NewspaperApp

Bonus Assignment

Document/Version Number:	**Version 1.1**
Creation Date:	**April 30, 2015**
Account/Client:	**Indiana University**
Author:	**Milind Gokhale**
Editor:	**Milind Gokhale**
Last Edit Date:	**April 30, 2015**
File Name:	**NewspaperApp.docx**

Table of Contents

1   Problem Statement

Develop an application for the following problem statement. These are the minimum requirements.

• The application would be of a leading newspaper (something like New York Times)

• An editor can login into the application. (A proper database register, login using database should be applied. The database is resource constrained.)

• An editor can update article headline in his space.

• A news update needs to reflect in newspaper feed, internet article feed and twitter feed of the application. (They may be in tabs or links and should have different database entries for themselves)

• User should able to register (at least 5 mandatory fields like first name, last name, address etc). Only authenticated user should have access to the application. Make sure the database connection is shared.

You can develop either web application or get user input in command prompt.

Implement the appropriate design patterns.

Due date is 04/30.

2   Solution

Developed a web application for the problem statement.

• The application is for newspaper Bloomington Times

• An editor can login into the application and super user can create more editors. Login is through validated through database. Database is constraint and is accessed through singleton.

• An editor can update article headline in his space only.

• A news update is reflected in newspaper feed, internet article feed and twitter feed of the application. All the three types of feeds are visible to the user after login.

• User is able to register and login and then view internet, newspaper as well as twitter feeds. During registration user must enter first name, last name, contact number, email address and the desired password. Mandatory condition is handled at database level by making the columns not null. Only authenticated users have access to the application.

3   Design Patterns Used

3.1   Singleton

The database access is through a single shared connection. DatabaseHandlerSingleton class provides the connection anytime any query is required to be fired to the database. The singleton object is static and volatile and connection object is static. The constructor is private and the initialization of the class object is synchronized and controlled to allow a single instance.

3.2 Factory

The **feed** is the **product** and each **space/department** is the **creator**. There are 3 departments – Business, Entertainment and Sports. So the _BusinessEditor, EntertainmentEditor and SportsEditor_ are the creators. The respective products they create are the feeds related to Business, Entertainment and Sports respectively. (_BusinessFeed, EntertainmentFeed_ and _SportsFeed_).

