Project Ditto

by

"**Anurag Kumar**"
Roll No. **1326453**

"August 2015"

A 6 weeks training project submitted in partial fulfilment of the requirements
for the degree of Bachelor of Science in Computer Science.

The Computer Science Department
Swami Vivekanand Institute of Engineering & Technology
chandigarh-patiala Highway, Sector-8, Ramnagar, Banur
"2015"

I hereby certify that this 6 weeks training project, submitted by “Anurag Kumar,” conforms to acceptable standards and is fully adequate in scope and quality for the degree of Bachelor of Science in Computer Science.

_____ _____

Ms. Lalita Bhotani **Date**
Head Of Department (HOD) of Computer Science

The Computer Science Department
Swami Vivekanand Institute of Engineering & Technology
chandigarh-patiala Highway, Sector-8, Ramnagar, Banur
“2015”

Acknowledgements

I always thank my God as I remember you in my prayers, Philemon 1:4

I would like to acknowledge the assistance of Mr. Gaurav Kwatra, who was my tutor and mentor during my 6 weeks training program on Java Web based Application Development at the company, Kyrion Networking Solutions Pvt. Ltd., as he helped me to understand the programming languages and frameworks that I implemented in this project.

Course Objections

At the conclusion of Project Ditto, I will:

- showcase my knowledge and command of computer science principles by completing an acceptable project.
- appreciate the ***grand ideas*** of computer science by understanding fundamental computer science concepts.
- enhance programming skills though actual program synthesis.
- enhance technical communication skills by doing written documentation of my project.

Index

Project Proposal

This project is a light hearted **socialisation** project named "**Project Ditto**". The word Ditto is used in this project as the main concern that arises in this web based application is to share any random art of humour or "**Silly Deed**" performed by one of the users, which of course will be visible to every user on the page itself, and if any other user has also done that deed or took it as a challenge and perform that deed after reading can increment the ditto or "**Me Too**" count of the deed itself.

Users could also use the idea of **challenging other users** to perform the task as an act of social service, they can put a selfless deed on the list of tasks and then whosoever performs the deed can acknowledge the greatness of deed and of course **increment the ditto count at a vary fast rate**. Or users could just use this application to add random acts of humour that they would like others to perform.

The **User Interface or UI** of this project is very lite as no pre-rendered style sets are used in this project. Also improving the relationship between a user and the application itself. So that it can be easily used at a very heavy amount of frequent user interactions without any technical glitch. I have tried to reduce the amount of button clicks used to interact with the application for the sake of creating memorable **User Experience or UX** for the account holders on the web page.

The below standard operations are available for any user to interact with the application:

- Signing up (highly recommended).
- Logging in.
- Publishing a Post or a Silly Deed.
- Edit any Post created by it's respective user only.
- Pushing the "Ditto Me" button for any deed to acknowledge that user has performed that deed himself / herself.
- Viewing the list of all the deeds uploaded on the web application ordered according to the publishing date (feature provided on all viewing pages).
- Deleting any published deed by it's respective user only.
- Logging out of the application (not recommended).

Resources Used

Below is the list of resources I used to develop this project (including programming languages, IDE and all the frameworks):

- **Spring Tool Suite (RELEASE 3.1.11)**: STS is an eclipse based spring framework oriented IDE for J2EE development.

- **Pivotal Virtual local-host server**: as a web application is deployed on a server, rather than using a real server that could be very expensive, I used a virtual local-host generated server for deploying the application.

- **Oracle Express Edition**: it is a well known database management tool used for back-end database support for any application that requires to save any user data externally and further retrieval of that data even after the application is rebooted.

- **Spring Framework**: Spring framework is used to solve a special purpose that is to introduce dependency injection in the project. In this project, spring is used to create a hibernate framework configuration model and then use it in different services provided by the application through auto-wiring that model to those services.

- **Hibernate Framework**: Hibernate framework is used to efficiently improve the back-end database connectivity. It is most commonly used to create a model-to-table relation that is to connect a custom class model used in the application and create a table in the database with the same configurations as the model itself. It also creates secure connections to the back-end.

- **Maven repository**: maven is used for automatically generating a standard project directory structure for a vast amount of varying applications and automate the process of adding dependencies for those projects. As we provide the details for any dependency that we want to be added to the project, maven finds those dependencies online and import them to our projects.

