Chapter 1

Introduction to Full Stack Development

Introduction

- The evolution of computer science increased rapidly by the late 20th century, computers became smaller and more powerful.
- However, there was no central solution to digitally connect the masses.
- The invention of the Internet in the 1970s.
- The Internet was available for public use after the start of the 1990s, as a result the world of Internet connectivity.
- The advent of the Internet causes the physical shops were transformed into digital entities.
- Today, businesses and governments use the Internet to facilitate their operations.
- The Internet mainly relies on the World Wide Web (WWW), also known simply as the Web.
- The WWW uses Hyper Text Transfer Protocol (HTTP) to distribute information across various networks.
- This information is written in websites are formatted in Hyper Text Markup Language (HTML).
- In the last three decades, the design, structure and capabilities of websites have changed a lot.
    - From static websites to dynamic websites.
    - From dynamic to responsive websites.
    - Due to the demands of mobile users, there has been a great deal of change in the web sphere.
- All these modifications were made possible due to the use of several web technologies.

  - With time, many technologies such as Java's Applets and newer ones such as Node.js came to the scene.

- Today, full stack web development is the contemporary practice to develop websites.

What is Full Stack Web Development?

- An e-commerce application, such as Amazon.com, is widely accessible via a website ww.amazon.com or via a mobile application.
  - Both these platforms are known as *front-end* or *client-side*.
- These *front-ends* connect with other application that resides somewhere in a remote cloud server.
  - This application is called back-end application.
  - The back-end application provides services that can feed data to the front-ends.
- The Figure shows the three-tier architecture
  - The front-end is referred as client tier
  - The back-end is divided into application tier and database tier.

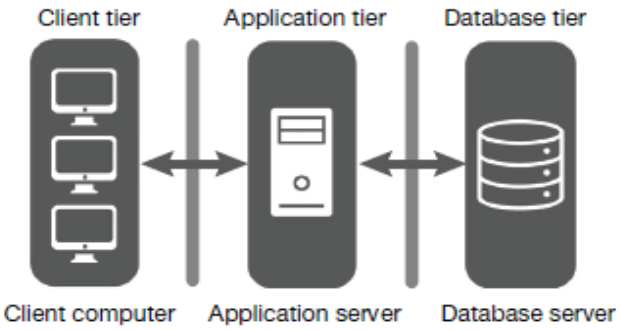


Figure 1.1 Three-tier architecture.

- **Front-end**
  - Usually, to design the view (or the client-side) of a web page which is visited by users through web browsers.

- It is all about graphics, how everything appears to users.
- The front-end usually consists all the menus, sliders, labels or anything you click or read on a website are generated with the help of front-end.
- To develop the front-end usually HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), JavaScript, jQuery, Bootstrap, and so on, are the fundamental technologies.
- Hence, front-end web development is also called web design.

- **Back-end**
  - The back-end, which is also known as the server-side.
  - The back-end is used for business logic.
  - It consists of a server which receives requests, an application which waits for requests and generates a response, and a database which stores all the data.
  - The back-end contains two parts: (a) application layer and (b) persistence layer, also known as database layer.
  - The back-end can be developed in Java, C#, Python, Ruby on Rails, Node.js and so on.
  - The back-end is divided into *application tier* and *database tier*.
  - The application tier contains business logic
  - The database tier stores data generated by the application or entered by a user from the front-end tier.

- Traditionally, front-end and back-end are handled by separate professionals who have mastered any of these fields to power the website.
- Full stack web development is a practice in which both the front-end and back-end are managed by the same professional.
- Full stack developers are generally expertise in both.
  - They have the conceptual knowledge and the technical expertise to create both the front-end and the back-end of websites from scratch.

- A full stack developer needs to learn technologies for all the layers shown in below Figure.

Figure 1.2 Technologies for full stack application development.

Introduction to Web Application Development

- A web application exists on a server and is used when it is opened via web browsers.
- The web application programs that are run on servers and provides useful functionalities to the end-user with the help of the Internet.
- They include both client-side and server-side programming.
- They provide capabilities such as linking with databases, giving back a response to browsers, and performing a task for a user.
- Some examples of web applications are E-commerce, social media websites, online banking and so on.
- HTML, CSS, and JavaScript are some commonly used technologies for the front-end of a web application development.
- The below Figure shows, these technologies fall into the following categories:
    - *Behavioral*, which deals with behavior of an application such as actions performed on various events
    - *Structural*, which deals with forming the structure of an application such as adding tables, titles, etc.

