CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Governors State University

Governors State University
OPUS Open Portal to University Scholarship

All Capstone Projects | Student Capstone Projects

Spring 2013

File Transfer Web Application

Madhavi Dintakurthy
Governors State University

Follow this and additional works at: http://opus.govst.edu/capstones

 Part of the Computer Sciences Commons

Recommended Citation
Dintakurthy, Madhavi, "File Transfer Web Application" (2013). *All Capstone Projects*. 60.
http://opus.govst.edu/capstones/60

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to
http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the Governors State Computer Science Department
This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

FILE TRANSFER WEB APPLICATION

By

Madhavi Dintakurthy

[redacted]

PROJECT DOCUMENTATION

For the Degree of Master of Science,

With a Major in Computer Science

Governors State University

University Park, IL 60484

2013

Abstract

Aim is to develop a web application using java web framework.

Description: Develop a web application in 3 tier architecture involving user interface, controller and database. The user interface will be a web page hosted on a server. The web page consists of both static and dynamic content. All the data required for the application is stored in database tables. Controller accesses the data from the database and provides it to the user through user interface (web page).

- ➔ Web page: The webpage serves as user interface to the application. Web page is developed using **Jsp**, **HTML**, **Java Script** and **Ajax**.
- ➔ Controller: The controller serves as the communicating interface between web page and database. Controller serves the data to the user based on the user request. Controller is developed using **Java 7**, **Java Servlets**, and **JDBC**.
- ➔ Database: Database is used to store all the required data for the application. Database tables are designed according to requirement and developed using **SQL**.
- ➔ Server: Server is used to host the application. The developed web application is deployed onto server. **Apache Tomcat** is the application server for the application.

Environment: Java 7, HTML, Java Script, JDBC, Ajax, Apache Tomcat, Derby Database, SQL.

CONTENTS

ABSTRACT

1. INTRODUCTION …….. 5

2. SOFTWARE PROCESS MODEL …….. 6

2.1 INTRODUCTION
2.2 WATERFALL MODEL

3. TECHNOLOGIES …… 8

4. SOFTWARE DESCRIPTION ……. 9

4.1 WEB PAGE
4.1.1 JSP
4.1.2 HTML
4.1.3 JAVA SCRIPT
4.1.4 AJAX
4.2 CONTROLLER
4.2.1 JAVA
4.2.2 JAVA SERVLETS
4.2.3 JDBC
4.3 DATABASE
4.3.1 SQL SERVER
4.4 SERVER
4.4.1 APCHE TOMCAT

5. SOFTWARE INTERFACE …… 29

6. DATABASE TABLES ……. 33

7. SYSTEM TESTING AND IMPLEMENTATIONS …… 34

7.1 STRATEGIC APPROACH OF SOFTWARE TESTING

BIBLIOGRAPHY ……. 36

1. Introduction

File Transfer Web Application is used to upload any type of files like pdf, mp3, word, video, etc.) into a database table and can download any type of files from the database.

This web application is developed using java web framework.

The web application is developed in 3 tier architecture involving user interface, controller and database. The user interface will be a web page hosted on a server. The web page consists of both static and dynamic content. All the data required for the application is stored in database tables. Controller accesses the data from the database and provides it to the user through user interface (web page).

User Interface:

The File Transfer web application contains four web pages like Home page, Upload page, download Page and About Us page. On every page there are four buttons named Home, Upload, download and AboutUs, and the user can click on any of the button to go to that particular page. The Home page and AboutUs page contains details about the project. In the Upload page, the user can upload any type of files into a database table by selecting any file from the computer using ‘choose a file’ button and once the user click on the upload button the file will be uploaded to a database table. The Download page contains all the files that are in the database table. When the user clicks on a particular file that is displayed on a Download page the file will be downloaded into the computer. In this way the user can upload and download files using File Transfer Web Application.

2. Software Process Model

2.1 Introduction:

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution. The model that is basically being followed is the WATER FALL MODEL.

2.2 Waterfall Model:

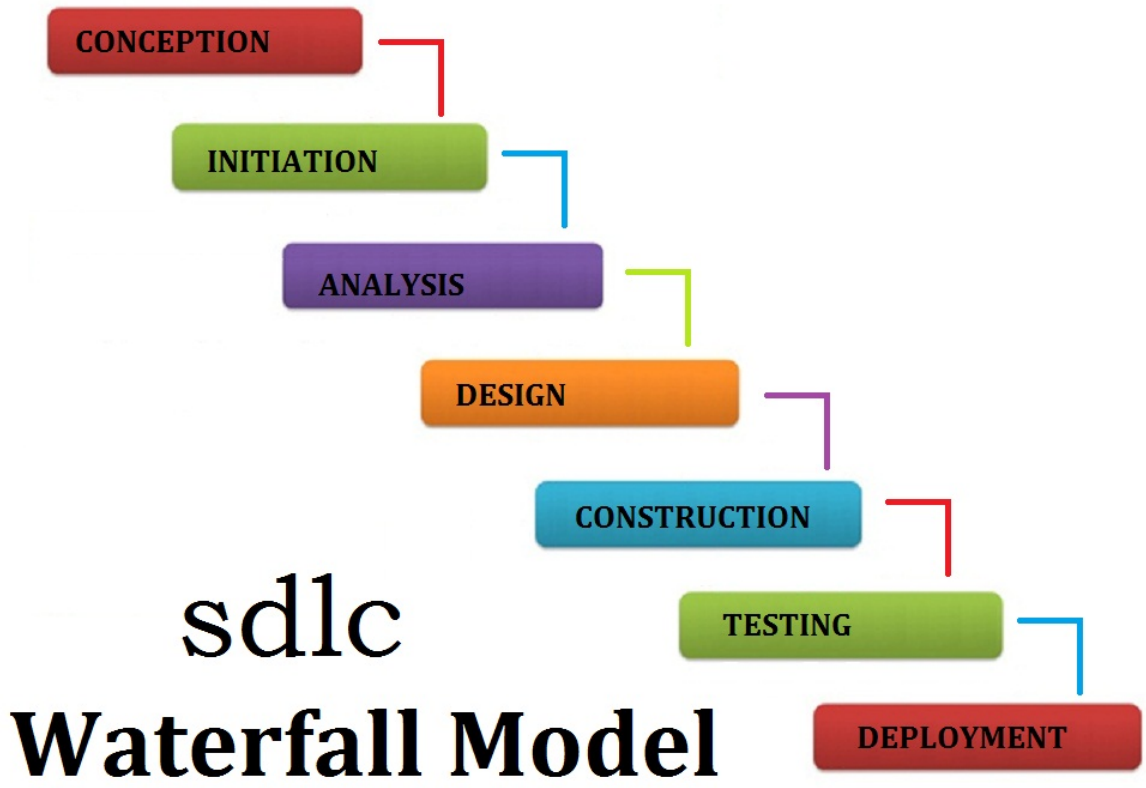
The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirement. Therefore, different projects may follow different approaches to handle such situations.