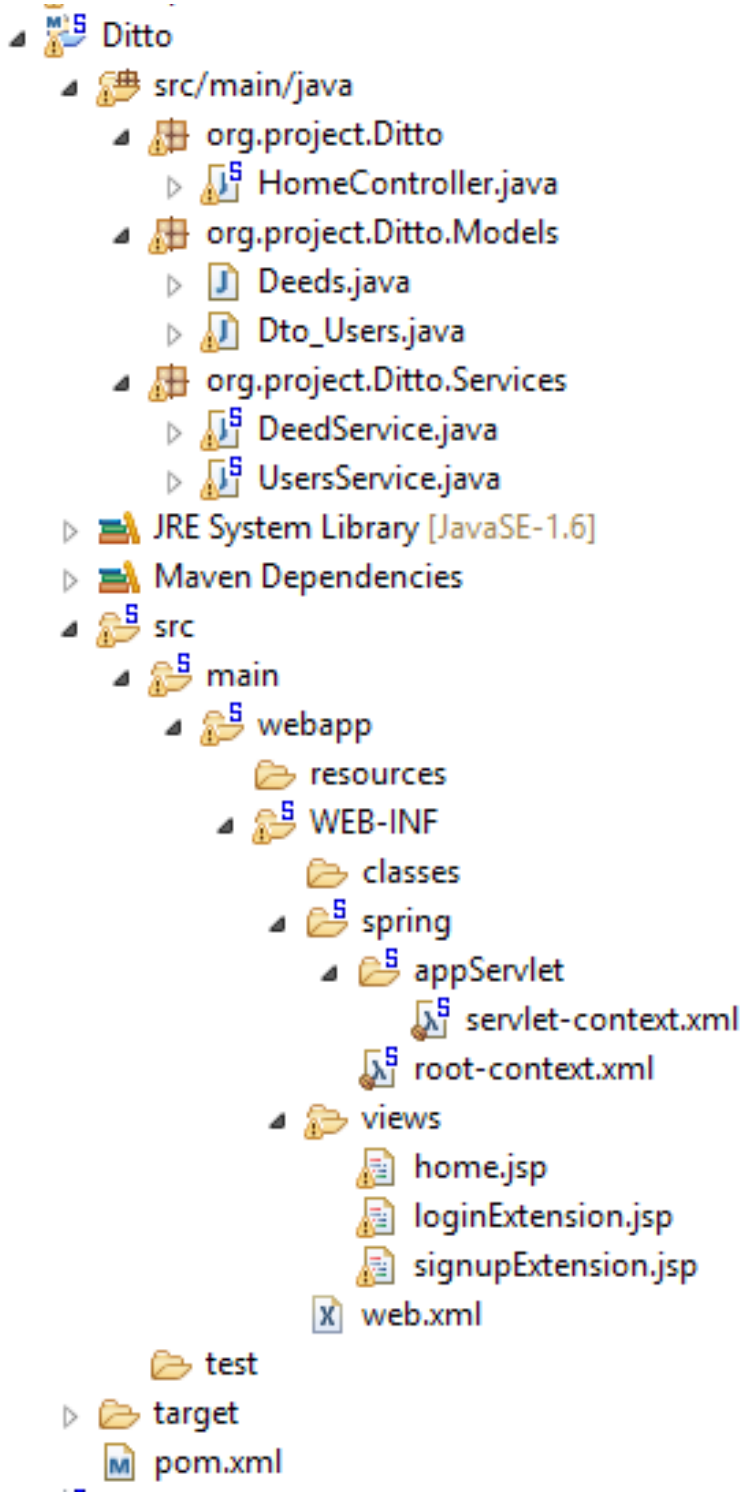
- **JAVA-J2EE programming language**: the heart of this project lies with JAVA, it is a very trending programming language, the most common use of JAVA includes web applications and Android platform application development. And J2EE include the web application part of JAVA.

- **HTML**: html is used to create static web page layouts.

- **CSS**: css is used for styling of the html web page.

Project Directory Hierarchy

Below is the image of project directories and files in hierarchical presentation.



Design Documentation

Most of the **user interface** introduced in this project are **custom style-sheets** designed by myself. I rejected the idea of using pre-rendered style sheets as the project wasn't huge enough to be worried about implementation of graphical interfaces and style-sheets css most commonly used like Twitter Bootstrap css libraries. I took this initiative for more personnel reasons including learning to create custom style-sheets and to improve the user experience by removing add-on libraries as they tend to over weight the project, they should only be used if necessary.

The whole web application interface is written in **JSP (JAVA Server Page)**, that is an essential part of JAVA web application as it provides the freedom of creating a **dynamically rendered page**, a JSP can get data from the browser and produce a layout as per the requirement and the provided data during runtime, which is the essence of JAVA web application development.

