**PROJECT REPORT**

**Abstract**

The aim of this project is to develop an online banking application, which allow tellers to provide login/logout interface for customers, create or edit customer’s information, create account for customers, deposit or withdraw money from the accounts, transfer funds between the accounts, check the balance etc. For handling crashes, there is a feature of load sharing, where if a server gets crashed, due to some reasons, all the data from the database can be safely migrated to another server without affecting client works. This ensures that the client side can work proficiently, even during server problems.

1. **INTRODUCTION**

Internet today, has reach to an unbelievable level of intelligence. Many of the tedious works of past have now, become easy due to presence of world wide web. Online banking is a great example of one of the advances of internet. Normal banking is not enough when it comes to internet. With normal banking, customers could find difficulties in accessing their accounts, especially when travelling to a places where ATMs are not available or available at long distances. Also, the processes to be followed are cumbersome and time-consuming, like in order to transfer funds from one account to another. Such headaches of traditional banking could be overcome by online banking. Customers will be happy because they can access their accounts anywhere easily if they have internet. The project is intended to provide online banking facilities such as checking accounts and making transfers over World Wide Web. Customer is supposed to provide login id and password to access his bank account online. With proper authorization the customer will be allowed to perform various banking activities.

There are two major ends in this project the customer end and the administrator end.

• Customer end is the end where customers log in, view profile, make transfers and also view transactions.

• Administrator end is responsible for validating the user i.e. not allowing any unauthorized person to access the account.

**1.2 Motivation**

The online banking application is quite appealing and idea of implementing the server and client application encouraged us to work towards the implementation. We were puzzled thinking about how to implement the system, logic to redirect the server if the current server is failed or crashed. We thought of trying to develop our online banking application in the distributed environment using the technology Java, Apache Tomcat server and MySQL database.

**1.3 Project Purpose**

The main purpose of Online Banking Solution is to provide customers with an ability to make transactions online over a very user friendly interface.

Facilities provided:

* Create account
* Login
* Check/view Account
* Deposit money
* Withdraw money
* Transfer funds
* Logout

1. **SYSTEM DESIGN**
   1. **MODULE DESCRIPTION**

* The will be used by two different types of users: customers and administrators. It will provide a different interface for each of the two types of users.
* ***Customer interface***
* *Create account:* This feature makes customer register with necessary details.
* *Deposit amount:* Deposit money and view in account. Hence, at each deposit, the account money gets summed up.
* *Withdraw amount:* Withdraw money from existing amount. If amount to withdraw is more than the amount in the account, then, it should not be allowed.
* *Check balance:* Here, a customer can view the information of the accounts that he/she holds with the bank.
* *Transfer amount:* This feature will allow a customer to transfer money from any of their accounts to another account.
* *Show transaction:* This feature will allow a customer to view the details of all transactions performed on any of their accounts held in the bank.
* *Close account:* This enables a customer to terminate his/ her account.
* *Logout:* A customer can end a session using logout.
* ***Administrator interface***
* *Add customer:* This feature will allow an administrator to create new accounts for users and assign role to them. In other words, we can say that the admin authorizes a registered customer to use the website.
* *Manage users:* Allows administrator to manage the user. Here, the admin maintains the database to store all the information.
* *Modify user/ account information:* As the admin maintains database, admin can authorize the information.

**2.2 Product Perspective**

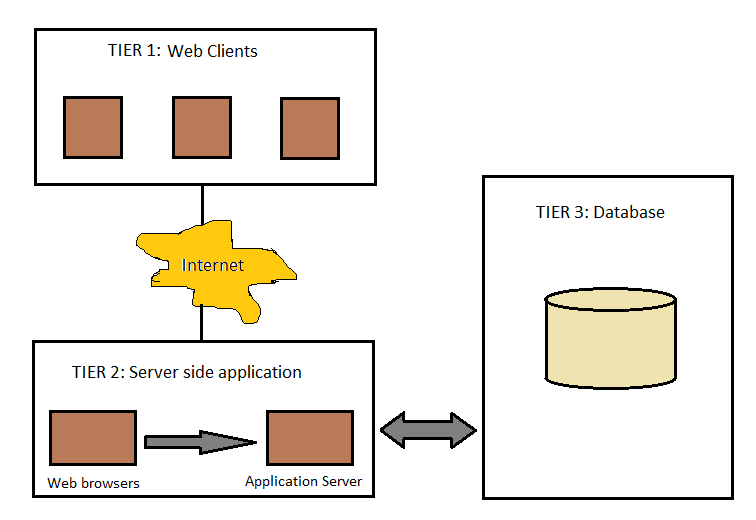
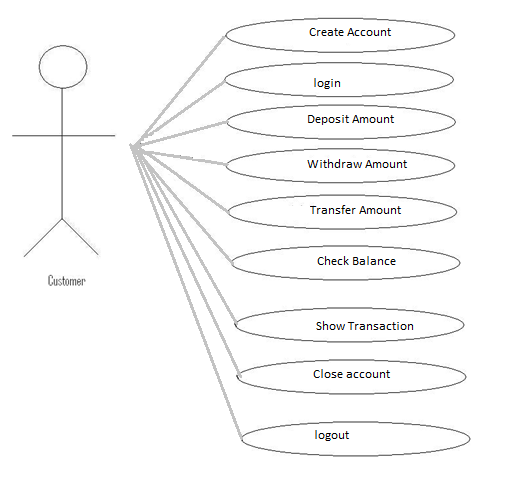