What is Waterfall methodology?

The waterfall approach is the earliest approach that was used for software development. Initially, most projects followed the waterfall approach because they did not focus on changing requirements.

Waterfall Approach Phases

a. Conception: Triggers when a problem is perceived. This phase involves identifying goals to be achieved after the problem is solved, estimating benefits in the new system over the current system, and identifying other areas that are affected by the solution. This phase also involves and developing the business case for the project. A business case provides the information that a manger needs to decide whether to support a proposed project, before resources are committed to its development.



b. Initiation: Involves a macro level study of the customer requirements. This phase also involves defining alternative solutions to the customer requirements and cost-benefit justification of these alternatives.

c. Analysis: Involves carrying out detailed study of the customer requirements and arriving at the exact requirements of the proposed system. The phase involves freezing the requirements before the design phase begins.

d. Design: Involves translating the identified requirements into a logical structure, called design that can be implemented in a programming logic.

e. Construction: Involves integrating and testing all the modules developed in the previous phase as a complete system.

f. Integration and Testing: Involves integrating and testing all the modules developed in the previous phase as a complete system.

g. Implementation and maintenance: Involves converting the new system design into operation. This may involve implementing the software system and training the operating staff before the software system is functional.

The waterfall approach assumes that requirements are stable and frozen across the project plan. However, this is usually not true in case of large projects where requirements may evolve across the development process.

3. TECHNOLOGIES

3.1 Software Requirements

Web Technology : Eclipse IDE

Web Server : Tomcat Server

Database : SQL Server 2012

Code Behind : Java, J2EE, JDBC, ODBC, JavaScript, Html, Ajax, Sql

3.2 Hardware Requirements

Intel Pentium : 600 MHz or above.

RAM (SD/DDR) : 512MB

Hard Disc : 30GB or above

System bus : 32 bits

RAM : 256MB of RAM

Monitor : SVGA COLOR

Keyboard : 108 keys

Mouse : 2 button mouse

4. Software Description

4.1 Web page: The webpage serves as user interface to the application. Web page is developed using **Jsp, HTML, Java Script** and **Ajax.**

4.1.1 JSP:

Java Server Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. Released in 1999 by Sun Microsystems, JSP is similar to PHP, but it uses the Java programming language.

To deploy and run JavaServer Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required.