3.3 Decorator

The **feed** is the **product** and the **decorators** are the flavors/variants of the feed like **newspaper**, **internet** and **twitter**. The same product news feed is visible in different variations to the user in the form of newspaper feed, internet deed and twitter feed. (Products are _BusinessFeed, EntertainmentFeed_ and _SportsFeed_) while Decorators are _BusinessFeedEditor_, _EntertainmentFeedEditor_ and _SportsFeedEditor_)
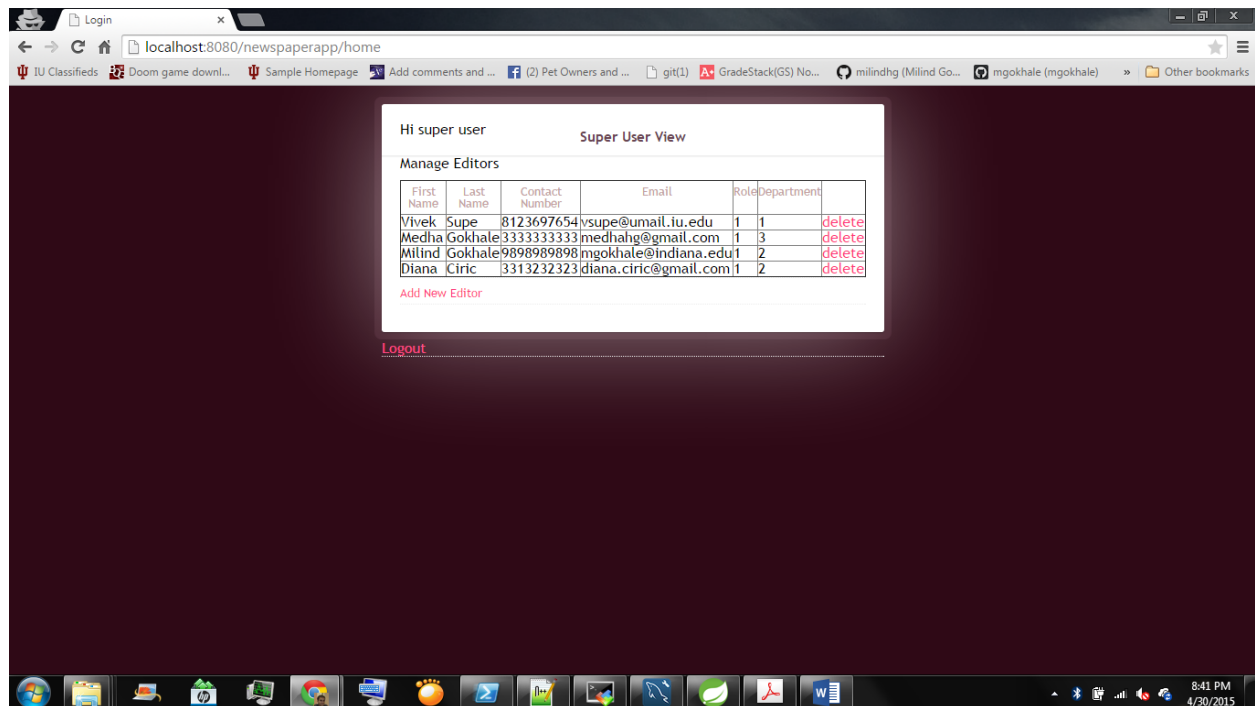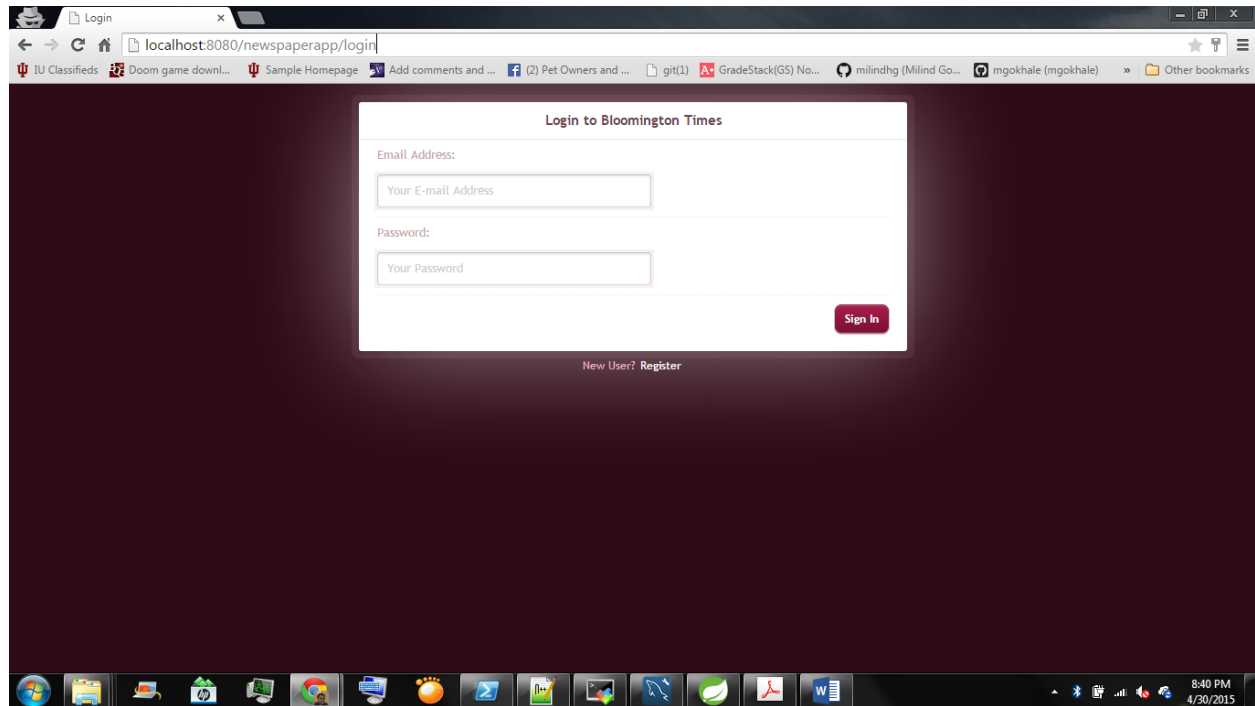
3.4 Proxy

