MY BLOG WEB APPLICATION

USING SPARK JAVA FRAMEWORK AND MONGODB DATABASE.

Submitted by:

AKSHAY MAREWAR

(amarewa1@binghamton.edu)

Under the Guidance of:

Prof. Leslie Lander

Department of Computer Science

Thomas J. Watson School of Engineering and Applied Science

Binghamton University

Table Contents:

1. INTRODUCTION:

A blog is a discussion or informational website published on the World Wide Web consisting of discrete, often informal diary-style text entries ("posts"). Posts are typically displayed in reverse chronological order, so that the most recent post appears first, at the top of the web page. Until 2009, blogs were usually the work of a single individual, occasionally of a small group, and often covered a single subject or topic. In the 2010s, "multi-author blogs" (MABs) have developed, with posts written by large numbers of authors and sometimes professionally edited. MABs from newspapers, other media outlets, universities, think tanks, advocacy groups, and similar institutions account for an increasing quantity of blog traffic. The rise of Twitter and other "microblogging" systems helps integrate MABs and single-author blogs into the news media. Blog can also be used as a verb, meaning to maintain or add content to a blog.

The purpose of MyBlog is to express yourself and share your passion, to share your knowledge, to refine the writing skills, to build the professional network, to earn the professional network, to build the online portfolio and to market our business.

2. PROJECT OVERVIEW:

The main purpose of the blog is to read the blog and to write the blog.

Main Features:

1. Read the Blog.
2. Write the Blog.
3. Comment on the blog.
4. Create the Tags.
5. Search the blogs using Tags.
6. Like the comment.
7. Get the latest 10 blogs.
8. Multiuser support.
9. User Authentication.

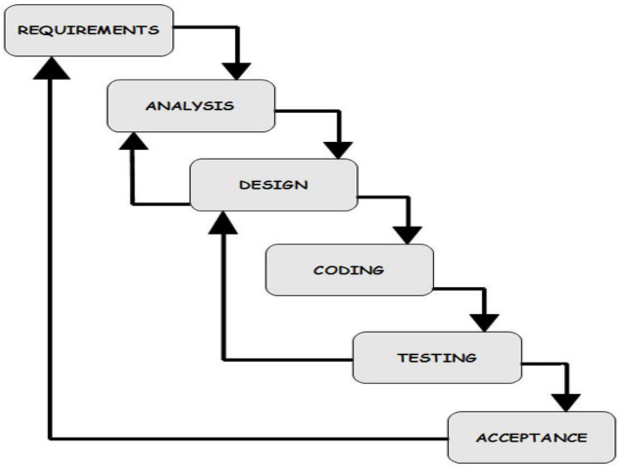
3. ANALYSIS AND DESIGN:

I have started the project from scratch and tried to overcome all the possible bugs and errors. The main development I would call is openness of this project. As I have programmed this project in Java, which is an open source language for developing software. This becomes easy for user to understand the software as well as to create new environments. User can also come to easily debug environment, if same exists.

3.1 Process Model

Process models are processes of the same nature that are classified together into a model. Thus, a process model is a description of a process at the type level. Since the process model is at the type level, a process is an instantiation of it. The same process model is used repeatedly for the development of many applications and thus, has many instantiations.

I have used The **waterfall model** which is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation, and Maintenance.



The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping.