Java Server Pages (JSP) is a technology that lets you mix regular, static HTML with dynamically-generated HTML. Many Web pages that are built by CGI programs are mostly static, with the dynamic part limited to a few small locations. But most CGI variations, including servlets, make you generate the entire page via your program, even though most of it is always the same. JSP lets you create the two parts separately. Here's an example:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Welcome to Our Store</TITLE></HEAD>
<BODY>
<H1>Welcome to Our Store</H1>
<SMALL>Welcome,
<!-- User name is "New User" for first-time visitors -->
<% out.println(Utils.getUserNameFromCookie(request)); %>
To access your account settings, click
<A HREF="Account-Settings.html">here.</A></SMALL>
<P>
Regular HTML for all the rest of the on-line store's Web page.
</BODY></HTML>
```

4.1.2 HTML:

HyperText Markup Language (HTML) is the main markup language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags, known as empty elements, are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content.

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML markup.

4.1.3 Java Script:

The JavaScript programming language, developed by Netscape, Inc., is not part of the Java platform.

JavaScript (JS) is an interpreted computer programming language. It was originally implemented as part of web browsers so that client-side scripts could interact with the user,

control the browser, communicate asynchronously, and alter the document content that was displayed.

JavaScript is a prototype-based scripting language that is dynamic, weakly typed, and has first-class functions. Its syntax was influenced by the language C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

JavaScript's use in applications outside of web pages for example, in PDF documents, site-specific browsers, and desktop widgets is also significant. Newer and faster JavaScript VMs and frameworks built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications.

JavaScript was formalized in the ECMAScript language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to computational objects within a host environment.

JavaScript does not create applets or stand-alone applications. In its most common form today, JavaScript resides inside HTML documents, and can provide levels of interactivity to web pages that are not achievable with simple HTML.

Listed are key differences between the Java and JavaScript:

- Java is an OOP programming language while Java Script is an OOP scripting language.
- Java creates applications that run in a virtual machine or browser while JavaScript code is run on a browser only.
- Java code needs to be compiled while JavaScript code are all in text.
- They require different plug-ins.

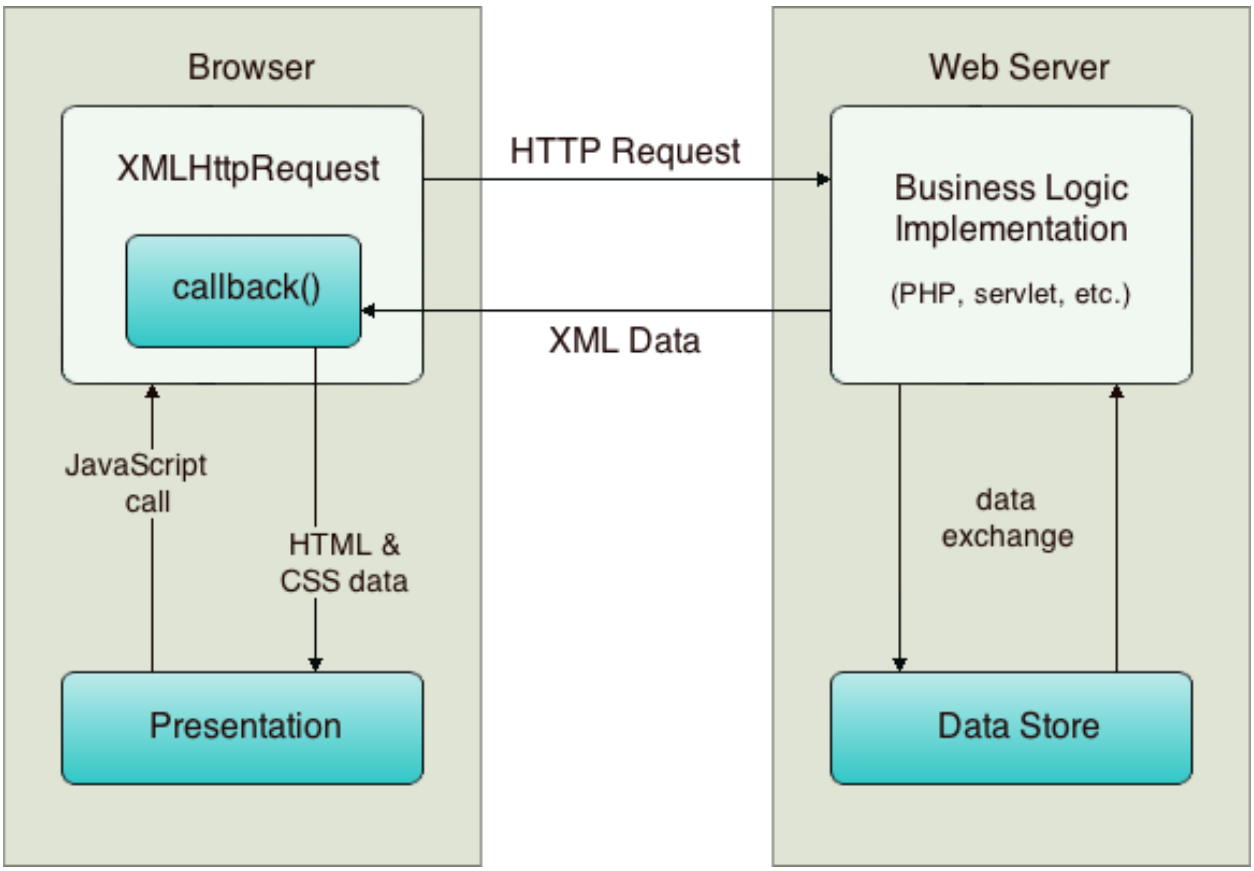
4.1.4 Ajax:

AJAX = Asynchronous JavaScript and XML.

AJAX is not a new programming language, but a new way to use existing standards.

AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.

Implementing auto-completion in a search field is something that can be performed using Ajax. Ajax works by employing an XMLHttpRequest object to pass requests and responses asynchronously between the client and server. The following diagram illustrates the process flow of the communication that takes place between the client and server.



The process flow of the diagram can be described by the following steps:

The user triggers an event, for example by releasing a key when typing in a name. This results in a JavaScript call to a function that initializes an XMLHttpRequest object.

The XMLHttpRequest object is configured with a request parameter that includes the ID of the component that triggered the event, and any value that the user entered. The XMLHttpRequest object then makes an asynchronous request to the web server.

On the web server, an object such as a servlet or listener handles the request. Data is retrieved from the data store, and a response is prepared containing the data in the form of an XML document.

Finally, the XMLHttpRequest object receives the XML data using a callback function, processes it, and updates the HTML DOM (Document Object Model) to display the page containing the new data.