Only authenticated users have permission to access the application and see the posts. For this the proxy design pattern is implemented. The proxy validates the user in isUserValid method and redirects to login page if the user is not authenticated. Similar flow is expected for editors – only authorized editors are allowed to modify add and delete feeds. So if a post is tried to be added or deleted by using modifying the URL, the user is validated in isEditorValid method and if the role of the user is not editor role then he is redirected to the login page. (_IUserController_ and _IFeedController_ are the two **interfaces**, _UserControllerProxy_ and _FeedControllerProxy_ their respective **proxy** implementations and the **main subjects** as _UserController_ and _FeedController_ respectively.

4   Data Structures Used

Arraylists are used to handle the multiple feeds fetched from the database. Arraylist allows easy iteration, decoration and addition deletion for feeds in the arraylist object.
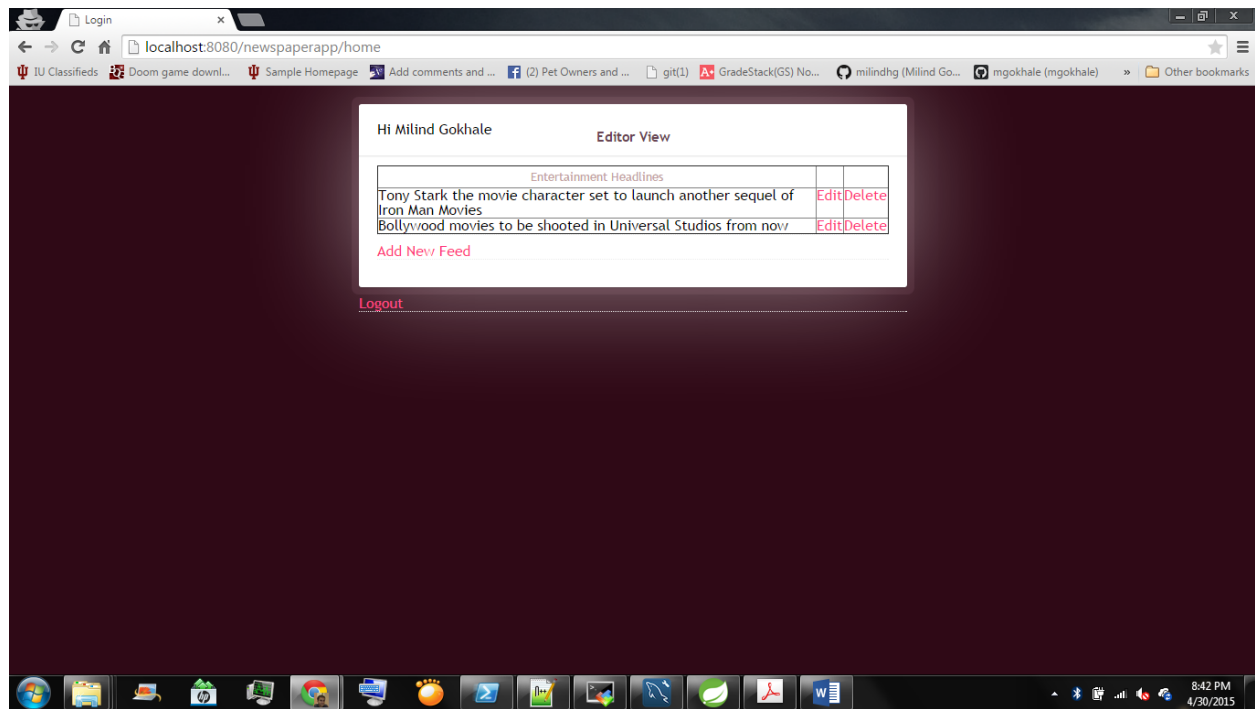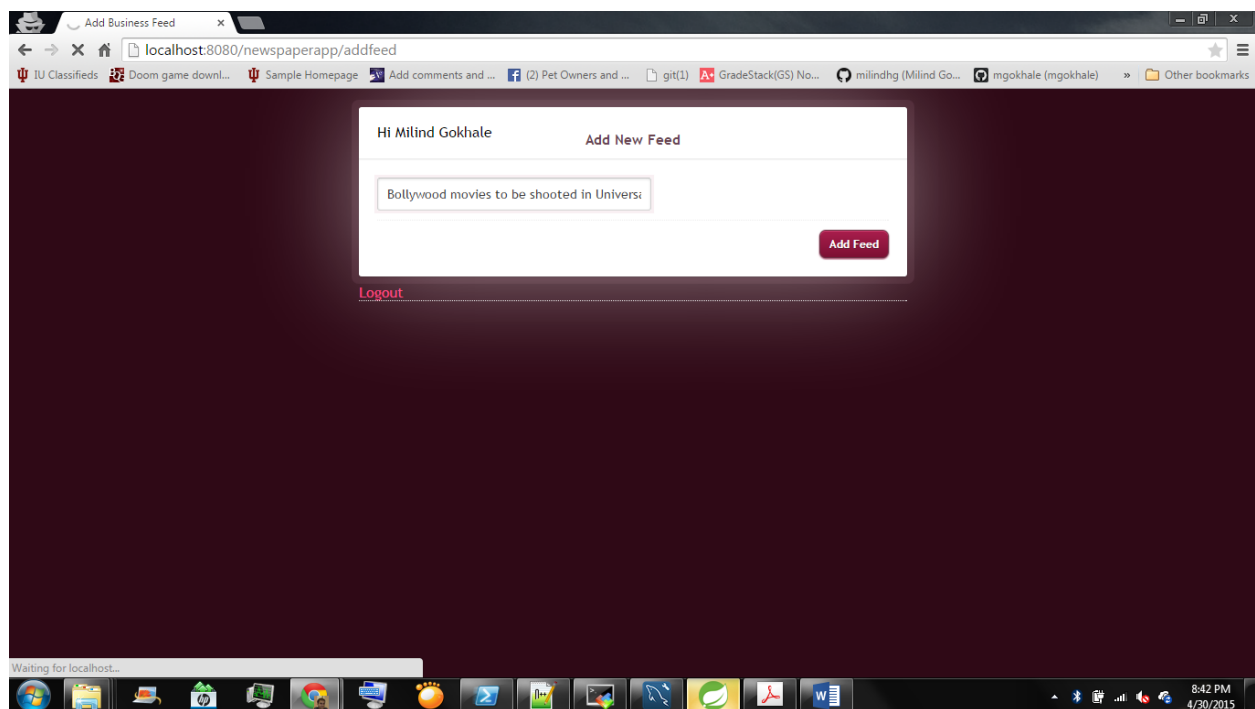
5   Screenshots