List of **basic elements of html used in JSP** for this project are as:

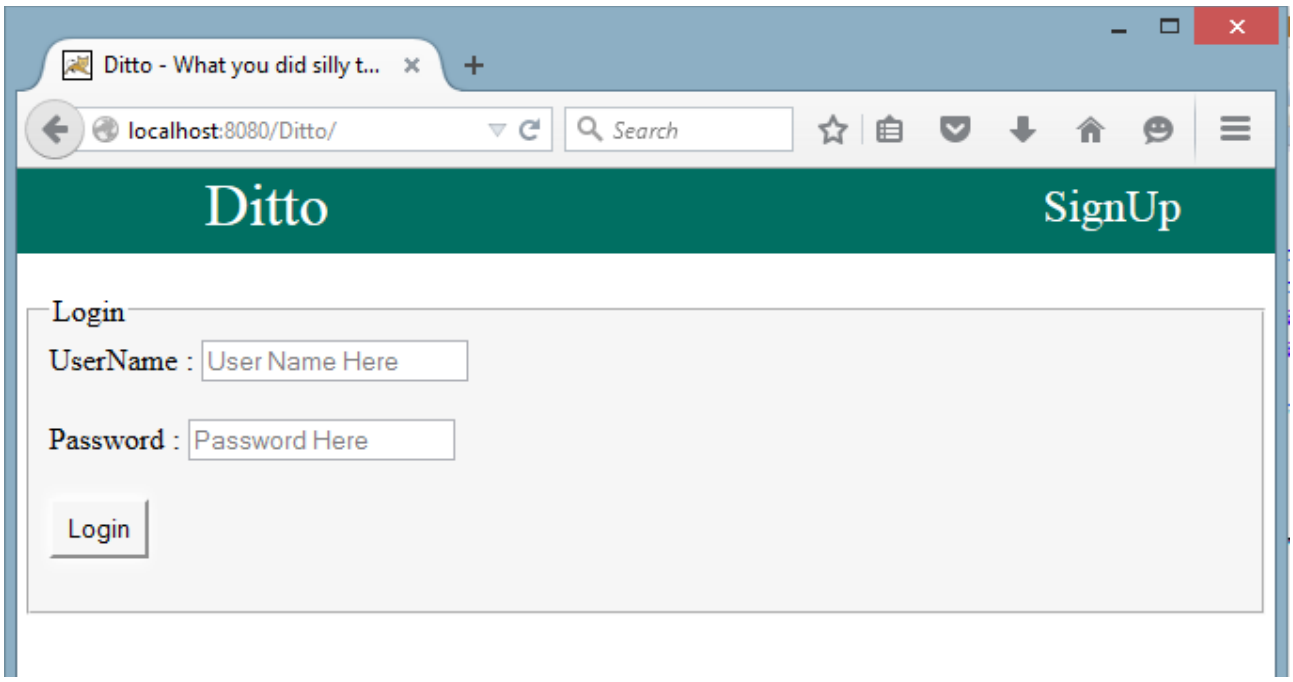
- Field-Sets
- Text-Fields
- Text-Areas
- Buttons

I have implemented least button clicks needed as much as possible, as it is an essential part of user-to-application interactions. The more clicks and entries that user require to do, the less users revisit. This procedure is well used in the industry for creating habit forming applications. Improving the **overall user experience**.

All the layout implementation is done in a **JSP format as HTML tags**, that is i have used the JSP to dynamically provide the necessary layout of the web application according to user needs, but the actual layout is still hard-coded in HTML only. All the extended custom makeover, or view style-sheets design is written in CSS language.

