

Name	Harsh Jadhav
Project Name	My Movie Plan
Phase	Capstone Project

Project Details

Project Name: MyMoviePlan

Project Description: Create a dynamic and responsive web application for booking movie tickets online for different genres and languages.

NMS Cinemas is a chain of single screen theatres that screen movie shows of different genres and languages at very genuine prices. It was established in 2004 in Pune, India. Recently, the business analysts noticed a decline in sales since 2010. They found out that the online booking of movie tickets from apps, such as BookMyShow and Paytm were gaining more profit by eliminating middlemen from the equation. As a result, the team decided to hire a Full Stack developer to develop an online movie ticket booking web application with a rich and user-friendly interface.

Features of the application:

1. Registration
2. Login
3. Payment gateway
4. Searching
5. Filtering
6. Sorting
7. Dynamic data
8. Responsive and compatible with different devices

Language used	java
Database used	MySQL
Framework	spring boot, Hibernate, Angular
Tool used	Spring Tool Suite 4 as code editor and MySQL workbench
Backend logic	Java programming, NodeJS
Frontend development	JSP, Angular, Bootstrap, HTML/CSS, and Javascript
Automation and testing technologies	Selenium, Jasmine, and TestNG
DevOps and production technologies	Git, GitHub, Jenkins, Docker, Kubernetes, and AWS
Methodology	Agile Scrum Methodology

Source Code

Controller:

- checkoutController.java
- bookingHistoryController.java
- bookedSeatsController.java
- userController.java
- theaterController.java
- showController.java
- movieController.java

Model:

- BookSeatsRequest.java
- BookingHistory.java
- BookedSeats.java
- authUser.java
- User.java
- Theater.java
- MovieShow.java
- Movie.java
- checkoutModel.java

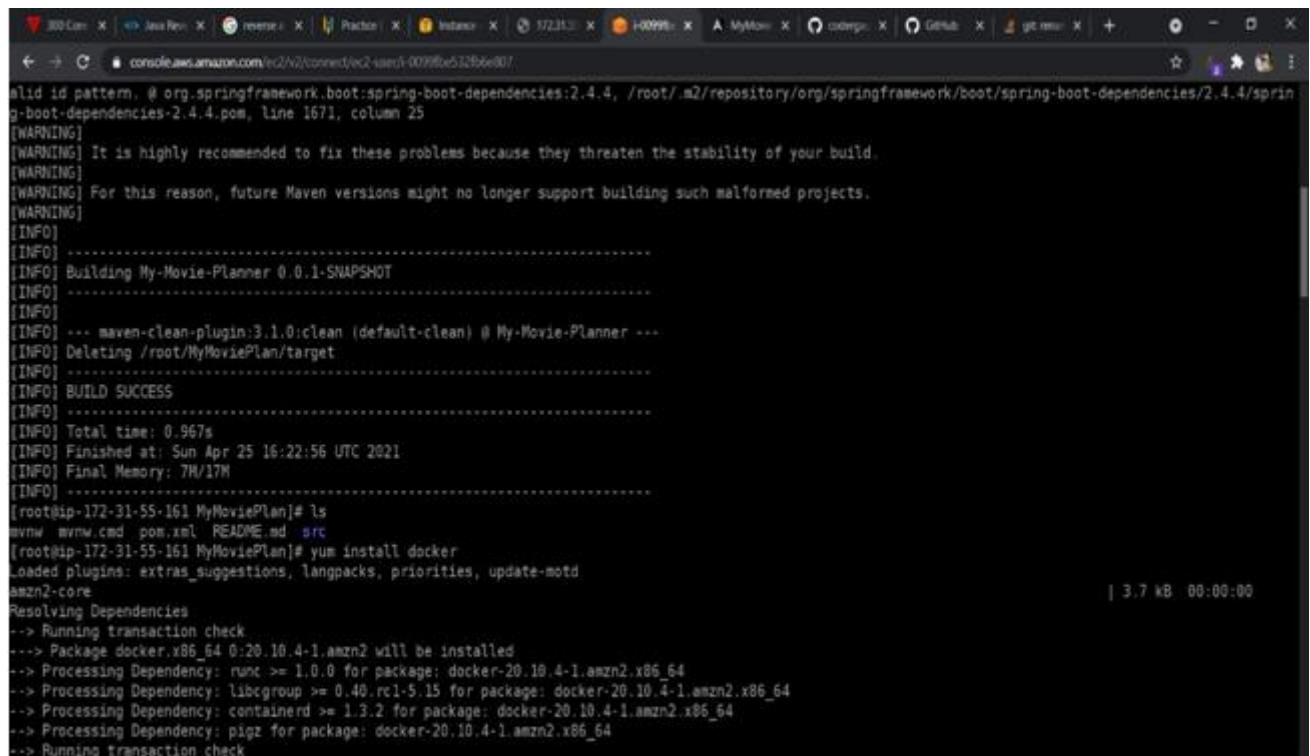
Repository:

- showRepository.java
- movieRepository.java
- bookingHistoryRepository.java
- bookedSeatsRepository.java
- userRepository.java