4. TECHNOLOGIES USED:

4.1 Hardware Requirements:

1. Computer with processor (at least 1.5 Ghz)
2. Ram – 512 MB (at least)

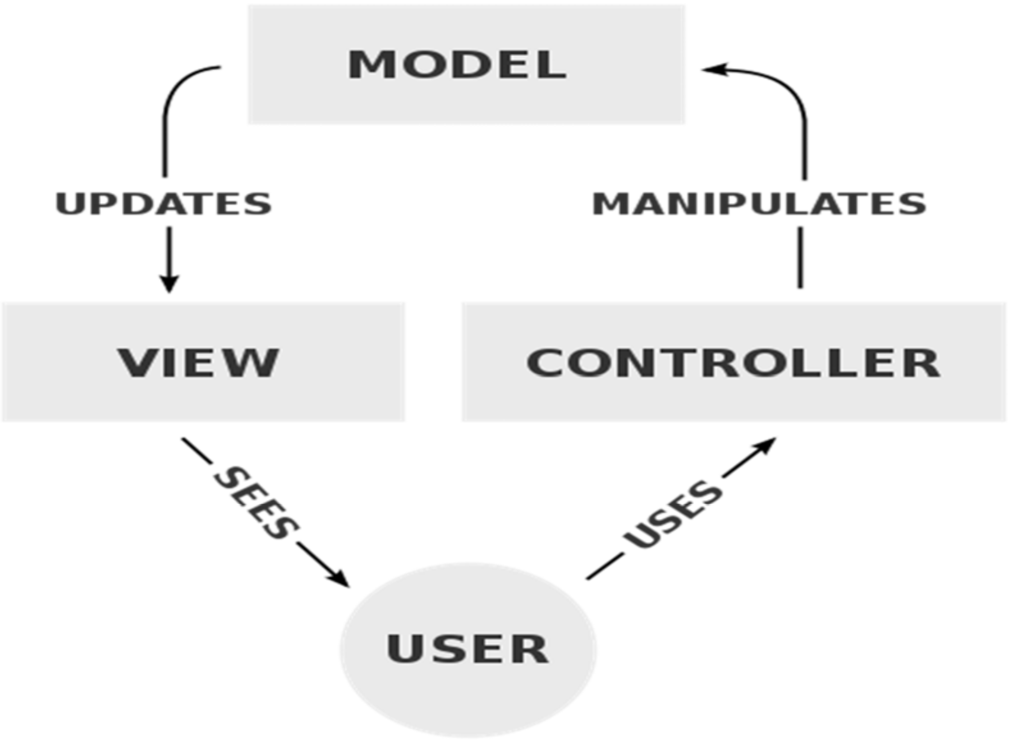
4.2 Software Requirements:

1. operating system (Windows) any version
2. JRE 1.8.0_121
3. Eclipse
4. Spark Jars and Freemarker Jars.
5. MongoDB 3.4 Database
6. MongoDB Java Driver
7. Jetty Server

5. IMPLEMENTATION:

For implementation I used Spark Java framework which is MVC framework.

The Architecture for MVC is as follows:



As with other software architectures, MVC expresses the "core of the solution" to a problem while allowing it to be adapted for each system. Particular MVC architectures can vary significantly from the traditional description here.

Components:

The model is the central component of the pattern. It expresses the application's behavior in terms of the problem domain, independent of the user interface. It directly manages the data, logic and rules of the application.

A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.

The third part, the controller, accepts input and converts it to commands for the model or view.

Interactions:

In addition to dividing the application into three kinds of components, the model–view–controller design defines the interactions between them.

A model stores data that is retrieved according to commands from the controller and displayed in the view.

A view generates new output to the user based on changes in the model.

A controller can send commands to the model to update the model's state (e.g., editing a document). It can also send commands to its associated view to change the view's presentation of the model (e.g., scrolling through a document).

The main components of application are:

1. BlogController
2. UserDAO
3. SessionDAO
4. BlogPostDAO

The BlogController is the main controller of the application which handles all the GET, POST, PUT, DELETE request to/from browser.

In computer software, a data access object (DAO) is an object that provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, the DAO provides some specific data operations without exposing details of the database. This isolation supports the Single responsibility principle. It separates what data access the application needs, in terms of domain-specific objects and data types (the public interface of the DAO), from how these needs can be satisfied with a specific DBMS, database schema, etc. (the implementation of the DAO).

The UserDAO is the Data Access Object which specifies the data operation regarding User collection in the Database. The data operation includes creation of the user and storing and retrieving the user details like Username, password and emailId for authentication.

The SessionDAO is the Data Access Object which specifies the data operation regarding Session collection in the Database. The data operation includes storing and retrieving the sessionId of the session and username.