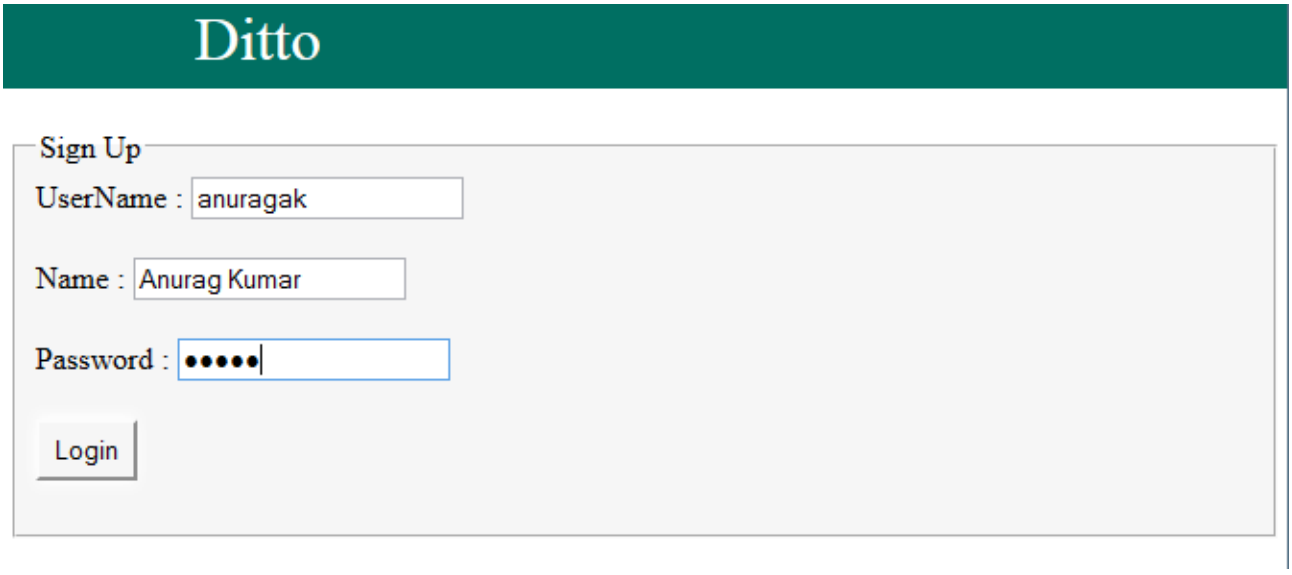
It also include some nice animations like if hover over the visible tile of deed on the page, little extension of that deed pops up below the tile offering the user to view its ditto count, edit or delete the deed and increment the ditto count. This is performed using the hover options of CSS itself, very handy feature of css.

Application Screen-shots



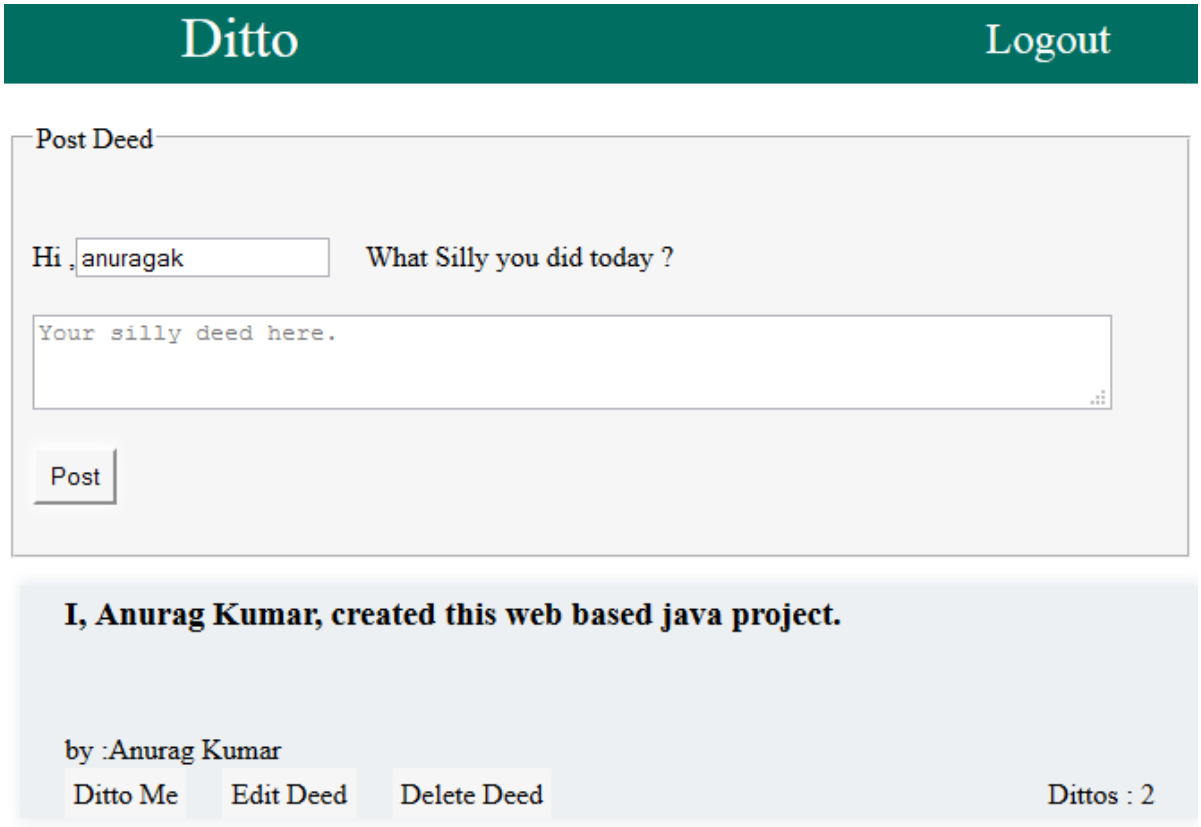
Home Page

The above page provides the facility of logging in to the user or signing up if the user doesn't hold any existing account for this application.