- *Presentational*, which focuses on the look and feel of an application like color, font style, alignment, etc.

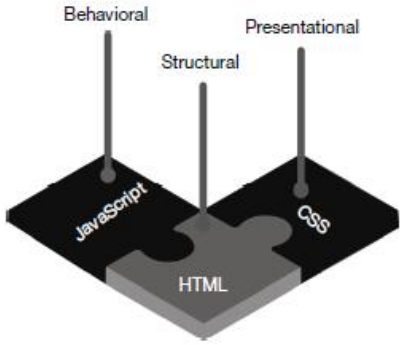


Figure 1.3 Front-end technologies for web development.

Front-End Technologies

- A web-based application usually runs on computer, laptop, mobile phone, tablet, smart watch, car dashboard screen, etc.
- Although the development of each of these front-end types would be different, the basic concept of design decisions remains the same.
- The main aim of the front-end is to provide a user-friendly interface to allow interaction with backend and database.
- The front-end development focus will be on HTML, CSS, jQuery, and Bootstrap.
- Let us just introduce these briefly.
- **HTML (Hypertext Markup Language)**
  - HTML is an irreplaceable component of web development.
  - HTML creates the arrangement of components of a web page with the help of markup.
  - Markup powers HTML elements to use text and present it in a certain way.
  - It can make some information on the page to appear "bold" on a website.
  - World Wide Web Consortium (W3C) is an organization that carries the responsibility to update it according to the ever changing web scene.

- Examples to create HTML file and run this file in any browser.

  - Open any text editor like notepad to paste the below code and save this file as "whatisfullstack.html".
  - In "whatisfullstack.html", the file extension is ".html".

  - An HTML file can be run in any web browser and produce a result.

```
<!DOCTYPE html>
<html>
        <head>
                <title>What Is Full Stack Web Development?</title>
        </head>
        <body>
                <h1> My Heading: Full Stack Development</h1>
                <p>In this book we will learn about Full Stack
                    Development.</p>
        </body>
</html>
```

- **CSS (Cascading Style Sheets)**

  - However, HTML elements are not too visually appealing.

  - CSS is heavily used to provide striking visuals to the web pages for incorporating layouts, designs, and variations.

  - As the name suggests, CSS adds "styles" to HTML components.

  - For example, let us add red color to our previous HTML example and also put it across the center.

```
<!DOCTYPE html>
<html>
        <style>
                h1 {
                        color: red;
                        text-align: center;
                }
        </style>
        <head>
                <title>What Is Full Stack Web Development?</title>
        </head>
```

```
<body>
        <h1> My Heading : Full Stack Development</h1>
        <p>In this book we will learn about Full Stack
            Development.</p>
    </body>
</html>
```

- **JavaScript**

  - After HTML and CSS, the JavaScript is the most in-demand language of recent years in the technology community.

  - JavaScript is a proper high-level programming language, a scripting one.

  - JavaScript enables developers to add more sophisticated functionality on the client-side.

  - It transforms a web page from a static to a dynamic one by adding interactivity and timely features such as animations, advanced maps, etc.

  - JavaScript has popular front-end frameworks like Angular JS, Vue JS, React JS, etc. which are used for the client-side, with the back-end with Node.js.

  - For example, let us add a simple JavaScript code to multiply two elements into the same file. We are doing a simple multiplication of two variables and placing the result in an HTML file which has id as “result”.

```
<!DOCTYPE html>
<html>
    <head>
        <title>What Is Full-Stack Web Development?</title>
        <style>
            h1 {
                color: red;
                text-align: center;
            }
        </style>
    </head>
    <body>
        <h1> My Heading : Full Stack Development</h1>
        <p>In this book we will learn about Full Stack
            Development.</p>
        <p>
            If you read 5 pages per day for 30 days, you will  complete
            <span id="result"></span> pages.
```

```
            </p>
        </body>

        <script>
            var pagesperday = 5;
            var readingdays = 30;
            var daysneeded = pagesperday * readingdays;
            document.getElementById("result").innerHTML = daysneeded;
        </script>
</html>
```