This document demonstrates how to construct the auto-complete scenario by following the process flow indicated in the above diagram. You first create the client-side files for the presentation and functionality needed to generate the XMLHttpRequest object. Then, you set up the server-side by creating the data store and business logic using Java-based technology. Finally, you return to the client-side and implement callback(), and other JavaScript functionality to update the HTML DOM.

4.2 Controller: The controller serves as the communicating interface between web page and database. Controller serves the data to the user based on the user request. Controller is developed using **Java 7**, **Java Servlets**, and **JDBC**.

4.2.1 Java:

This project is implemented using Java. Java goes back to 1991 when a group of sun engineers led by James Gosling , wanted to design a small computer language that could be used for consumer devices and named it as Green Project. Their idea was to develop a

portable language that could generate intermediate code for virtual machines. This intermediate code then can be used on any machines that has the correct interpreter.

Java is a programming language that lets us to do almost anything we can do with traditional programming language for distributed applications. It is platform in-dependent and having a lot of networking features included within it. A java program can run equally well on any architecture that has a java interpreter.

FEATURES OF JAVA

a) Encapsulation

Data Encapsulation is one of the most sticking features of OOP's. Encapsulation is the wrapping up of data and function into single unit called class. The wrapped defines the behaviour and protects the code and data from being arbitrarily accessed by the outside world and only those function which are wrapped in the class can access it. This type of insulation of data from direct access by the program is called data hiding.

b) Inheritance

Inheritance is the process by which objects a class can acquire the properties of objects of another class i.e. In OOPs the concept of inheritance provides idea of reusability providing the means of adding additional features to an existing class without modifying it. This is possible by deriving a new class from the existing on thus the newly created class will have the combined features of both the parent and the child classes.

c) Object –Oriented

Almost everything in java is a clear, a method or an object. Only the most basic primitive operative and data types are at a sub-class level.

d) Data Abstraction

Data Abstraction is an act of representing essential features without including the background details and explanation.

e) Platform Independent

Java programs are compiled with a byte code format that can be read and run by interpreters on many platforms including Windows 95, Windows NT and later.

f) Multi-Thread

Java is inherently multi-threaded. A single java program can make many different things processing independently and continuously.

g) High Performance

Java can be compiled on the fly with a Just-in-time compiler (JIT) to code that rivals C++ in speed.

h) Safe

Java code can be executed in an environment that prohibits it from viruses, deleting or modifying files or otherwise performing data destroying and computer crashing operation.

i) Simple

Java has the bare bones functionally needed to implement its rich feature set.

Components

Java has several in-built components:

Javac : Compiler for java programs that could generate byte codes

Java : Interpreter to read and execute java byte codes.

Javap : To disassemble and debug the java bytecodes.

Javadoc: Document generator.

Javah : To write and link native codes with java programs.

4.2.2 Java Servlets:

Servlets are the Java platform technology of choice for extending and enhancing Web servers. Servlets provide a component-based, platform-independent method for building Web-based applications, without the performance limitations of CGI programs. And unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), servlets are server- and platform-independent. This leaves you free to select a "best of breed" strategy for your servers, platforms, and tools.

Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection.

Today servlets are a popular choice for building interactive Web applications. Third-party servlet containers are available for Apache Web Server, Microsoft IIS, and others. Servlet containers are usually a component of Web and application servers, such as BEA WebLogic Application Server, IBM WebSphere, Sun Java System Web Server, Sun Java System Application Server, and others.

You might want to check out the latest information on JavaServer Pages (JSP) technology. JSP technology is an extension of the servlet technology created to support authoring of HTML and XML pages. It makes it easier to combine fixed or static template data with dynamic content. Even if you're comfortable writing servlets, there are several compelling reasons to investigate JSP technology as a complement to your existing work.

4.2.3 JDBC

Driver - JDBC-ODBC bridge:

The JDBC-ODBC bridge, is a database driver implementation that employs the ODBC driver to connect to the database. The driver converts JDBC method calls into ODBC function calls.

The driver is platform-dependent as it makes use of ODBC which in turn depends on native libraries of the underlying operating system the JVM is running upon. Also, use of this driver leads to other installation dependencies; for example, ODBC must be installed on the computer having the driver and the database must support an ODBC driver. The use of this driver is discouraged if the alternative of a pure-Java driver is available. The other implication is that any application using a type 1 driver is non-portable given the binding between the driver and platform. This technology isn't suitable for a high-transaction environment. Type 1 drivers also don't support the complete Java command set and are limited by the functionality of the ODBC driver.

Sun provides a JDBC-ODBC Bridge driver: sun.jdbc.odbc.JdbcOdbcDriver. This driver is native code and not Java, and is closed source.

If has been written so that loading it causes an instance to be created and also calls DriverManager.registerDriver with that instance as the parameter (as it should do), then it is in the DriverManager's list of drivers and available for creating a connection.

It may sometimes be the case that more than one JDBC driver is capable of connecting to a given URL. For example, when connecting to a given remote database, it might be possible to use a JDBC-ODBC bridge driver, a JDBC-to-generic-network-protocol driver, or a driver supplied by the database vendor. In such cases, the order in which the drivers are tested is significant because the DriverManager will use the first driver it finds that can successfully connect to the given URL.

First the DriverManager tries to use each driver in the order it was registered. (The drivers listed in jdbc.drivers are always registered first.) It will skip any drivers that are untrusted code unless they have been loaded from the same source as the code that is trying to open the connection.

It tests the drivers by calling the method Driver.connect on each one in turn, passing them the URL that the user originally passed to the method DriverManager.getConnection. The first driver that recognizes the URL makes the connection.

Advantages

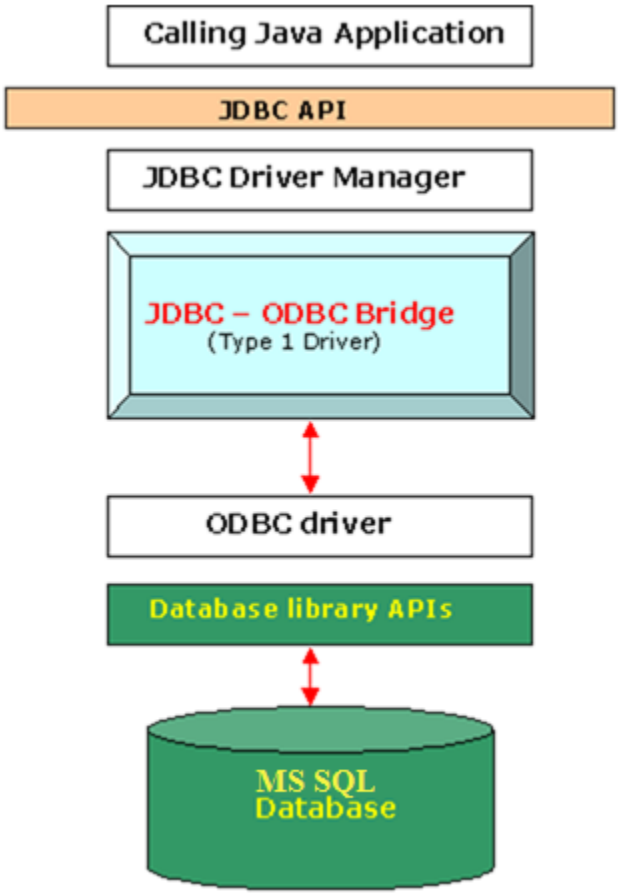
Almost any database, for which ODBC driver is installed, can be accessed.

Disadvantages

Performance overhead since the calls have to go through the jdbc Overhead bridge to the ODBC driver, then to the native db connectivity interface (thus may be slower than other types of drivers).

The ODBC driver needs to be installed on the client machine.

Not suitable for applets, because the ODBC driver needs to be installed on the client.



4.3 Database: Database is used to store all the required data for the application.

Database tables are designed according to requirement and developed using **SQL**.

4.3.1 SQL Server:

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

SQL SERVER TABLES

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

PRIMARY KEY