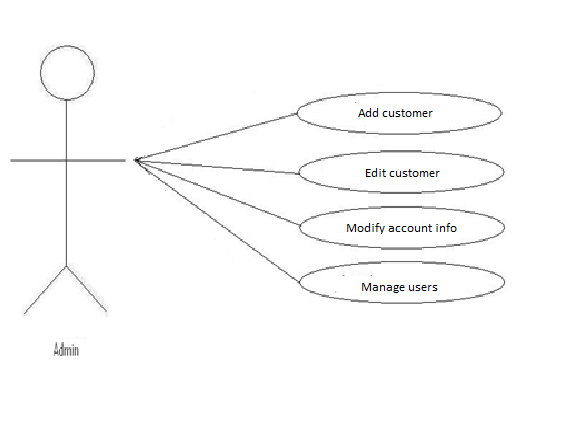
Figure 1: 3-Tier Architechture

* 1. **USECASE DIAGRAM**

FOR CUSTOMER



FOR ADMIN



**2.3 Sequence Diagram:**

Figure, shows the sequence diagram of online banking system. Here, The user first needs to register. It is a one time process where the user gives his/ her details to store in the database. From next time onwards, the user needs to login. During login, the user gives username and password which, the admin checks from the database. If the username and password are true, it is a successful login, else a failure. Once, logged in, the user can check the balance, from database, via the admin. To deposit the funds, user requests the admin to do so. The admin updates the database with new fund in the account and replies success to the user. In case of withdrawal, the user makes request to the admin. Admin checks the database to see if there is enough fund to withdraw from. If yes, the withdrawal is successful, if no, the withdrawal is declined. Similarly, for making transfers, the user request is validated by the admin. If the transfer is valid, it is allowed, else declined. Finally, the session can be ended by logging out.