- theaterRepository.java
- MyMoviePlannerApplication.java
- MyMoviePlannerApplication.java

va

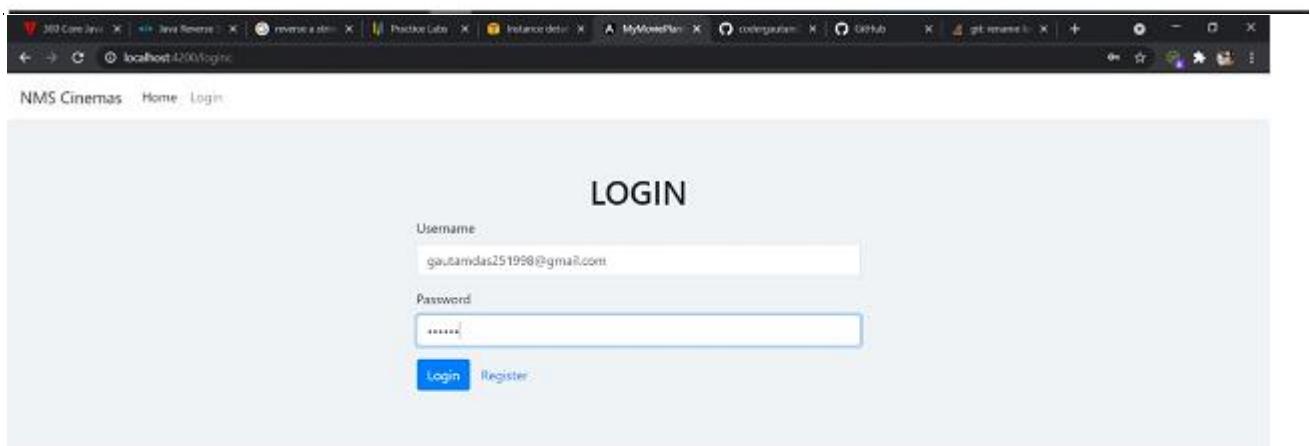
Screenshots



```
valid id pattern, @ org.springframework.boot:spring-boot-dependencies:2.4.4, /root/.m2/repository/org/springframework/boot/spring-boot-dependencies/2.4.4/spring-boot-dependencies-2.4.4.pom, line 1671, column 25
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----
[INFO] Building My-Movie-Planner 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ My-Movie-Planner ---
[INFO] Deleting /root/.MyMoviePlan/target
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.967s
[INFO] Finished at: Sun Apr 25 16:22:56 UTC 2021
[INFO] Final Memory: 7M/17M
[INFO] -----
[root@ip-172-31-55-161 MyMoviePlan]# ls
mvnw mvnw.cmd pom.xml README.md src
[root@ip-172-31-55-161 MyMoviePlan]# yum install docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                               | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.4-1.amzn2 will be installed
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.4-1.amzn2.x86_64
--> Processing Dependency: libcgrouper >= 0.40.rc1-5.15 for package: docker-20.10.4-1.amzn2.x86_64
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.4-1.amzn2.x86_64
--> Processing Dependency: pigz for package: docker-20.10.4-1.amzn2.x86_64
--> Running transaction check
```