Figure 1: User Login

Figure 2: Super user view

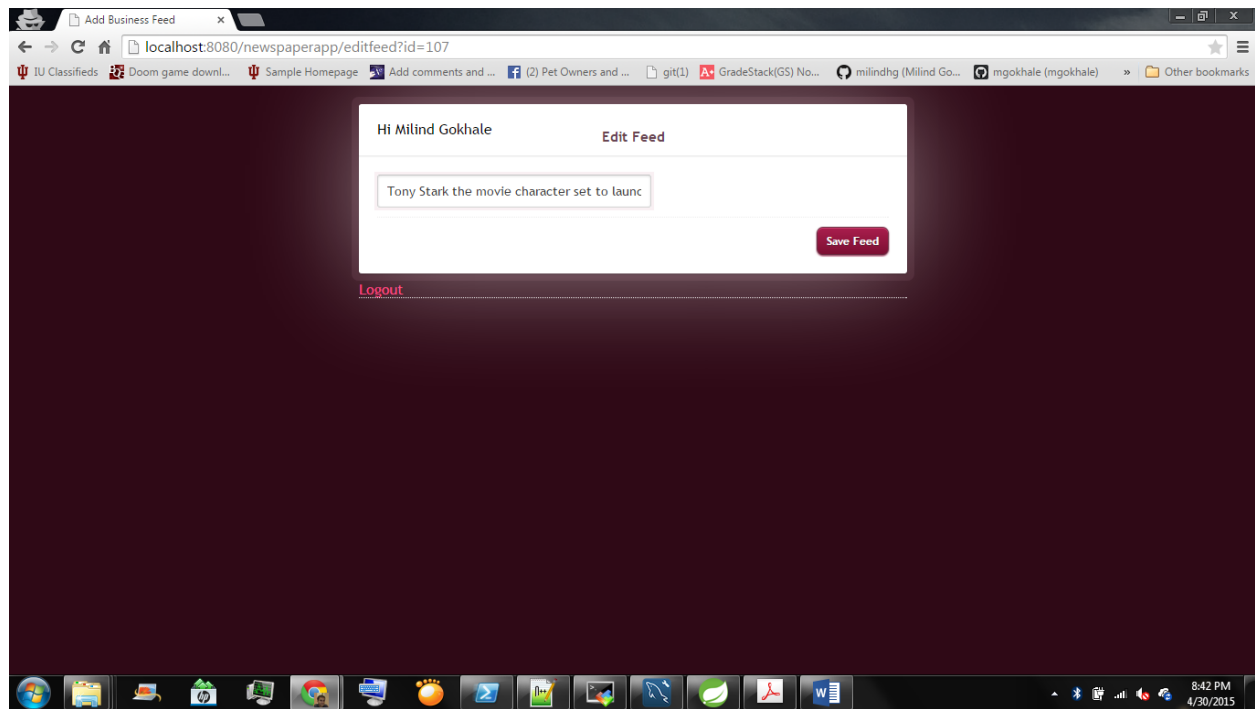*Figure 3: Editor View*

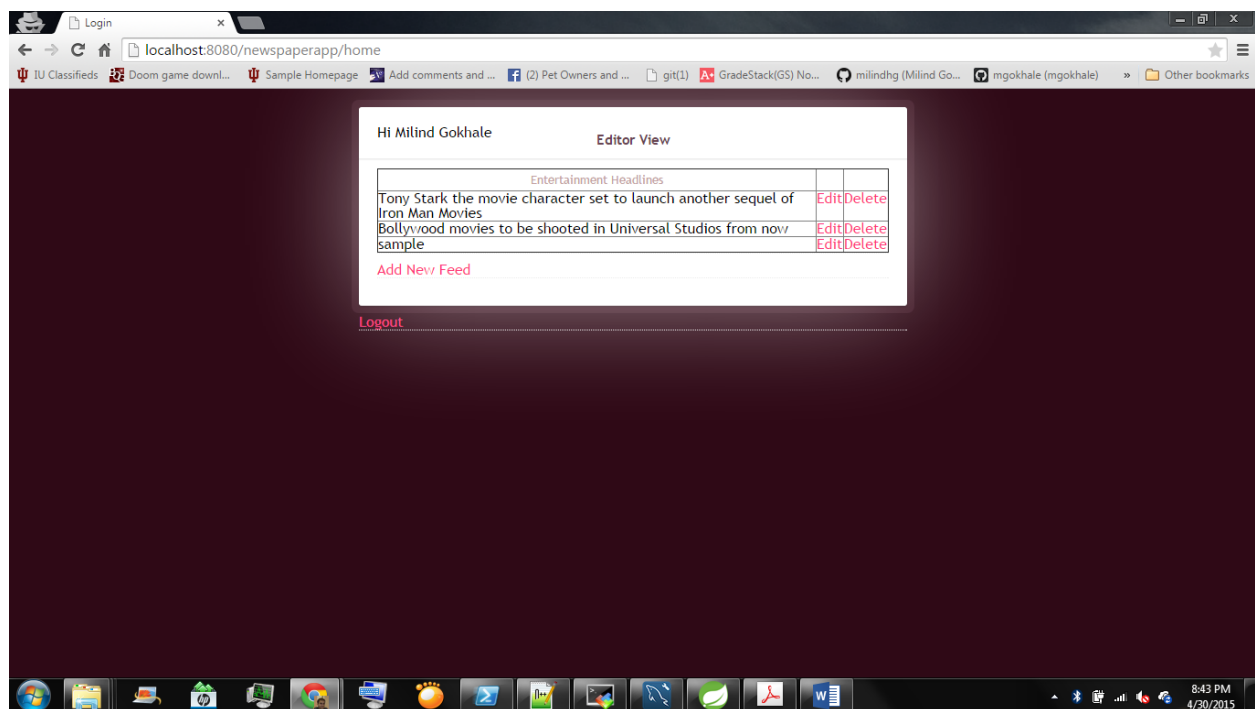*Figure 