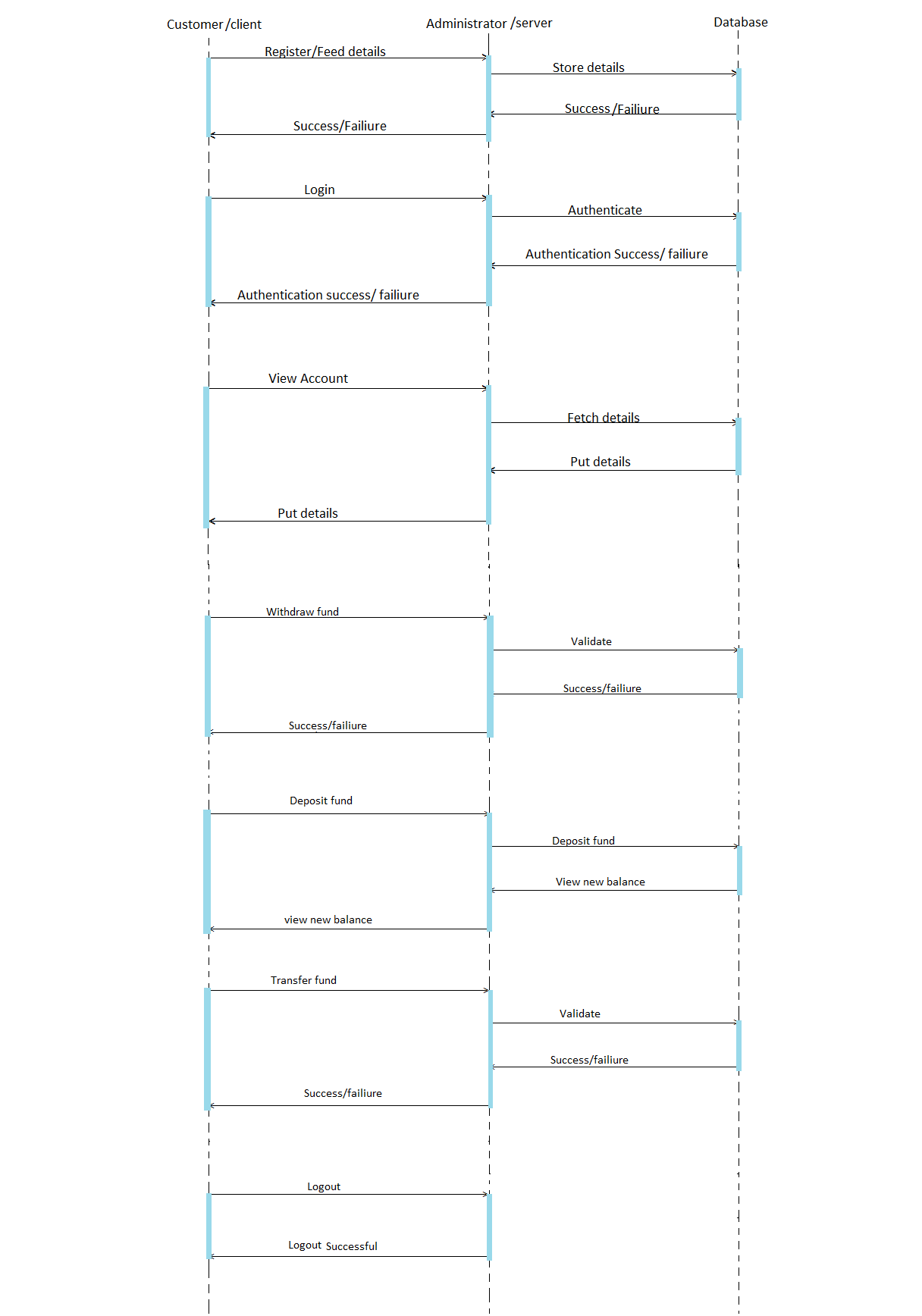


Figure:

**3. Tools to use for Implementation:**

**3.3.1 JSP**

**Java Server Pages** (**JSP**) is a technology that helps [software developers](http://en.wikipedia.org/wiki/Software_developer) create   
[dynamically generated web pages](http://en.wikipedia.org/wiki/Dynamic_web_page) based on [HTML](http://en.wikipedia.org/wiki/HTML), [XML](http://en.wikipedia.org/wiki/XML), or other document types[1]. Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlet at runtime; each JSP servlet is cached and re-used until the original JSP is modified. JSP can be used independently or as the view component of a server-side-model–view–controller-design, normally with JavaBeans as the model and Java servlets (or a framework such asApache Struts) as the controller. This is a type of Model 2 architecture. JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, use Java byte code rather than a native software format. Like any other Java program, they must be executed within a Java virtual machine (JVM) that integrates with the server's host operating system to provide an abstract platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of Output Stream, they can deliver other types of data as well. [4]

The Web container creates JSP implicit objects like pageContext, servletContext, session, request & response.

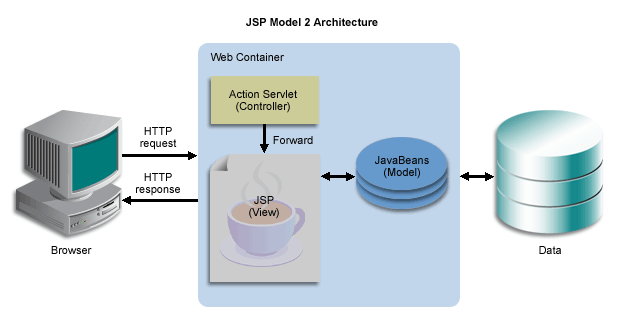


Figure: Model 2 architecture of JSP

**3.3.2 Servlet**

The **servlet** is a [Java programming language](http://en.wikipedia.org/wiki/Java_programming_language) [class](http://en.wikipedia.org/wiki/Class_(computer_programming)) used to extend the capabilities of a [server](http://en.wikipedia.org/wiki/Server_(computing)). Although servlets can respond to any types of requests, they are commonly used to extend the applications hosted by [web servers](http://en.wikipedia.org/wiki/Web_server), so they can be thought of as [Java applets](http://en.wikipedia.org/wiki/Java_applet) that run on [servers](http://en.wikipedia.org/wiki/Server_(computing)) instead of in [web browsers](http://en.wikipedia.org/wiki/Web_browser).