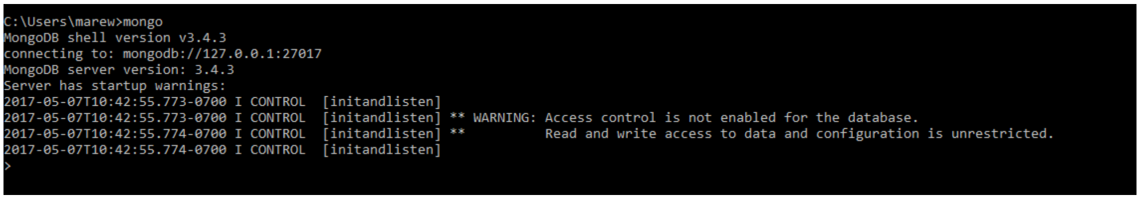
The BlogPostDAO is the Data Access Object which specifies the data operation regarding Posts collection in the Database. The data operation includes storing and retrieving author, title, tags, comments, body regarding posts.

For the View i.e. Front end of the application I used Freemarker template.

For the deployment I used the Jetty Server.

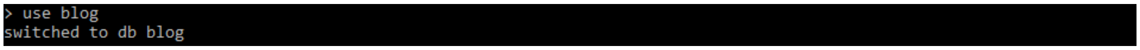
I used the MongoDB NoSQL database for application.

To connect MongoDB database.



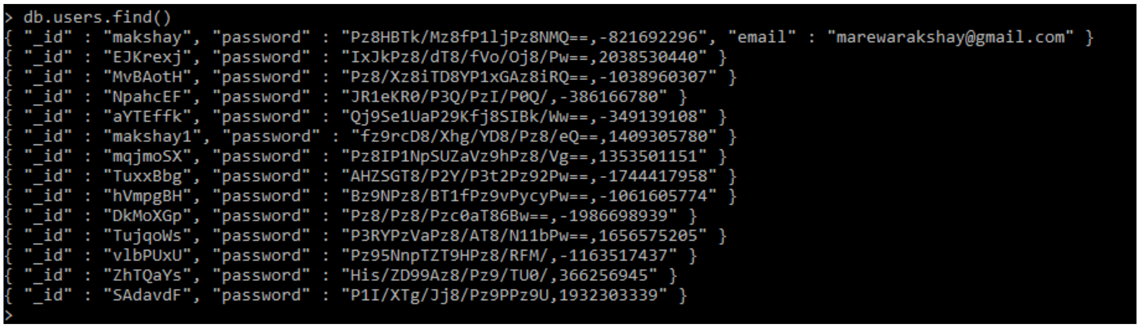
```
C:\Users\marew>mongo
MongoDB shell version v3.4.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.4.3
Server has startup warnings:
2017-05-07T10:42:55.773-0700 I CONTROL  [initandlisten]
2017-05-07T10:42:55.773-0700 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2017-05-07T10:42:55.774-0700 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2017-05-07T10:42:55.774-0700 I CONTROL  [initandlisten]
>
```

Connect to the blog database:



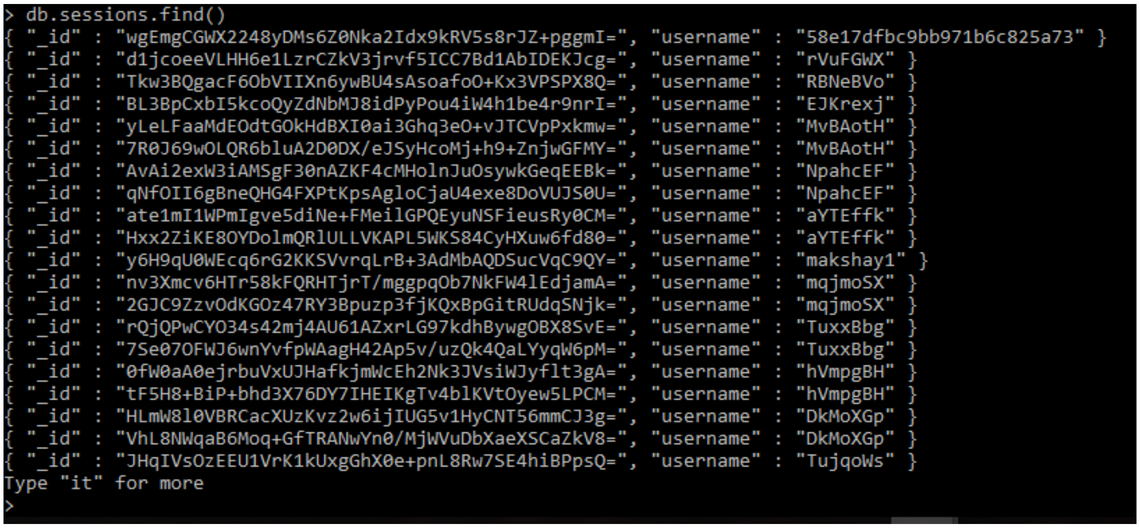
```
> use blog
switched to db blog
```

User collection:



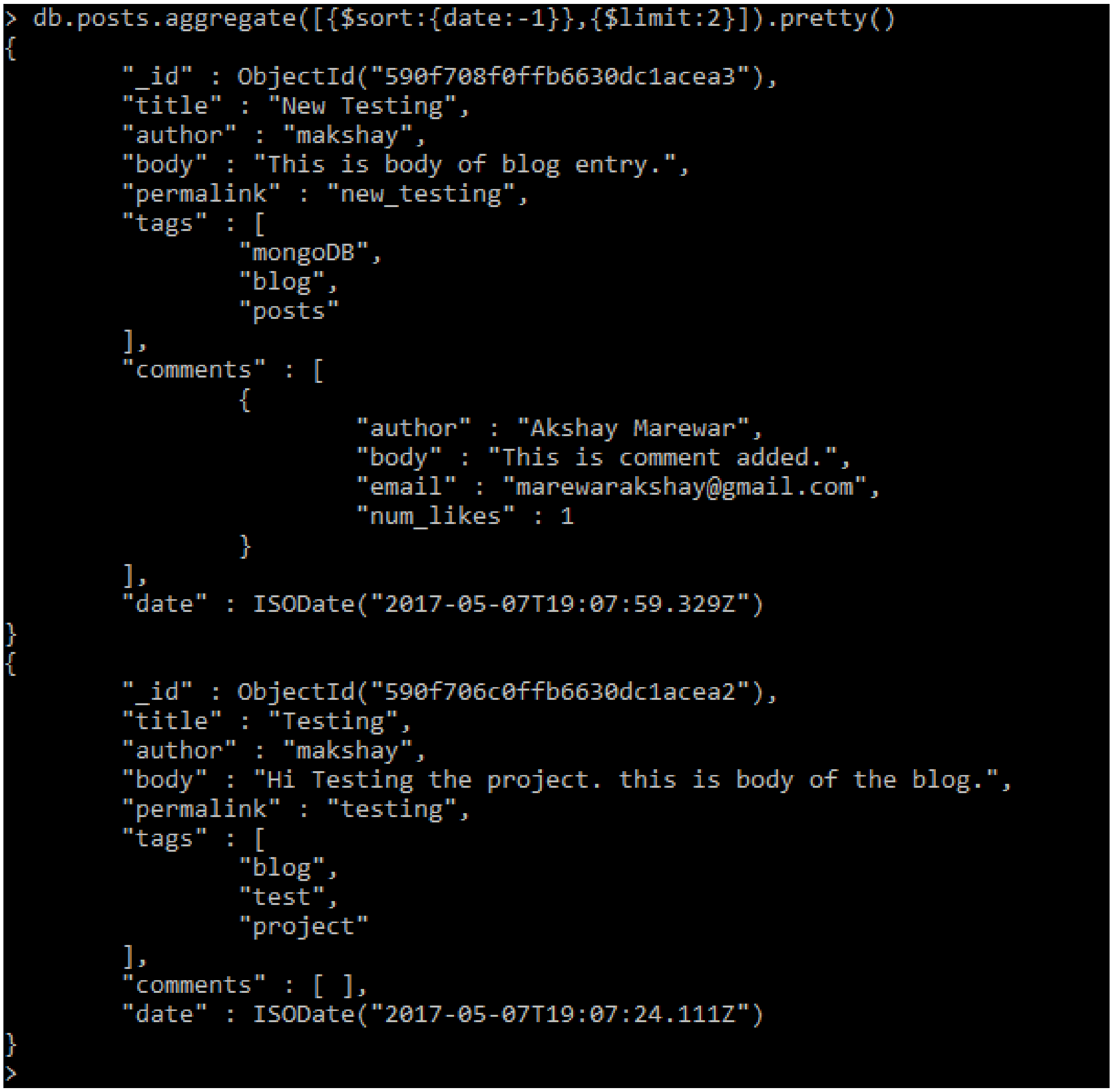
```
> db.users.find()
{ "_id" : "makshay", "password" : "Pz8HBTk/Mz8fP1ljPz8NMQ==,-821692296", "email" : "marewarakshay@gmail.com" }
{ "_id" : "EJKrexj", "password" : "IxJkPz8/dT8/fVo/Oj8/Pw==,2038530440" }
{ "_id" : "MvBAotH", "password" : "Pz8/Xz8iTD8YP1xGAz8iRQ==,-1038960307" }
{ "_id" : "NpahcEF", "password" : "JR1eKR0/P3Q/PzI/P0Q/,-386166780" }
{ "_id" : "aYTEffk", "password" : "Qj9Se1UaP29Kfj8SIBk/Ww==,-349139108" }
{ "_id" : "makshay1", "password" : "fz9rcD8/Xhg/YD8/Pz8/eQ==,1409305780" }
{ "_id" : "mqjmoSX", "password" : "Pz8IP1NpSUZaVz9hPz8/Vg==,1353501151" }
{ "_id" : "TuxxBbg", "password" : "AHZSGT8/P2Y/P3t2Pz92Pw==,-1744417958" }
{ "_id" : "hVmpgBH", "password" : "Bz9NPz8/BT1fPz9vPycyPw==,-1061605774" }
{ "_id" : "DkMoXGp", "password" : "Pz8/Pz8/Pzc0aT86Bw==,-1986698939" }
{ "_id" : "TujqoWs", "password" : "P3RYPzVaPz8/AT8/N11bPw==,1656575205" }
{ "_id" : "vlbPUxU", "password" : "Pz95NnpTZT9HPz8/RFM/,-1163517437" }
{ "_id" : "ZhTQaYs", "password" : "His/ZD99Az8/Pz9/TU0/,366256945" }
{ "_id" : "SAdavdF", "password" : "P1I/XTg/Jj8/Pz9PPz9U,1932303339" }
>
```