i-0099f5e532fb6e807

Public IPs: 54.146.93.247 Private IPs: 172.31.55.161



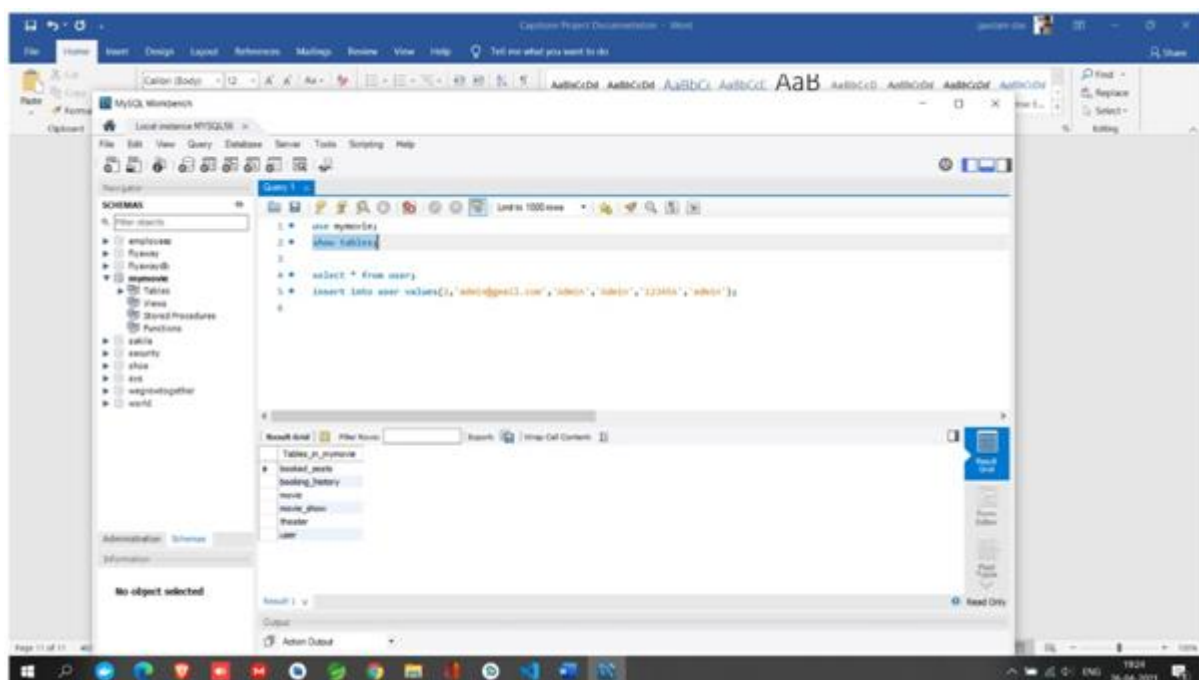
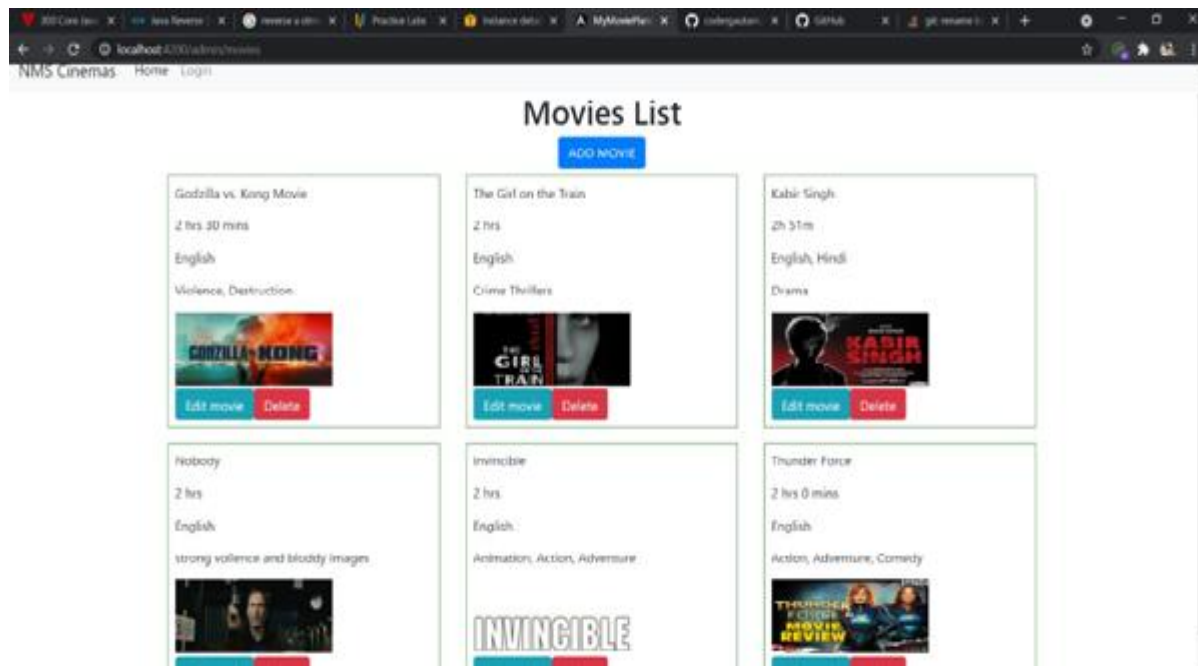
NMS Cinemas Home Login

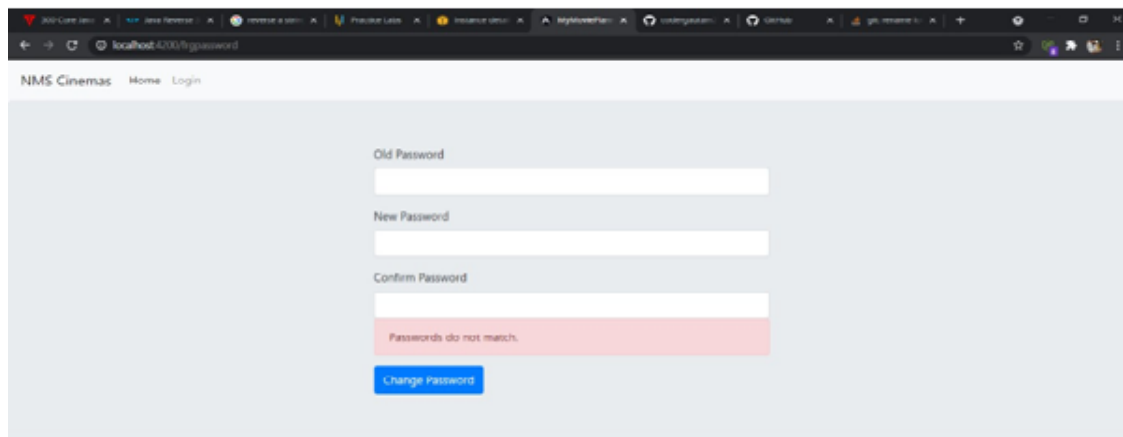
LOGIN

Username

Password

[Login](#) [Register](#)