Servlets are often used for below applications:

* Process or store data that was submitted from an HTML form.
* Provide dynamic content such as the results of a database query.
* Manage state information that does not exist in the stateless HTTP protocol.

So Technically, a servlet is a Java Class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes which respond to requests. Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP protocol. Thus servlet is often used as shorthand for HTTP servlet.

**3.3.3 Tomcat**

**Apache Tomcat** is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, and provides a "pureJava"HTTP web server environment for Java code to run in. In the simplest config Tomcat runs in a single operating system process. The process runs a Java virtual machine (JVM). Every single HTTP request from a browser to Tomcat is processed in the Tomcat process in a separate thread.

Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

**4. Overview Of the project:**

1. Login page

This is the first page of our banking application, In which if the user has already registered his/her self into the system then they can just provide user ID and password and can login into the system, but if they are new to the system, then they can just click ‘Register user’ and can register their selves to utilize the system.

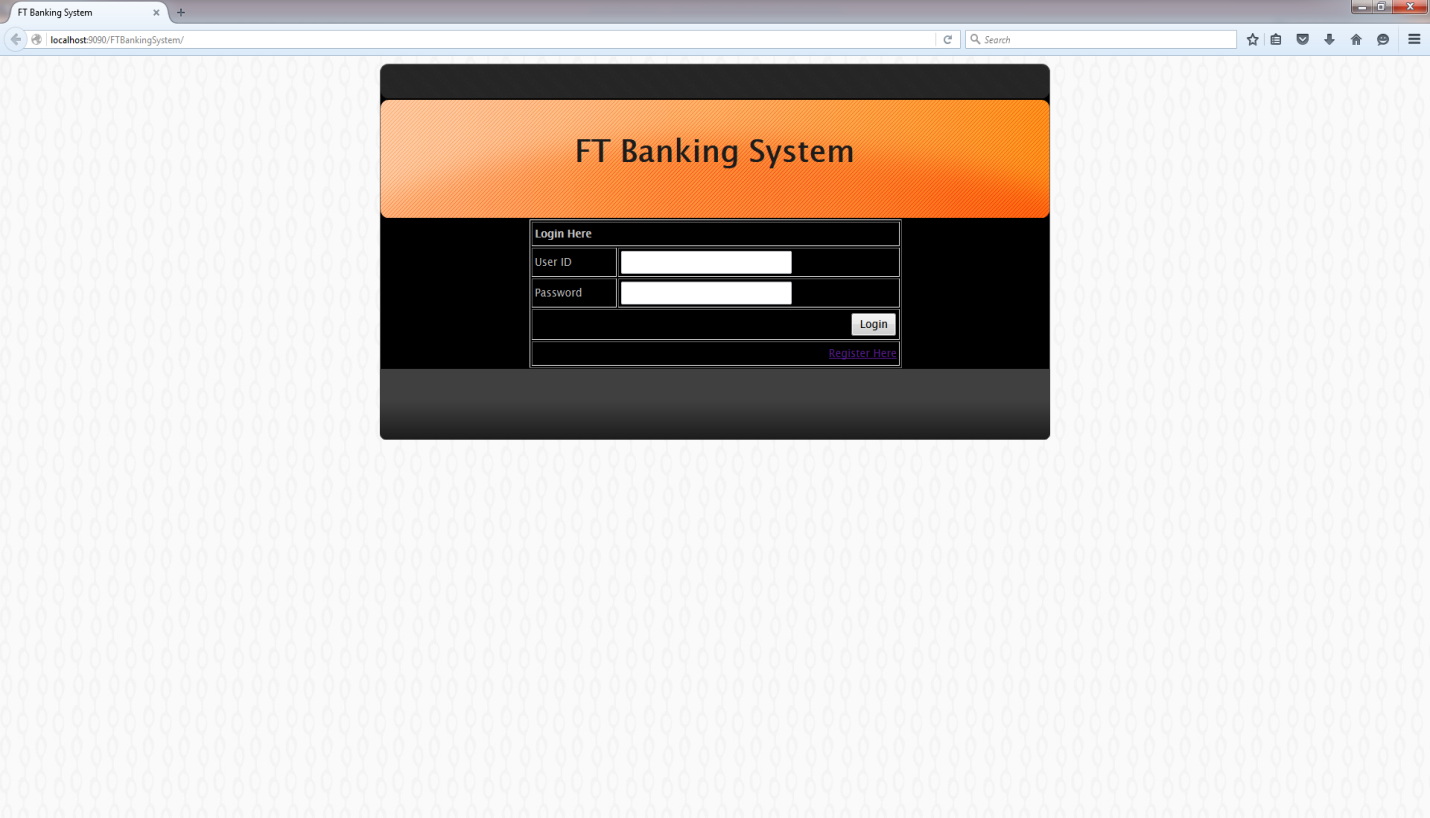


Figure 1: Index page of the web site.

1. Registration Page:

The second page is the registration page, when user clicks on “Register Here” in the login page, it gets redirected to this page in which user have to provide basic information like first name, last name, desired user ID and password and then need to click on submit.