Session Collection:



```
> db.sessions.find()
{ "_id" : "wgEmgCGWX2248yDMs6Z0Nka2Idx9kRV5s8rJZ+pggmI=", "username" : "58e17dfbc9bb971b6c825a73" }
{ "_id" : "d1jcoeeVLHH6e1LzrCZkV3jrvf5ICC7Bd1AbIDEKJcg=", "username" : "rVuFGWX" }
{ "_id" : "Tkw3BQgacF6ObVIIXn6ywBU4sAsoafoO+Kx3VPSPX8Q=", "username" : "RBNeBVo" }
{ "_id" : "BL3BpCxbI5kcoQyZdNbMJ8idPyPou4iW4h1be4r9nrI=", "username" : "EJKrexj" }
{ "_id" : "yLeLFaaMdEOdtGOkHdBXI0ai3Ghq3eO+vJTCVpPxkmw=", "username" : "MvBAotH" }
{ "_id" : "7R0J69wOLQR6bluA2D0DX/eJSyHcoMj+h9+ZnjwGFMY=", "username" : "MvBAotH" }
{ "_id" : "AvAi2exW3iAMSgF30nAZKF4cMHolnJuOsywkGeqEEBk=", "username" : "NpahcEF" }
{ "_id" : "qNfOII6gBneQHG4FXPtKpsAgloCjaU4exe8DoVUJS0U=", "username" : "NpahcEF" }
{ "_id" : "ate1mI1WPmIgve5diNe+FMeilGPQEyuNSFieusRy0CM=", "username" : "aYTEffk" }
{ "_id" : "Hxx2ZiKE8OYDolmQRlULLVKAPL5WKS84CyHXuw6fd80=", "username" : "aYTEffk" }
{ "_id" : "y6H9qU0WEcq6rG2KKSVvrqLrB+3AdMbAQDSucVqC9QY=", "username" : "makshay1" }
{ "_id" : "nv3Xmcv6HTr58kFQRHTjrT/mggpqOb7NkFW4lEdjamA=", "username" : "mqjmoSX" }
{ "_id" : "2GJC9ZzvOdKGOz47RY3Bpuzp3fjKQxBpGitRUdqSNjk=", "username" : "mqjmoSX" }
{ "_id" : "rQjQPwCYO34s42mj4AU61AZxrLG97kdhBywgOBX8SvE=", "username" : "TuxxBbg" }
{ "_id" : "7Se07OFWJ6wnYvfpWAagH42Ap5v/uzQk4QaLYyqW6pM=", "username" : "TuxxBbg" }
{ "_id" : "0fW0aA0ejrbuVxUJHafkjmWcEh2Nk3JVsiWJyflt3gA=", "username" : "hVmpgBH" }
{ "_id" : "tF5H8+BiP+bhd3X76DY7IHEIKgTv4blKVtOyew5LPCM=", "username" : "hVmpgBH" }
{ "_id" : "HLmW8l0VBRCacXUzKvz2w6ijIUG5v1HyCNT56mmCJ3g=", "username" : "DkMoXGp" }
{ "_id" : "VhL8NWqaB6Moq+GfTRANwYn0/MjWVuDbXaeXSCaZkV8=", "username" : "DkMoXGp" }
{ "_id" : "JHqIVsOzEEU1VrK1kUxgGhX0e+pnL8Rw7SE4hiBPpsQ=", "username" : "TujqoWs" }
Type "it" for more
>
```