NMS Cinemas Home Login

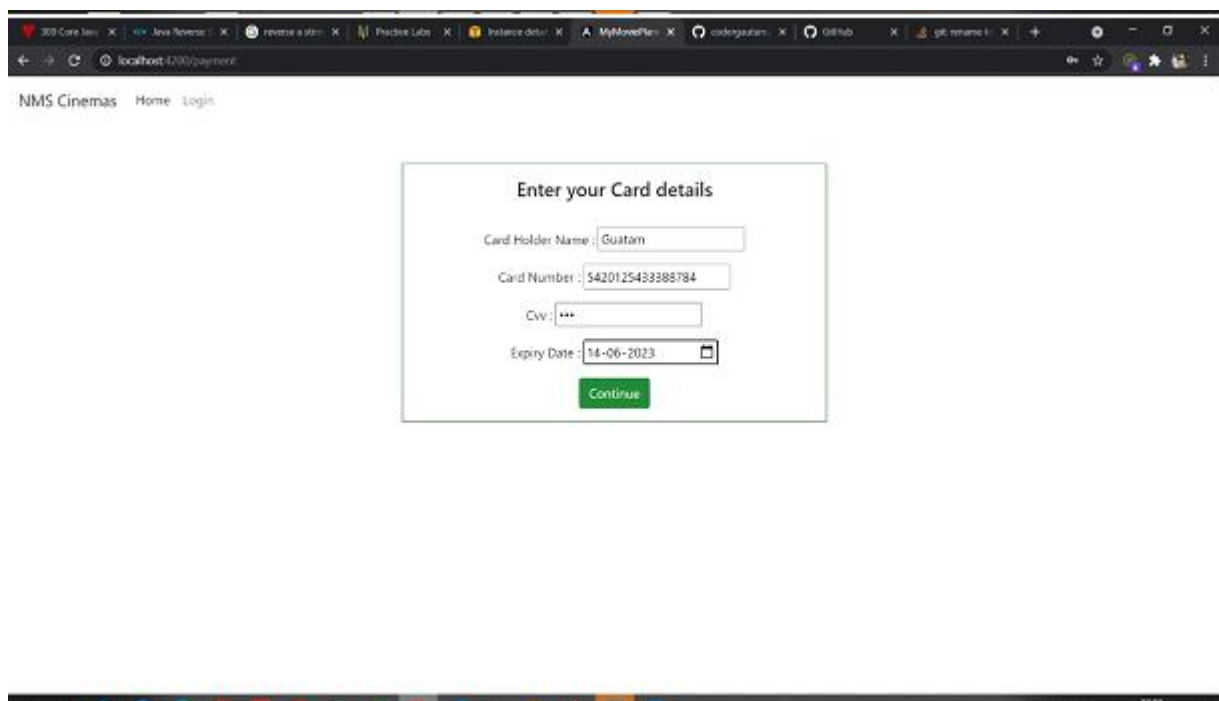
Old Password

New Password

Confirm Password

Passwords do not match.