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

RELATIONAL DATABASE

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and

enables you to define relationships between the tables and enables you to define relationships between the tables.

FOREIGN KEY

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

REFERENTIAL INTEGRITY

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

DATA ABSTRACTION

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

Physical level: This is the lowest level of abstraction at which one describes how the data are actually stored.

Conceptual Level: At this level of database abstraction all the attributed and what data are actually stored is described and entries and relationship among them.

View level: This is the highest level of abstraction at which one describes only part of the database.

ADVANTAGES OF RDBMS

- Redundancy can be avoided
- Inconsistency can be eliminated
- Data can be shared
- Standards can be enforced
- Security restrictions can be applied
- Integrity can be maintained
- Conflicting requirements can be balanced
- Data independence can be achieved.

DISADVANTAGES OF DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

FEATURES OF SQL SERVER (RDBMS)

SQL SERVER is one of the leading database management systems (DBMS) because it is the only Database that meets the uncompromising requirements of today's most demanding information systems. From complex decision support systems (DSS) to the most rigorous online transaction processing (OLTP) application, even application that require simultaneous DSS and OLTP access to the same critical data, SQL Server leads the industry in both performance and capability

SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.

SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

SQL SERVER with transactions processing option offers two features which contribute to very high level of transaction processing throughput, which are

- The row level lock manager

ENTERPRISE WIDE DATA SHARING

The unrivaled portability and connectivity of the SQL SERVER DBMS enables all the systems in the organization to be linked into a singular, integrated computing resource.

PORTABILITY

SQL SERVER is fully portable to more than 80 distinct hardware and operating systems platforms, including UNIX, MSDOS, OS/2, Macintosh and dozens of proprietary platforms. This portability gives complete freedom to choose the database server platform that meets the system requirements.

OPEN SYSTEMS

SQL SERVER offers a leading implementation of industry standard SQL. SQL Server's open architecture integrates SQL SERVER and non –SQL SERVER DBMS with industry's most comprehensive collection of tools, application, and third party software products SQL Server's Open architecture provides transparent access to data from other relational database and even non-relational database.

DISTRIBUTED DATA SHARING

SQL Server's networking and distributed database capabilities to access data stored on remote server with the same ease as if the information was stored on a single local computer. A single SQL statement can access data at multiple sites. You can store data where system requirements such as performance, security or availability dictate.

UNMATCHED PERFORMANCE

The most advanced architecture in the industry allows the SQL SERVER DBMS to deliver unmatched performance.

SOPHISTICATED CONCURRENCY CONTROL

Real World applications demand access to critical data. With most database Systems application becomes "contention bound" – which performance is limited not by the CPU power or by disk I/O, but user waiting on one another for data access. SQL Server employs full, unrestricted row-level locking and contention free queries to minimize and in many cases entirely eliminates contention wait times.

NO I/O BOTTLENECKS

SQL Server's fast commit groups commit and deferred write technologies dramatically reduce disk I/O bottlenecks. While some database write whole data block to disk at commit time, SQL Server commits transactions with at most sequential log file on disk at commit time, On high throughput systems, one sequential writes typically group commit multiple transactions. Data read by the transaction remains as shared memory so that other transactions may access that data without reading it again from disk. Since fast commits write all data necessary to the recovery to the log file, modified blocks are written back to the database independently of the transaction commit, when written from memory to disk.