Posts table:



```
> db.posts.aggregate([{$sort:{date:-1}},{$limit:2}]).pretty()
{
        "_id" : ObjectId("590f708f0ffb6630dc1acea3"),
        "title" : "New Testing",
        "author" : "makshay",
        "body" : "This is body of blog entry.",
        "permalink" : "new_testing",
        "tags" : [
                "mongoDB",
                "blog",
                "posts"
        ],
        "comments" : [
                {
                        "author" : "Akshay Marewar",
                        "body" : "This is comment added.",
                        "email" : "marewarakshay@gmail.com",
                        "num_likes" : 1
                }
        ],
        "date" : ISODate("2017-05-07T19:07:59.329Z")
}
{
        "_id" : ObjectId("590f706c0ffb6630dc1acea2"),
        "title" : "Testing",
        "author" : "makshay",
        "body" : "Hi Testing the project. this is body of the blog.",
        "permalink" : "testing",
        "tags" : [
                "blog",
                "test",
                "project"
        ],
        "comments" : [ ],
        "date" : ISODate("2017-05-07T19:07:24.111Z")
}
>
```

I use Apache Maven tool for this project.

Maven is a build automation tool used primarily for Java projects. Maven addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies. Contrary to preceding tools like Apache Ant, it uses conventions for the build procedure, and only exceptions need to be written down. An XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain well-defined tasks such as compilation of code and its packaging. Maven projects are configured using a Project Object Model, which is stored in a pom.xml-file. This POM only defines a unique identifier for the project (coordinates) and its dependency on the JUnit framework. However, that is already enough for building the project and running the unit tests associated with the project. Maven accomplishes this by embracing the idea of Convention over Configuration, that is, Maven provides default values for the project's configuration.

Below is pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.mongodb</groupId>
    <artifactId>blog-final</artifactId>
    <version>1.0-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>org.mongodb</groupId>
            <artifactId>mongodb-driver</artifactId>
            <version>3.2.2</version>
        </dependency>
        <dependency>
            <groupId>com.sparkjava</groupId>
            <artifactId>spark-core</artifactId>
            <version>1.1.1</version>
        </dependency>

        <dependency>
            <groupId>org.freemarker</groupId>
            <artifactId>freemarker</artifactId>
            <version>2.3.19</version>
        </dependency>

        <dependency>
            <groupId>org.apache.commons</groupId>
            <artifactId>commons-lang3</artifactId>
            <version>3.1</version>
        </dependency>