Change Password



NMS Cinemas Home Login

Enter your Card details

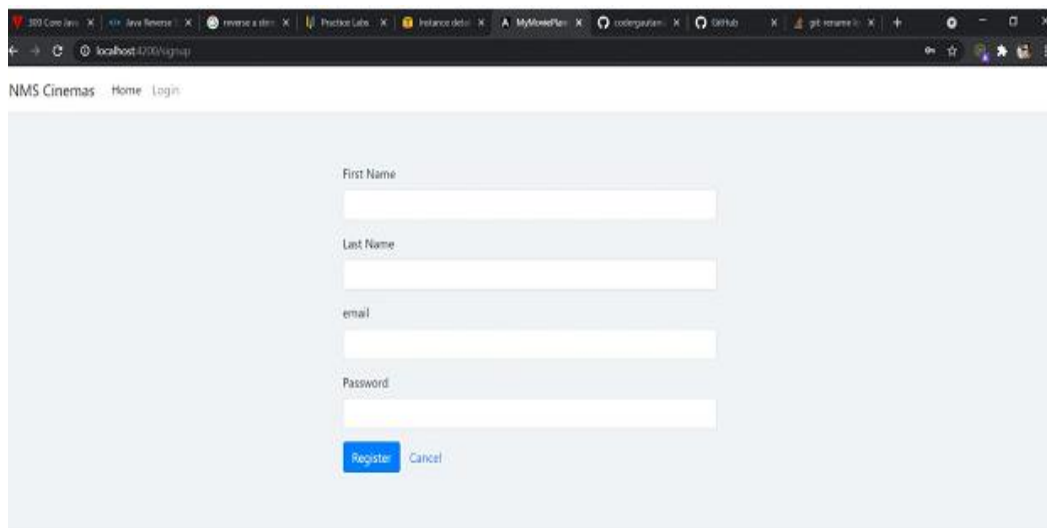
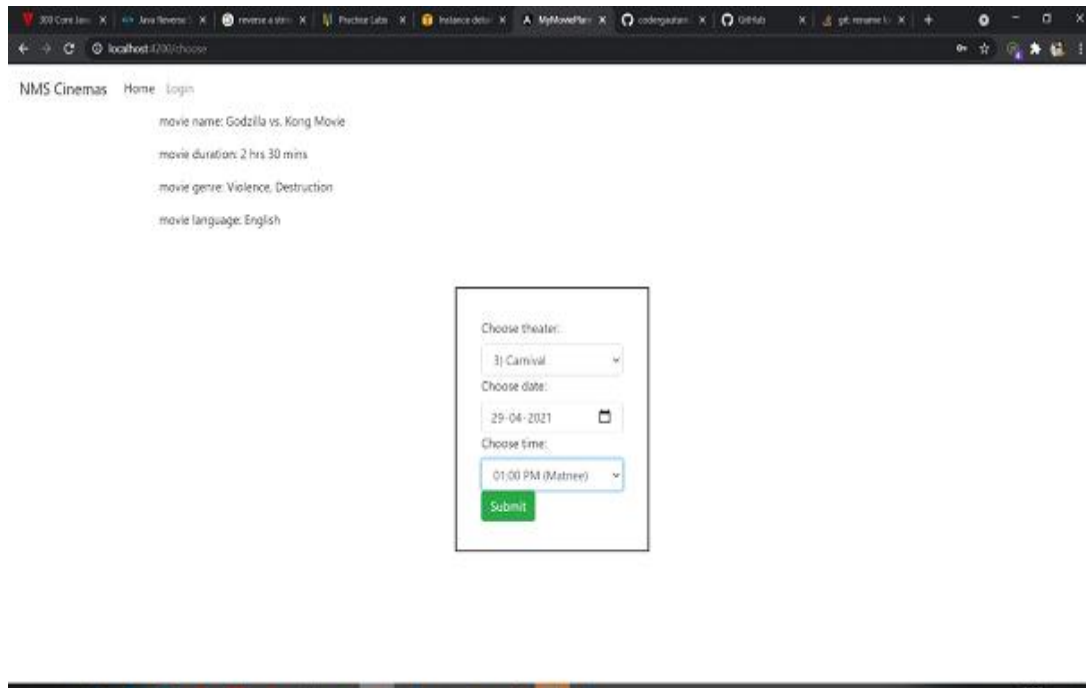
Card Holder Name : Guatam

Card Number : 542012543388784

Cvv : ***

Expiry Date : 14-06-2023

Continue



1. TECHNOLOGIES USED

1.1 JAVA

In my project, I have chosen **Java** language for developing the code.

About Java

Initially the language was called as “Oak” but it was renamed as “Java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language.
- Except for those constraints imposed by the internet environment, Java gives the programmer, full control.
- Finally, java is to internet programming where C was to system programming.

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because, Java expands the Universe of objects are transmitted between the Server and the Personal Computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called Applet.

Java can be used create two types of programs

Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java-compatible web browser.

And applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is intelligent program, not just a media file. It can react to the user input and dynamically change.

Features of java

- **Security**

Every time you that you download a “normal” program; you are risking a viral infection. Prior to Java, most users did not download executable programs frequently, and those who did scan them for viruses prior to execution. Most users still worried about malicious program exists that must be

guarded against. This type of program can gather private information. Such as credit card numbers, bank account balances, and passwords. Java answers the both of these concerns by providing a “firewall” between a networked application and your computer.

- **Portability**

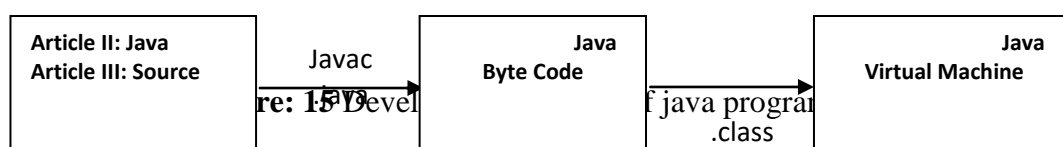
For programs to be dynamically downloaded to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed as you will see the same mechanism that helps ensure security also helps create portability. Indeed, Java’s solution to these two problems is both elegant and efficient.

- **The Byte code**

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code. Translating a java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-time package exists for a given system, any Java program can run on it.

Although Java was designed for interpretation, there is technically nothing about Java that prevents on-the-fly compilation of byte code into native code. Sun has just completed its Just In Time (JIT) compiler for byte code. When the JIT compiler is a part of JVM, it compiles byte code into executable code in real time, on a piece –by-piece, all at once, because Java performs various run-time checks that can be done only at run Java Virtual Machine (JVM).