4.4 Server: Server is used to host the application. The developed web application is deployed onto server. **Apache Tomcat** is the application server for the application.

4.4.1 Apache Tomcat:

Apache Tomcat (or simply Tomcat, formerly also Jakarta Tomcat) is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run.

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

Components:

Tomcat 4.x was released with Catalina (servlet container), Coyote (an HTTP connector) and Jasper (a JSP engine).

Catalina:

Catalina is Tomcat's servlet container. Catalina implements Sun Microsystems' specifications for servlet and JavaServer Pages (JSP). In Tomcat, a Realm element represents a "database" of usernames, passwords, and roles (similar to Unix groups) assigned to those users. Different implementations of Realm allow Catalina to be integrated into environments where such authentication information is already being created and maintained, and then use that information to implement Container Managed Security as described in the Servlet Specification.

Coyote:

Coyote is Tomcat's HTTP Connector component that supports the HTTP 1.1 protocol for the web server or application container. Coyote listens for incoming connections on a specific TCP port on the server and forwards the request to the Tomcat Engine to process the request and send back a response to the requesting client.

Jasper:

Jasper is Tomcat's JSP Engine. Jasper parses JSP files to compile them into Java code as servlets (that can be handled by Catalina). At runtime, Jasper detects changes to JSP files and recompiles them.

As of version 5, Tomcat uses Jasper 2, which is an implementation of the Sun Microsystems's JSP 2.0 specification. From Jasper to Jasper 2, important features were added:

JSP Tag library pooling - Each tag markup in JSP file is handled by a tag handler class. Tag handler class objects can be pooled and reused in the whole JSP servlet.

Background JSP compilation - While recompiling modified JSP Java code, the older version is still available for server requests. The older JSP servlet is deleted once the new JSP servlet has finished being recompiled.

Recompile JSP when included page changes - Pages can be inserted and included into a JSP at runtime. The JSP will not only be recompiled with JSP file changes but also with included page changes.

JDT Java compiler - Jasper 2 can use the Eclipse JDT (Java Development Tools) Java compiler instead of Ant and javac.

Three new components were added with the release of Tomcat 7:

Cluster:

This component has been added to manage large applications. It is used for Load balancing that can be achieved through many techniques. Clustering support currently requires the JDK version 1.5 or later.

High availability:

A high-availability feature has been added to facilitate the scheduling of system upgrades (e.g. new releases, change requests) without affecting the live environment. This is done by dispatching live traffic requests to a temporary server on a different port while the main server is upgraded on the main port. It is very useful in handling user requests on high-traffic web applications.

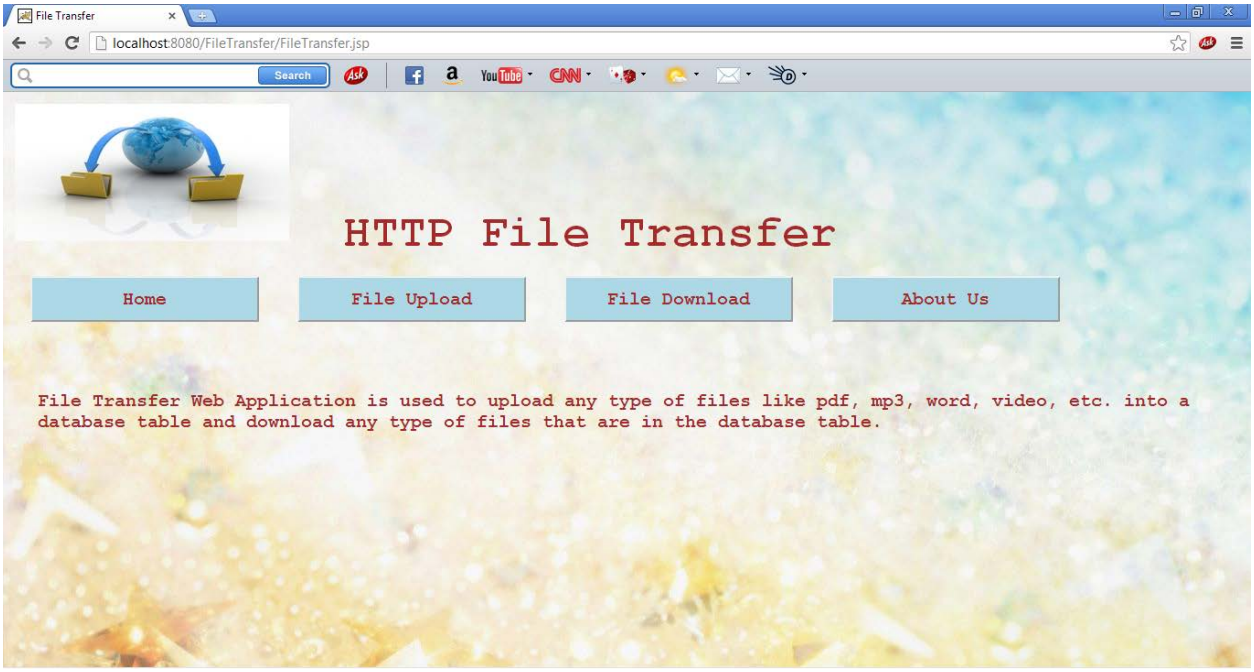
Web Application:

It has also added user as well as system based web applications enhancement to add support for deployment across the variety of environments. It also tries to manage session as well as applications across the network.

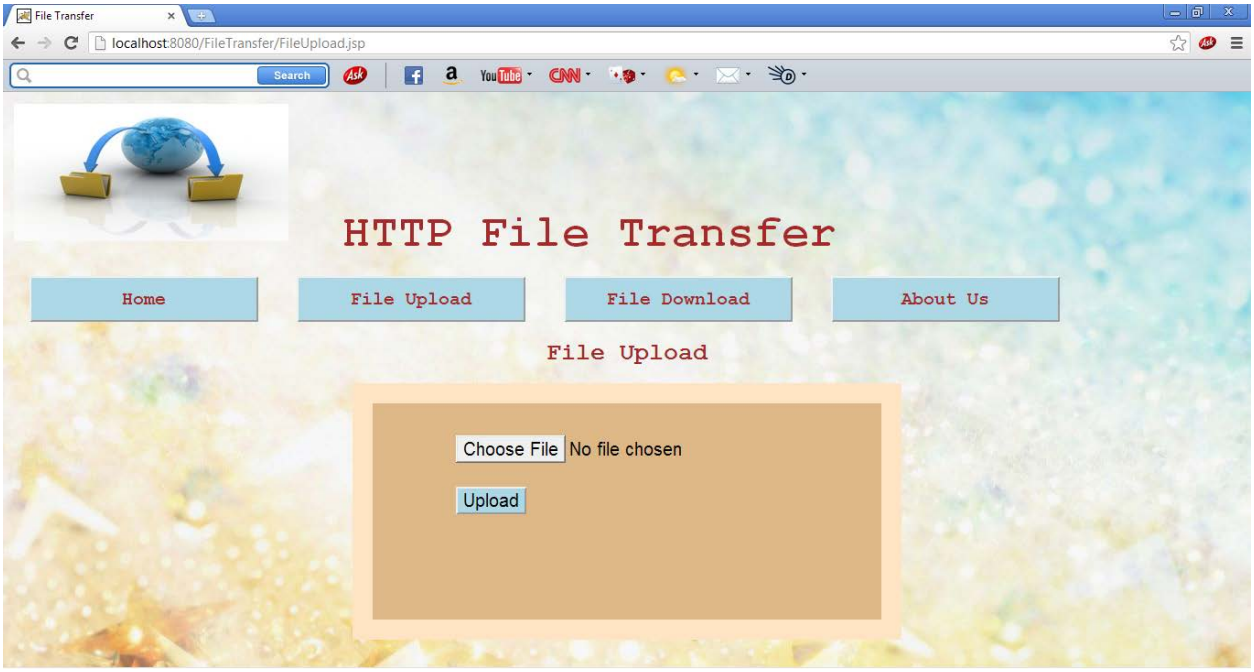
Tomcat building is additional components. A number of additional components may be used with Apache Tomcat. These components may be built by users should they need them or they can be downloaded from one of the mirrors.

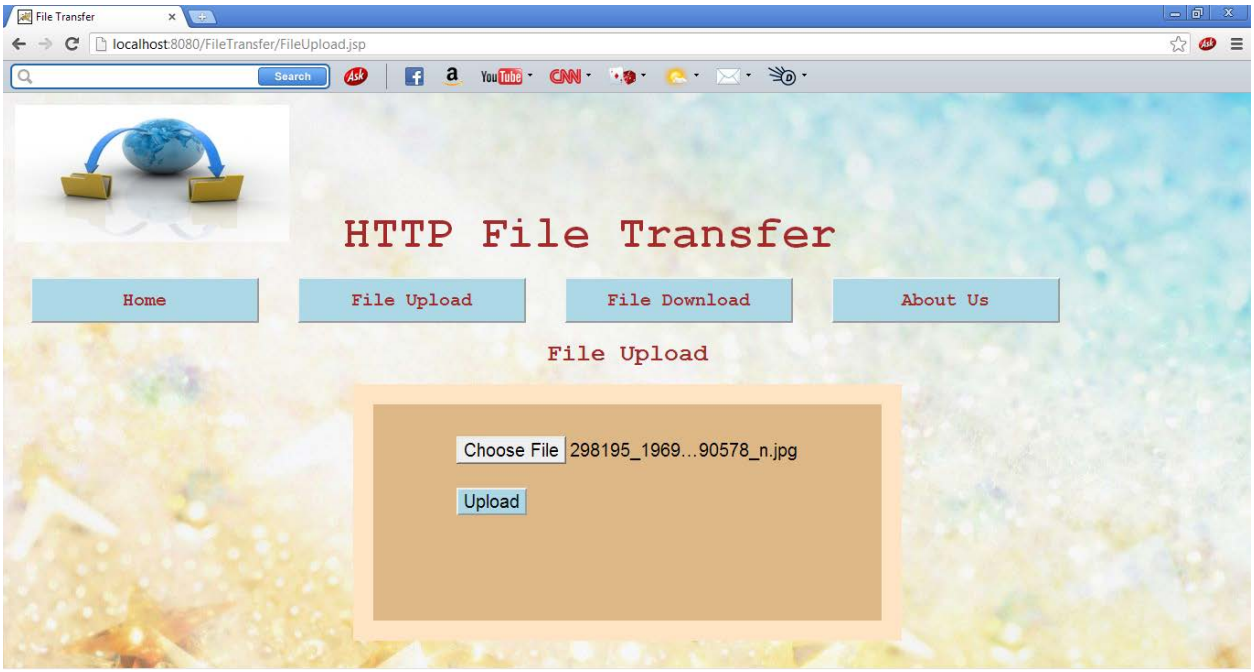
5. Software Interface

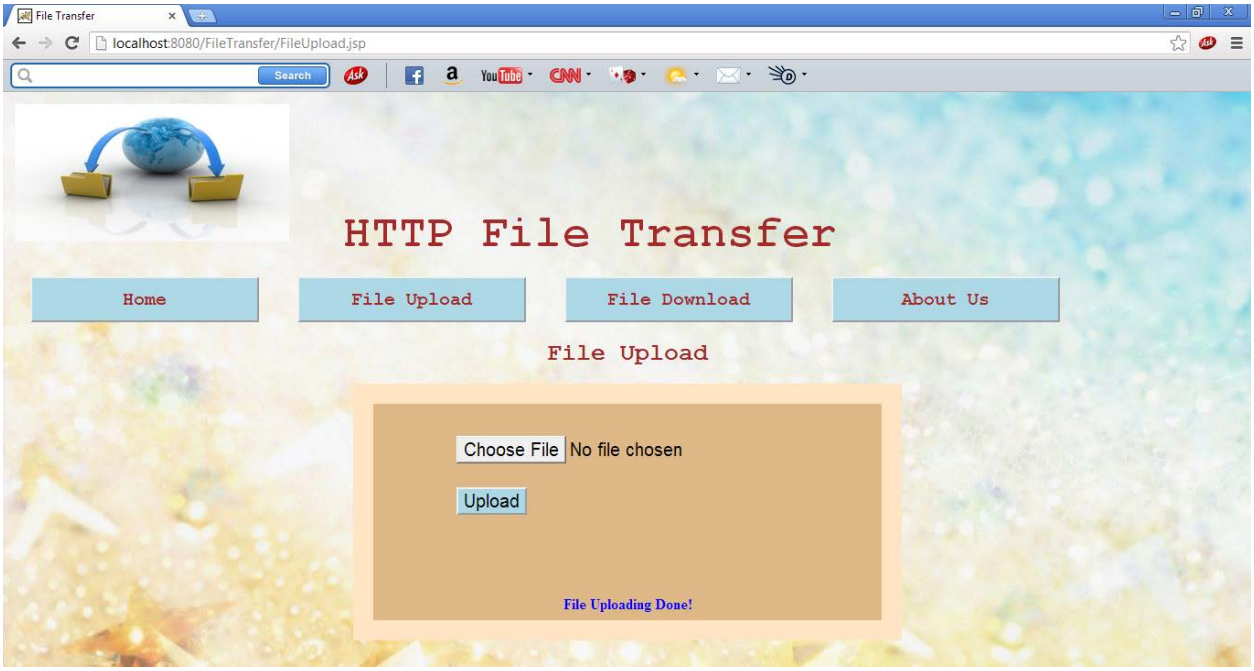
Home Page:



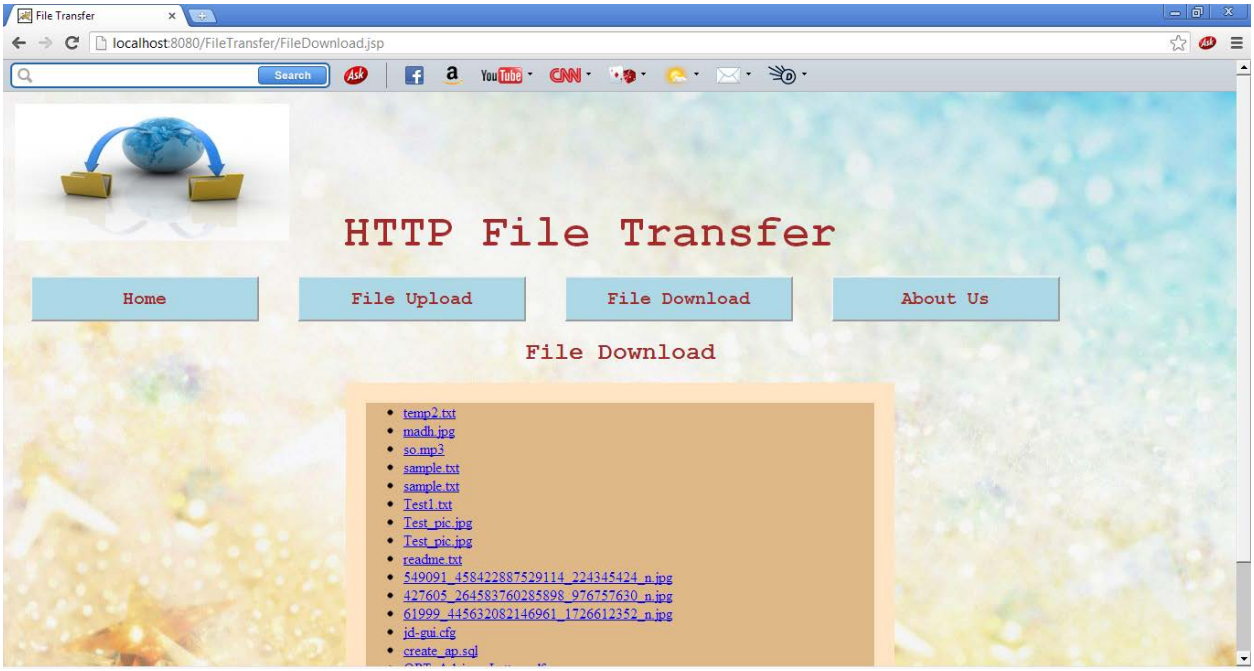
Upload Page:

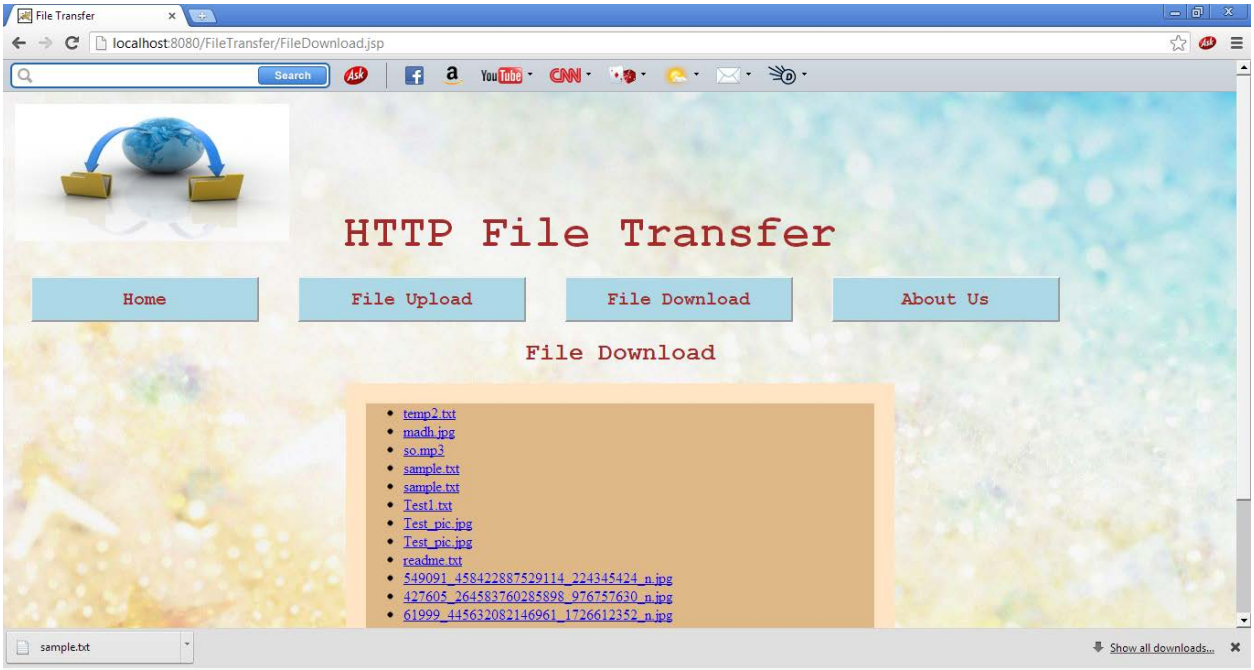

File Transfer
localhost:8080/FileTransfer/FileUpload.jsp
Search
HTTP File Transfer
Home
File Upload
File Download
About Us
File Upload
Choose File
No file chosen
Upload


File Transfer
localhost:8080/FileTransfer/FileUpload.jsp
Search
HTTP File Transfer
Home
File Upload
File Download
About Us
File Upload
Choose File
298195_1969...90578_n.jpg
Upload

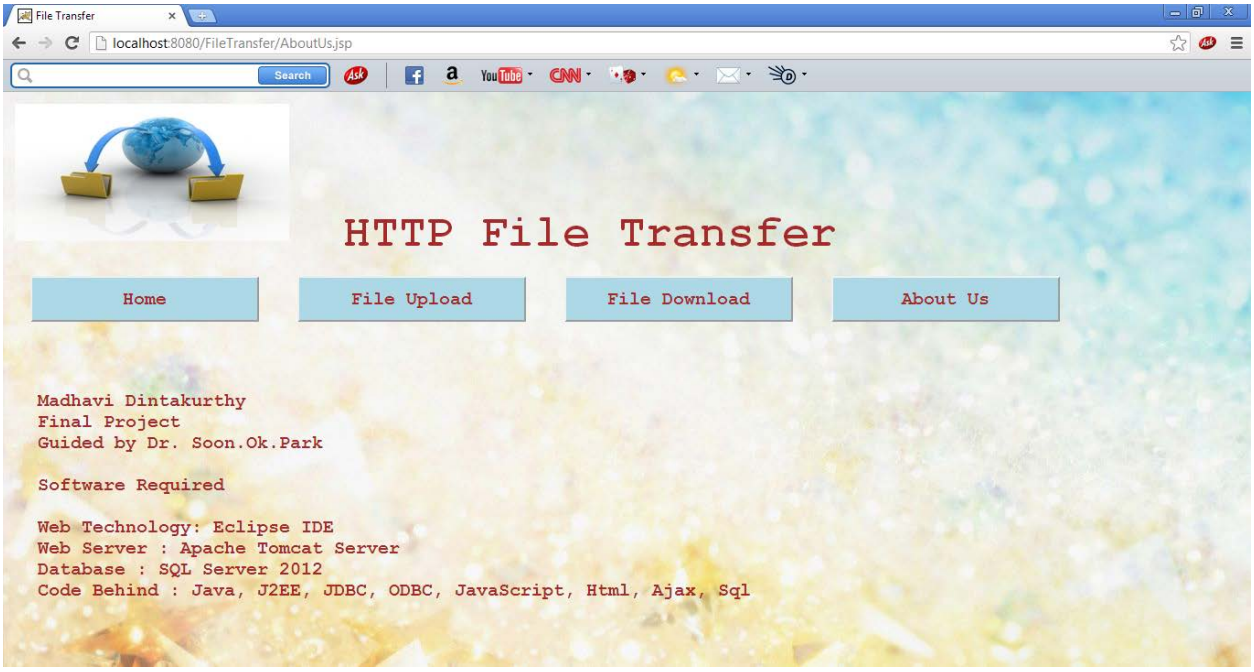


Download Page:





AboutUs Page:



6. DATABASE TABLE

File Transfer Table

MADHAVI-HP\SQLEX...Transfer - dbo.FT

Column Name	Data Type	Allow Nulls
FID	int	☐
FType	varchar(250)	☑
FData	varbinary(MAX)	☑
		☐

7. SYSTEM TESTING AND IMPLEMENTATIONS

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum amount of efforts.

Two classes of inputs are provided to test the process

1. A software configuration that includes a software requirement specification, a design specification and source code.

2. A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

Testing is divided into several distinct operations.

7.1 STRATEGIC APPROACH OF SOFTWARE TESTING

a. Unit Testing

Unit test comprises of a set tests performed by an individual program prior to the integration of the unit into large system. A program unit is usually the smallest free functioning part of the whole system. Module unit testing should be as exhaustive as possible to ensure that each representation handled by each module has been tested. All the units that makeup the system must be tested independently to ensure that they work as required.

During unit testing some errors were raised and all of them were rectified and handled well. The result was quiet satisfactory and it worked well.

b. Integration Testing

Integration testing is a system technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. Bottom-up integration is the traditional strategy used to integrate the components of a software system into functioning whole. Bottom-up integration consists of unit test followed by testing of the entire system. A sub-system consists of several modules that communicated with other defined interface.

The system was done the integration testing. All the modules were tested for their compatibility with other modules .They test was almost successful. All the modules coexisted very well, with almost no bugs. All the modules were encapsulated very well so as to not hamper the execution of other modules.

c. Validation Testing

After validation testing, software is completely assembled as a package, interfacing errors that have been uncovered and corrected and the final series of software test; the validation test begins. Steps taken during software design and testing can greatly improve the probability of successful integration in the larger system. System testing is actually a series of different tests whose primary purpose is to fully exercise the compute –based system.

d. Recovery Testing

It is a system that forces the software to fail in a variety of ways and verifies that the recovery is properly performed.

e. Security Testing

It attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration. The system's security must of course be tested from in vulnerability form frontal attack.

f. Stress Testing

Stress tools are designed to confront programs with abnormal situations. Stress testing executes a system in a manner that demands resources in abnormal quantity and volume.

g. Black Box Testing

Black box testing is done to find out the following information as shown in below:

1. Incorrect or missing functions.
2. Interface errors.
3. Errors or database access.
4. Performance error.
5. Termination error.

The mentioned testing is carried out successfully for this application according to the user's requirement specification.

h. Test Data Output

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

BIBLIOGRAPHY

- Herhert Schildt --The Complete Reference Java2 (Fifth Edition)
- E. Balaguruswamy—Core Java
- Java and Software Design Concepts by APress
- Jim Farely $ William Crawford—Java Enterprise (Third edition)
- http://tomcat.apache.org/
- http://www.jdbc-tutorial.com/
- www.msdn.microsoft.com