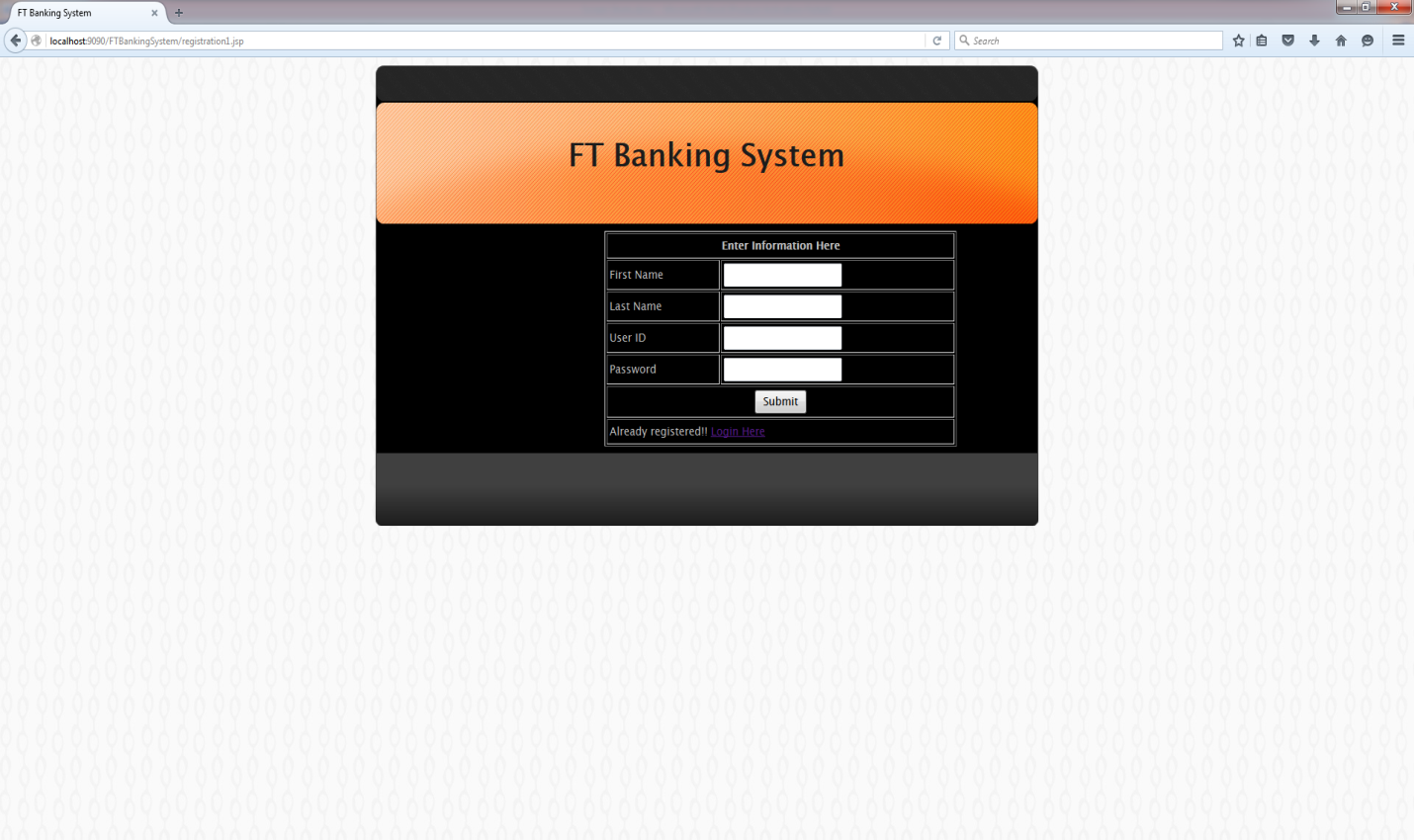


Figure 2: Registration Page

1. Application homepage:

After user successfully login into the system, it redirects it to the systems homepage where on the left