Beyond the language, there is the Java virtual machine. The java virtual machine is an important element of the java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that’s has been generated by the compiler will not corrupt the machine that it’s loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code



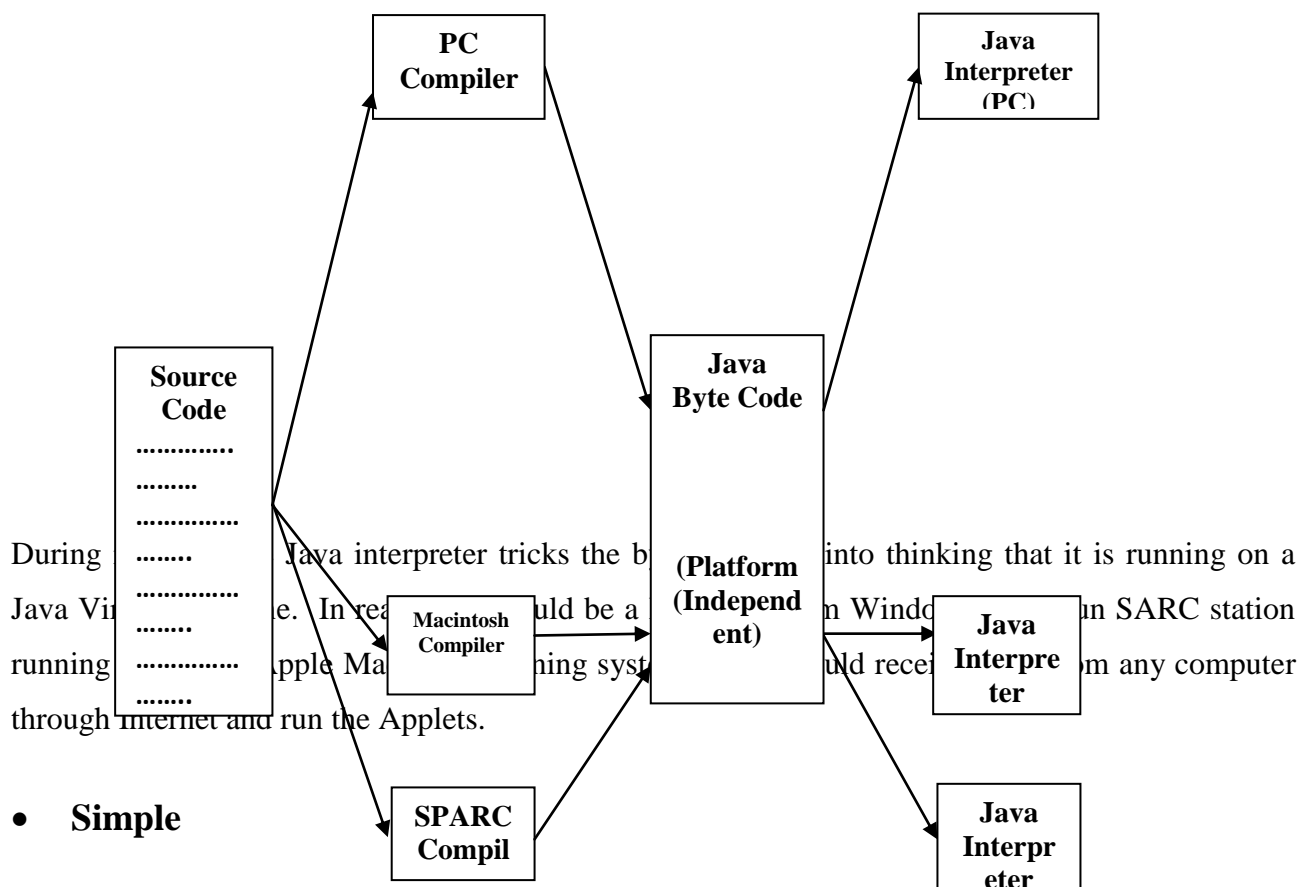
- **Java Architecture**

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

- **Compilation of code:**

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for on machine and interpreted on all machines. This machine is called Java Virtual Machine.

- **Compiling and interpreting Java Source Code**



- **Simple**

Java was designed to be easy for the Professional programmer to learn and to use effectively. If

you are an experienced C++ programmer, learning Java will be even easier. Because Java inherits the C/C++ syntax and many of the object oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In java there are a small number of clearly defined ways to accomplish a given task.

- **Object-Oriented**

Java was not designed to be source code compatible with any other language. This allowed the Java team the freedom to design with a blank slate. One outcome of this was a clean usable, pragmatic approach to objects. The object model in java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

- **Robust**

The multi-platform environment of the Web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deallocation, which is completely automatic. In a well-written Java program, all run time errors –can and should –be managed by your program.

1.2 SERVLETS

Introduction

The Java Web server is javasofts own web Server. The java web server is just a part of a larger framework, intended to provide you not just with a web server, but also with tools. To build customized network servers of ay Internet or Intranet client/server system. Servlets are to a web server, how applets are to browser.

About servlets

Servlets provide a Java –based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side-object byte codes that can be dynamically loaded off the net. They differ from applet that they are faceless objects (without graphics or a GUI component). They serve as platform independent, dynamically loadable, pluggable helper byte code objects on the server side that can be used to dynamically extend server-

side functionality.

For example, an HTTP Servlets can be used to generate dynamic HTML content. When you use servlets to do dynamic content you get the following advantages:

- They're faster and cleaner than CGI scripts.
- They use a standard API (the Servlets API).
- They provide all the advantages of Java (run on a variety of servers without needing to be rewritten).

1.3 JAVA SCRIPT

JavaScript is a script-based programming language which was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java.

JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write web server programs that can process information submitted by a Web browser and then updates the browser's accordingly.

Even though **JavaScript** supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it.

JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags `<SCRIPT>...</SCRIPT>`.

```
<SCRIPT LANGUAGE="JavaScript">
```

```
    JavaScript statements
```

```
</SCRIPT>
```

Here are a few things we can do with JavaScript:

- Validate the contents of a form and make calculations.
- Add scrolling or changing messages to the Browser's status line.
- Animate images or rotate images that change when we move the mouse over them.
- Detect the browser in use and display different content for different browsers.

- Detect installed plug-ins and notify the user if a plug-in is required.

We can do much more with JavaScript, including creating entire application.

JavaScript vs. Java

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

- Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
- While JavaScript is best suited to simple applications and adding interactive features to Web pages. Java can be used for incredibly complex applications. There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

Advantages

- JavaScript can be used for Server-side and Client-side scripting. It is more flexible than VBScript.
- JavaScript is the default scripting languages at Client-side since all the browsers supports it.

1.4 JAVA SERVER PAGES (JSP)

A JSP page is a text-based document that describes how to process a request to create a response. The description intermixes template data with some dynamic actions and leverages on the Java Platform.

The Java Server Pages specification includes:

- Standard directives
- Standard actions
- Script language declarations, script lets and expressions
- A portable tag extension mechanism.

Directives and Actions

There may be two types of elements in a JSP page: **directives** or **actions**. Directives provide global information that is conceptually valid independent of any specific request received by the JSP page.

For example, a directive can be used to indicate the scripting language to use in a JSP page. Actions may, and often will, depend on the details of the specific request received by the JSP page. If a JSP container uses a compiler or translator, the directives can be seen as providing information for the compilation/translation phase, while actions are information for the subsequent request processing phase.

An action may create some objects and may make them available to the scripting elements through some scripting-specific variables.

Directive elements: Syntax:

```
<% @ directive ... %>
```

Action elements: tag:<mytag attr1="attribute value" ...>

body

```
</mytag>
```

or an empty tag

```
<mytag attr1="attribute value" .../>
```

An element type abstracts some functionality by defining a specialized (sub)language that allows more natural expression of the tasks desired, can be read and written more easily by tools and also can even contribute specialized yet portable tool support to create them. The JSP specification provides a Tag Extension mechanism that enables the addition of new actions, thus allowing the JSP page “language” to be easily extended in a portable fashion. A typical example would be elements to support embedded database queries. Tag libraries can be used by JSP page authoring tools and can be distributed along with JSP pages to any JSP container like Web and Application servers.

The Tag Extension mechanism can be used from JSP pages written using any valid scripting language, although the mechanism itself only assumes a Java run time environment. Custom actions provide access to the attribute values and to their body; they can be nested and their bodies can include scripting elements.

Execution

A JSP page is executed in a JSP container, which is installed on a Web server, or on a Web enabled application server. The JSP container delivers requests from a client to a JSP page and responses from the JSP page to the client. All JSP containers must support HTTP as a protocol for requests and responses, but a container may also support additional request/response protocols. The default request and response objects are of type `HttpServletRequest` and `HttpServletResponse`, respectively.

Compilation

JSP pages may be compiled into its JSP page implementation class plus some deployment information. This enables the use of JSP page authoring tools and JSP tag libraries to author a Servlet. This has several benefits:

Removal of the start-up lag that occurs when a JSP page delivered as source receives the first request & Reduction of the footprint needed to run a JSP container, as the java compiler is not needed.

Objects and Scopes

JSP page can create and/or access some Java objects when processing a request. The JSP specification indicates that some objects are created implicitly, perhaps as a result of a directive other objects are created explicitly through actions; objects can also be created directly using scripting code, although this is less common. The created objects have a scope attribute defining where there is a reference to the object and when that reference is removed.

1.5 HTML

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produce Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, **SGML (Standard Generalized Markup Language)**, but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preferences.

A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized words that load to other documents or some portions of the same documents.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not tag is part of the document itself.

Basic HTML Tags

<!-- -->	specifies comments
<A>.....	Creates hypertext links
.....	Formats text as bold
<BIG>.....</BIG>	Formats text in large font
<BODY>.....</BODY>	Contains all tags and text in the HTML document
<CENTER>....</CENTER>	Creates text
<DD>.....</DD>	Definitions of a term
<DL>.....</DL>	Creates definition list
.....	Formats text with a particular font
<FORM>.....</FORM>	Encloses a fill-out form
<FRAME>....</FRAME>	Defines a particular frame in a set of frames
<H#>.....</H#>	Creates headings of different levels
<HEAD>.....</HEAD>	Contains tags that specify information about a document
<HR>.....</HR>	Creates a horizontal rule
<HTML>.....</HTML>	Contains all other HTML tags
<META>.....</META>	Provides meta-information about a document
<SCRIPT>.....</SCRIPT>	Contains client-side or server-side script
<TABLE>.....</TABLE>	Creates a table
<TD>.....</TD>	Indicates table data in a table
<TR>.....</TR>	Designates a table row
<TH>.....</TH>	Creates a heading in a table.

Advantages

- A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
- HTML is platform independent and HTML tags are not case-sensitive.