4: Add Feed*

Figure 5: Edit Feed
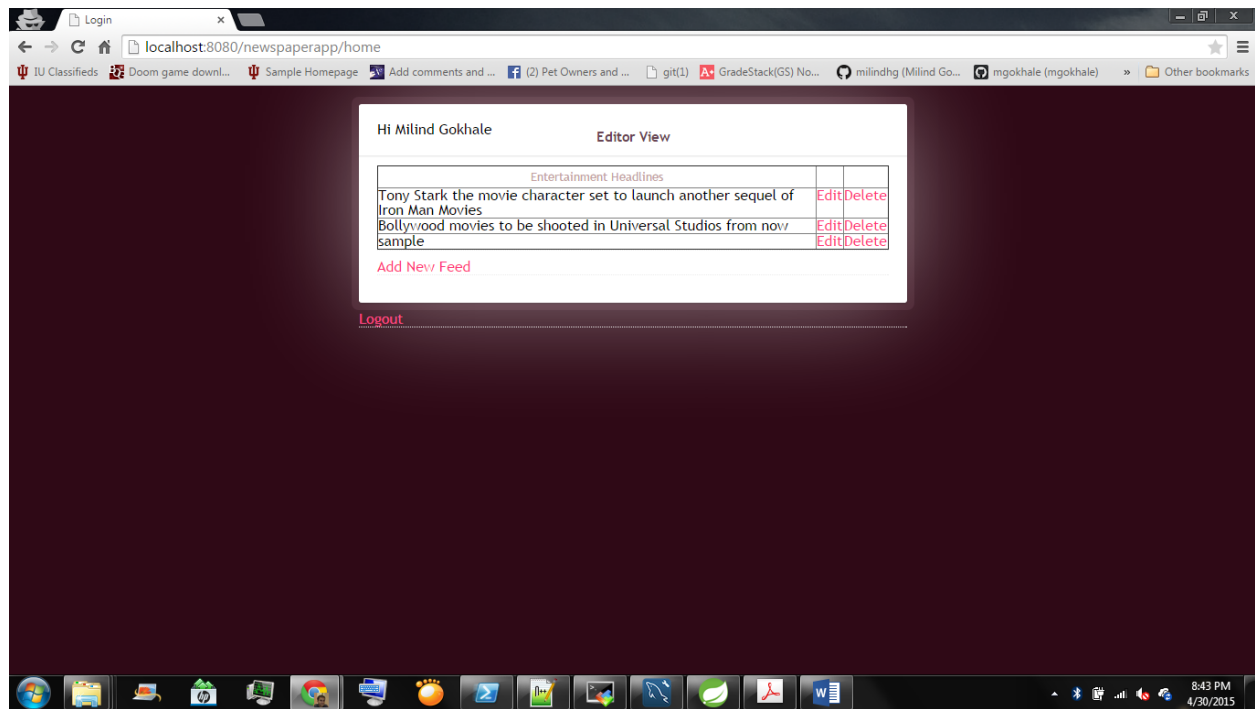

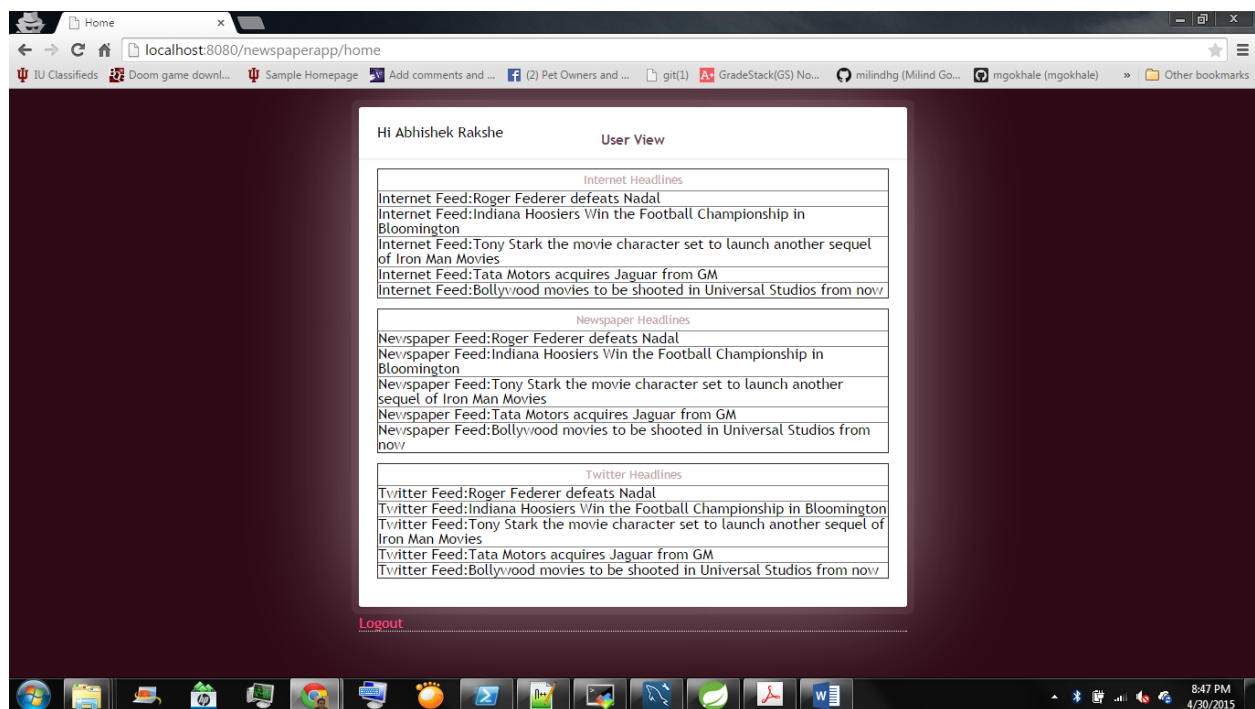*Figure 6: Before Delete*

Figure 7: After Delete

Figure 8: User view

References:

1. *iMomen Website Templates for Login and Register, www.iMomen.com.*