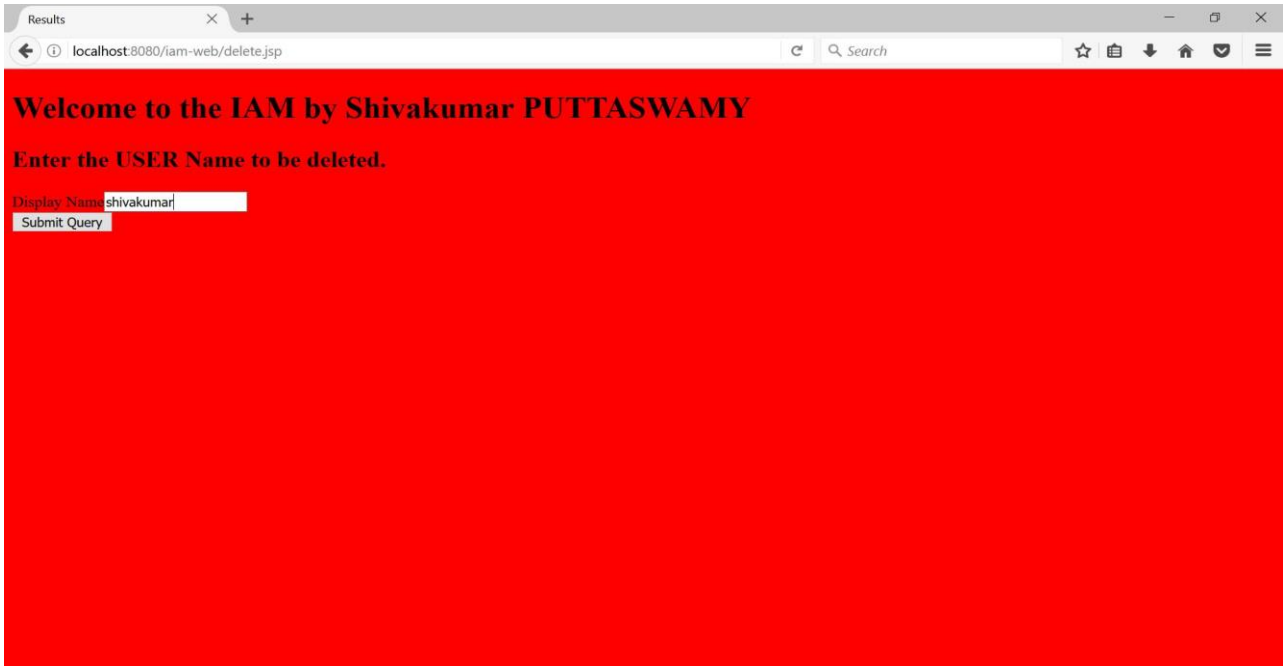


Fig6.7 Permission is given to delete user name with display name

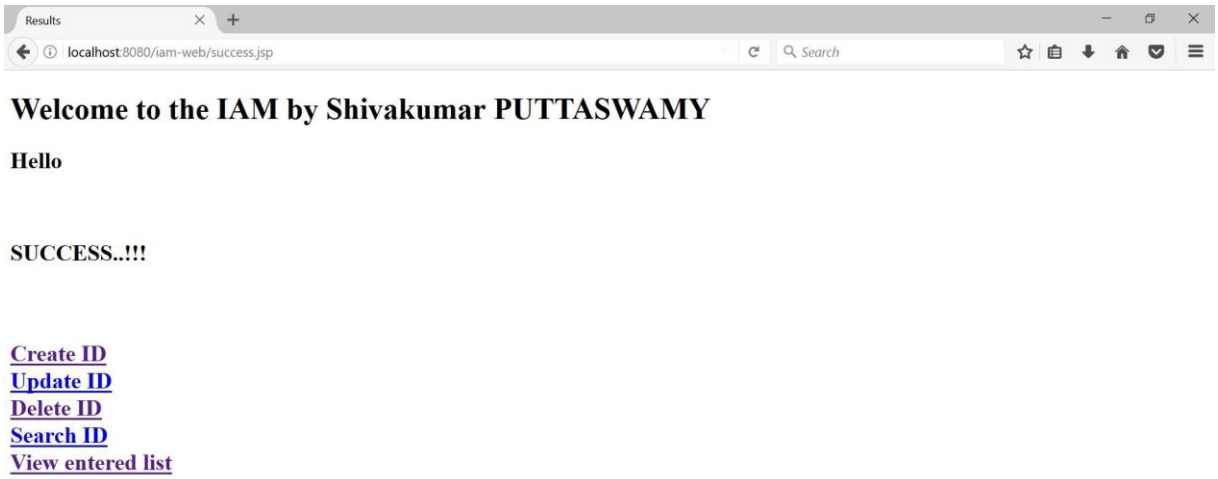
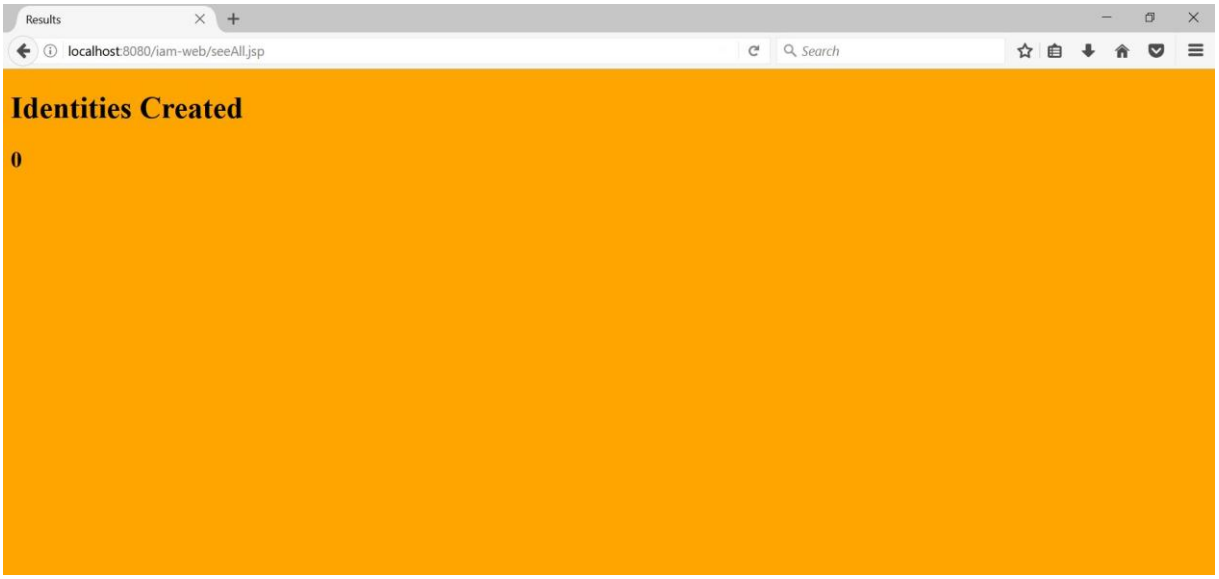
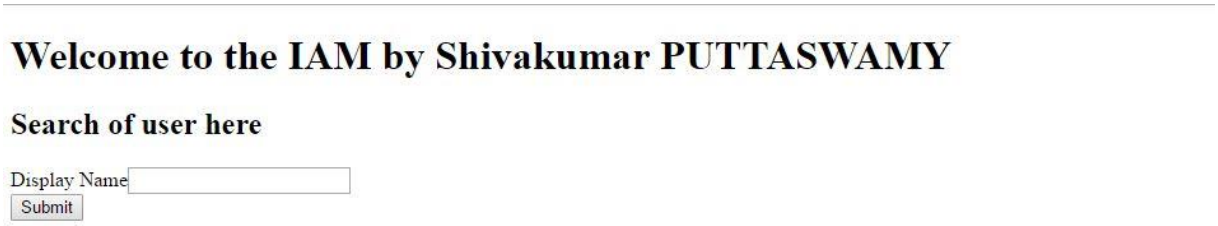


Fig 6.8 Success message after deleting user



6.9 User can view how many identities are remaining



6.10 Can search user with display name

Bibiliography

http://thomas-broussard.fr/

https://www.tutorialspoint.com

http://www.w3schools.com/

https://maven.apache.org/

http://www.youtube.com/