1.6 Java Database Connectivity

What is JDBC

JDBC is a Java API for executing SQL statements. (As a point of interest, JDBC is a trademarked name and is not an acronym; nevertheless, JDBC is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java programming language. JDBC provides a standard API for tool/database developers and makes it

possible to write database applications using a pure Java API.

Using JDBC, it is easy to send SQL statement to virtually any relational database. One can write a single program using the JDBC API, and the program will be able to send SQL statements to appropriate database. The combinations of java and JDBC lets a programmer write it once and run it anywhere.

What Does JDBC Do

Simply put, JDBC makes it possible to do three things:

- Establish a Connection with a database
- Send SQL statements
- Process the results

JDBC versus ODBC and other APIs

At this point, Microsoft's ODBC (open database Connectivity) API is that probably the most widely used programming interface for accessing relational databases. It offers the ability to connect to almost all databases on almost on almost all platforms.

So why not just use ODBC from java? The answer is that you can use ODBC from java, but this is best done with the help of JDBC in the form of the JDBC-ODBC.

Bridge, which we will cover shortly. The question now becomes "why do you need JDBC? " There are several answers to this question:

- ODBC is not appropriate for direct use from java because it uses a c Interface. Calls from java to native c code have a number of drawbacks in the security, implementation, robustness, and automatic portability of applications.
- A literal translation of the ODBC C API would not be desirable. For example, java has no pointers, and ODBC makes copious use of them including the notoriously error-prone generic pointer "void *". You can think of JDBC as ODBC translated into an object-oriented interface that is natural for java programmers.
- ODBC is hard to learn. It mixes simple and advanced features together, and it has complex options even for simple queries. JDBC, on the other hand, was designed to keep simple things simple while allowing more advanced capabilities where required.
- A java API like JDBC is needed in order to enable a "pure java" solution. When ODBC is used, the ODBC drivers manage and drivers must be manually installed on every client machine.

When the JDBC driver is written completely in java, however, JDBC code is automatically installable, portable and secure on all java platforms from network computers to mainframes.

1.7 ORACLE

Oracle is a relational database. The language used to access relation database is Structured Query language, SQL is flexible, efficient language, with features designed to manipulate and examine relational data.

SQL is fourth generation language. This means that the language describes what should be done, but not how to do it. Fourth generation languages are fairly simple and have fewer commands, 4 GL's also insulate the user from underlying data structures and algorithms.

Why to choose oracle

Oracle is popular relational database. Software is available on a wide number of platforms. Oracle provides complete control organizing the data storage to obtain good performance using indexing, clustering.

Introduction to SQL

STRUCTURED QUERY LANGUAGE is a tool of communication b/w a user and RDBMS. SQL is a simple and powerful language in the sense that most of the operations in RDBMS can be done using SQL. It is a 4th GL. The important feature is that it is not procedural, advanced of using non-procedural languages are:

- Reduction of code
- Simplicity in writing code.
- Ease of maintenance.

SQL is made of 3 sub languages:

DDL: Data Definition Language consists of commands to create the objects, such as tables, views, indexes etc.

DML: Data Manipulation Language is used to querying, inserting deleting and updating of information stored in the database.

DCL: Data Control Language is used to control data and access to the database in the multi-user environment for security purpose. The DCL is used for giving access to different users.

ORACLE Utilities

Oracle utilities enhance RDBMS support of data entry, maintenance and retrieval. They are Export, import and SQL Loader.

EXPORT

Export is the only utility provided for oracle. Export writes data from an oracle database into oracle binary format files. The export makes a copy of data structure in an operating system file.

To export

- A user must have DBA privileges to Oracle database.
- A user must own tables to be exported.
- Before running there must be enough storage space on disk of tape to write exported files. If there is not enough space, export will terminate with a write failure error.
- The information that is to be exported depends upon the mode chosen.

- Export modes are tables, user, and full database.
- Export can be of three types incremental, cumulative and complete.

IMPORT

Import reads files created by export and places data in the database. To import user should have connection or privileges to an oracle database and access to an export file.

Only a DBA can import a file exported by him/her. A table is imported in to its original table space using the original storage parameters. If the storage space does not exist then the system uses default space of the current user. If an error occurs the import prints an error message and skips to the next table and continues the processing.

SQL * Loader

It is a tool for loading data from external files into oracle database. There are two mechanisms used to load data. Two types of inputs are to be provided to SQL * Loader.

Project Details

Project Name: MyMoviePlan

Project Description: Create a dynamic and responsive web application for booking movie tickets online for different genres and languages.

Language used : java

Database used : MySQL

Framework : spring boot, Hibernate, Angular.

Tool used : Spring Tool Suite 4 as code editor and MySQL workbench.

Backend logic : Java programming, NodeJS

Frontend development : JSP, Angular, Bootstrap, HTML/CSS, and Javascript

Automation and testing technologies : Selenium, Jasmine, and TestNG

DevOps and production technologies : Git, GitHub, Jenkins, Docker, Kubernetes, and AWS

Methodology : Agile Scrum Methodology

Sprint required : 4 sprints of 7 days each

GitHub URL

<https://github.com/HarshJadhav26?tab=repositories>