        <dependency>
            <groupId>commons-codec</groupId>
            <artifactId>commons-codec</artifactId>
            <version>1.7</version>
        </dependency>
    </dependencies>
    <!-- Configure maven-compiler-plugin version. -->
    <build>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>2.3.2</version>
          <configuration>
            <source>1.5</source>
            <target>1.5</target>
          </configuration>
        </plugin>
      </plugins>
    </build>

</project>
```

6. FUTURE SCOPE:

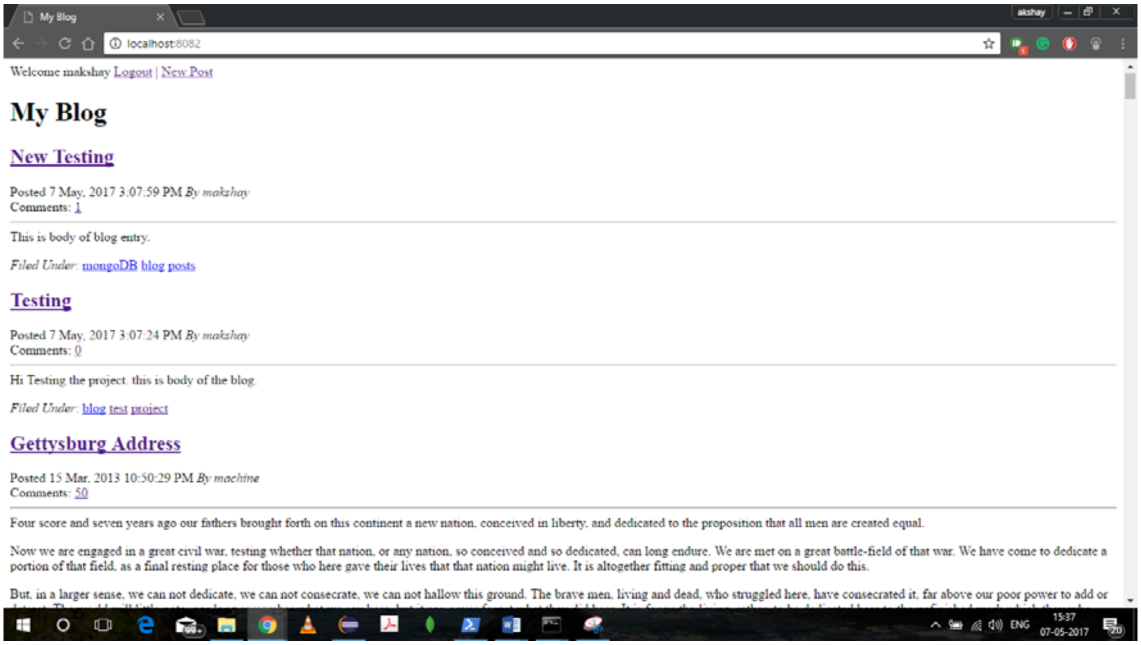
Following features can be added to:

- Upload or embed pictures and multimedia.
- Search blog entries using keywords.
- Delete our own comments.
- Configure the appearance and layout.

6. PROJECT SUMMARY:

MyBlog is a simple application is to express yourself and share your passion, to share your knowledge, to refine the writing skills, to build the professional network. It provides simple application interface which can help applications to interact with the user.

Below is the screenshot of the application:



Welcome makshay Logout | New Post

My Blog

New Testing

Posted 7 May, 2017 3:07:59 PM *By makshay*
Comments: 1

This is body of blog entry.

Filed Under: mongoDB blog posts

Testing

Posted 7 May, 2017 3:07:24 PM *By makshay*
Comments: 0

Hi Testing the project. this is body of the blog.

Filed Under: blog test project

Gettysburg Address

Posted 15 Mar, 2013 10:50:29 PM *By machine*
Comments: 50

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation, so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we can not dedicate, we can not consecrate, we can not hallow this ground. The brave men, living and dead, who struggled here, have consecrated it, far above our poor power to add or

7. BIBLIOGRAPHY:

References:

http://stackoverflow.com/

https://docs.mongodb.com/manual/release-notes/3.4/

https://docs.mongodb.com/ecosystem/drivers/java/

https://spark.apache.org/docs/0.9.1/java-programming-guide.html