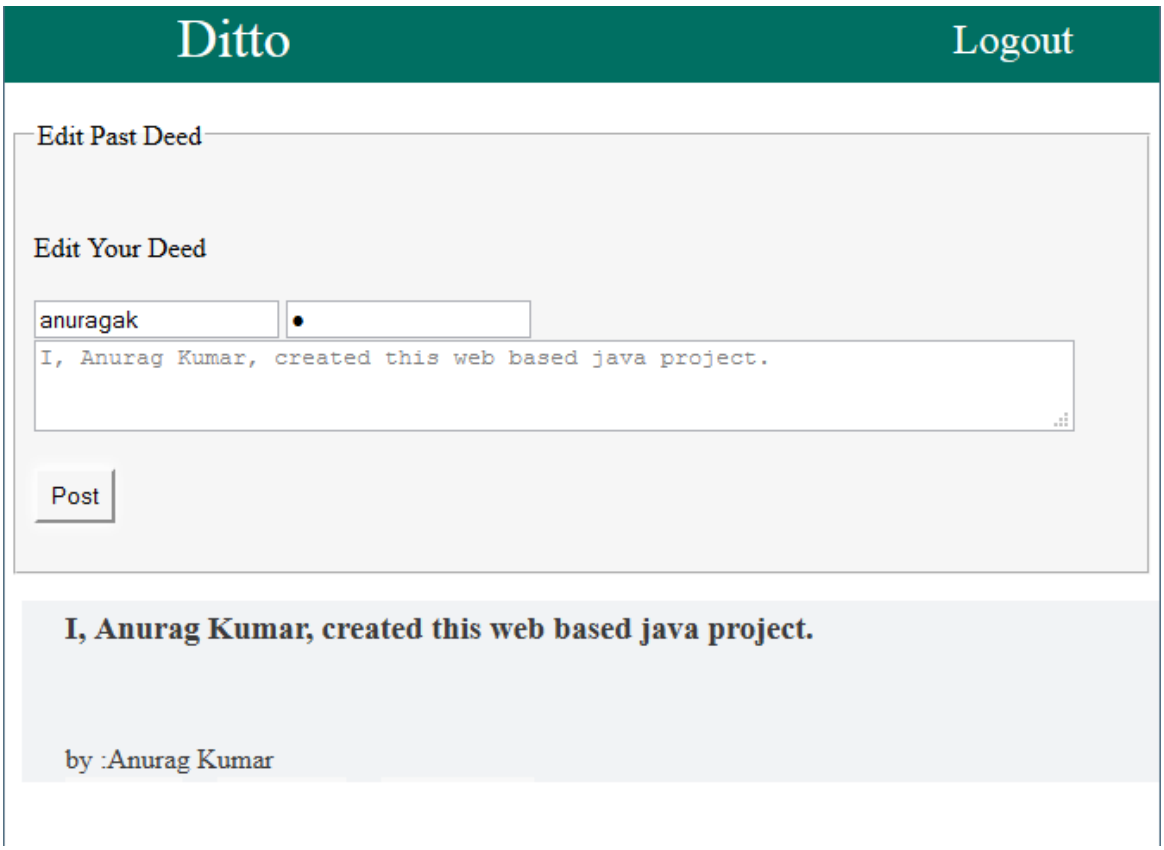
Sign-Up Page

Here the new users can sign up for an account on the application to access the privileges that only an account holder can access.



Posting Deed Add-on to the user page

In the above page, any logged in used can view all the posts or deeds ever uploaded in the application, and also add a fresh new post to the list for others to see, as the user hovers over the posts, he/she will find the options for the posts visible in the screen-shot for ditto Me, Editing, Deleting the posts that can only be done by the user that created this post. And view the count of Dittos, or the number of times other users performed this deed.



Editing Deed Add-on to the user page

The above page provides the facility of editing any previous deed that the current logged in user posted. And after editing, the new deed will be visible for all the users to acknowledge.

Below is the Github link for the documented project, as I prefer to give a link from where you could also note the improves and version control with full transparency, so rather than providing a hard copy of the project files I provided the Github link here:

Project-Ditto:

https://github.com/anuragkumarak95/Project-Ditto/