side of the page user can see the options for transfer funds, deposit amount, withdraw amount, check balance, close account etc.

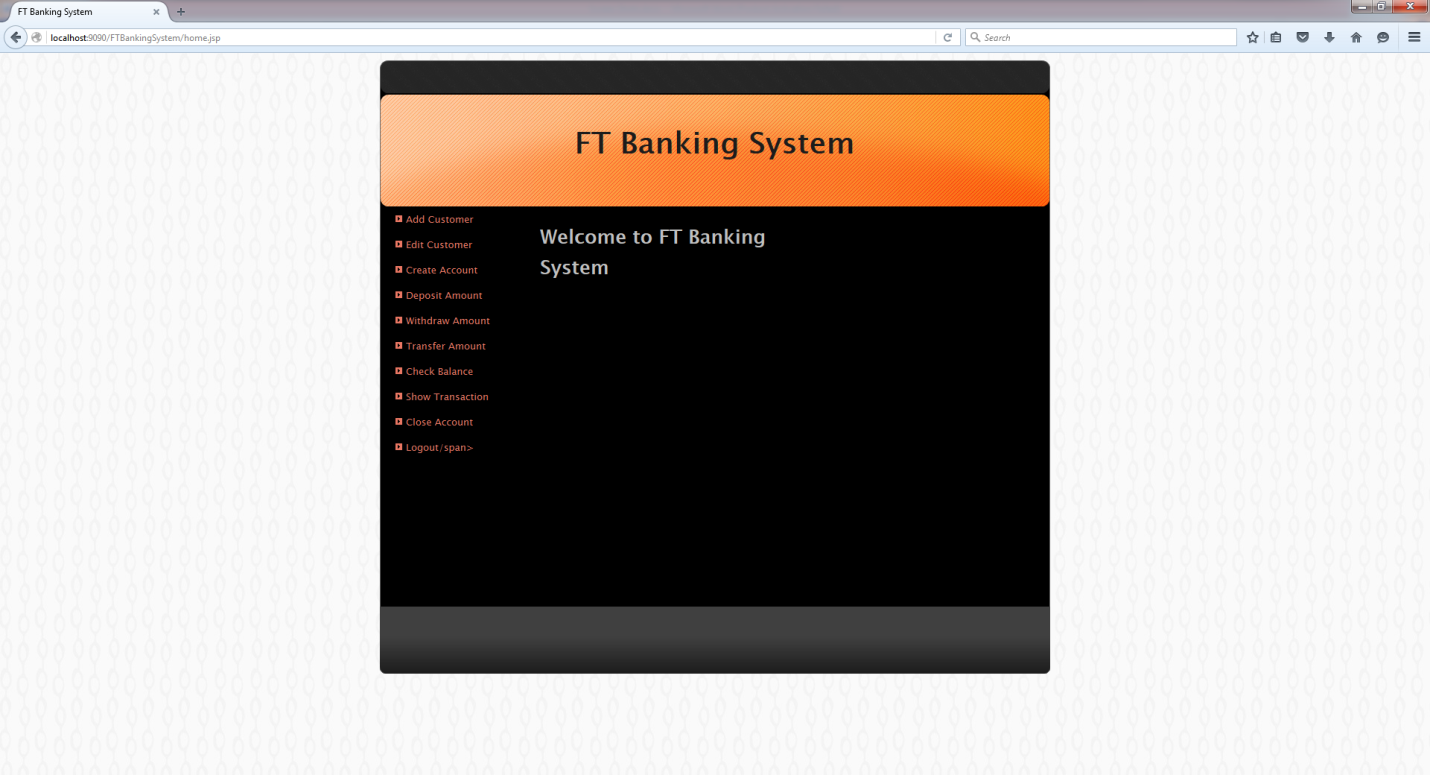


Figure 3: Application Homepage