- **jQuery**
  - jQuery is a small but powerful JavaScript library.
  - It is used to ease scripting on the client-side with HTML.
  - The primary objective of jQuery is to add interactivity on a website.
  - Since it can help to write shorter codes than JavaScript, therefore it is also called "write less do more".
  - jQuery is a cross-platform. For a simple jQuery example, let us see the following.
  - This example adds dynamicity to HTML as it hides the text by a simple button click.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- jQuery CDN link. -->
    <script src=
        "https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
    </script>
  </head>

  <body>

        <h1>JQuery Example Code</h1>
        <p><b>This line will get removed once the following button is
                clicked.</b></p>

      <button id="btn">Hide</button>

        <!-- Using hide() method to hide <p/> element. -->
```

```
        <script>
            $(document).ready(function () {
                $("#btn").click(function () {
                $("p").hide();
                });
            });
        </script>
    </body>
</html>
```

- **Bootstrap**
    - In the late 2000s, several smart phones and tablets were introduced in the market.
    - At that time, websites were primarily designed for desktops and laptops. When these websites are opened with the help of smart phones, peoples are extremely unsatisfied.
    - To solve this issue, Twitter's Jacob Thornton and Mark Otto developed Bootstrap in 2011.
    - Bootstrap as not a new front-end markup language, but it used the existing front-end technologies HTML, CSS, and JavaScript and represent of a website with respect to its device and screen size.
    - The web designers makes website easily for wider audience, due to the advent of Bootstrap, It also provides developers with templates for HTML and CSS to add tables, image carousels, modals, buttons, forms, typography, etc. on websites.
    - The best thing about Bootstrap is that it is free and hosted on GitHub.
    - Following are some of the other advantages of Bootstrap:

        1. **Responsive grid:** Bootstrap provides you with its own grid system. By using it, developers can directly fill their containers with content. Setting up your custom breakpoints for each column is super easy by using their small, medium, and large breaks. You can just choose the default as it could already meet your requirements.
        2. **Responsive images:** Bootstrap helps you to automatically resize and optimize according to the screen size by using its own code. Furthermore, you can also change the shape of the images. This could be easily done without continuously switching between the code and your design software.
        3. **Components:** Bootstrap provides a handful of components which could be tracked onto your web page. This includes:

- Navigation bars.
- Dropdowns.
- Progress bars.
- Thumbnails.

4. **JavaScript in Bootstrap:** Bootstraps provides developers with tons of jQuery plugins. jQuery facilitates with a greater level of interactivity. This creates easy solutions for modal popups, image carousels, transitions, etc.

5. **Documentation:** The document of Bootstrap is one of the best documentations in the software industry. Each piece of code is well described to the smallest detail in their official websites.

6. **Bootstrap community:** Just like any other open-source software, Bootstrap also has a very large and friendly community of developers and designers. Developers find it easy to modify and contribute to the Bootstrap's database as it is hosted on GitHub. People often collaborate, give useful pieces of advice, and interact with each other to solve doubts.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Bootstrap Example</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <script
    src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
    <body>
        <div class="container">
        <h2>Table Example</h2>
        <p></p>
        <table class="table">
            <thead>
                <tr>
                    <th>Student Name</th>
                    <th>Course Marks</th>
                    <th>Total Marks</th>
                </tr>
            </thead>
            <tbody>
```

```
                    <tr>
                        <td>Merry</td>
                        <td>Introduction to Computer Vision</td>
                        <td>95</td>
                    </tr>
                    <tr>
                        <td>Jim</td>
                        <td>Distributed Computing</td>
                        <td>63</td>
                    </tr>
                    <tr>
                        <td>Morrison</td>
                        <td>System and Software Security</td>
                        <td>74</td>
                    </tr>
                </tbody>
            </table>
            </div>
        </body>
</html>
```

Back-End Technologies (Server-Side)

- The server-side requires the use of a single or multiple programming languages to write business logic, user authentication, or database (DB) related work.
- There is no backend language that is the best solution for all needs; each back-end language is good to solve a certain problem.
- The popular back-end languages as follows:

1. PHP:

- PHP is the most common tool for designing the back-ends of a website.
- Setting up websites with PHP requires minimal effort and time. It is generally used with frameworks such as Laravel, CodeIgniter, etc. However, there are various content management platforms such as WordPress, Joomla, etc. that are popular to run websites. These frameworks are customizable with plugins.

2. Node JS:

- This allows developers to write both the front-end and back-end in a single language − JavaScript.

- This is a great option for those who are proficient in the foundations of JavaScript.
- Node JS is often used in a technology stack known as MEAN (short for MongoDB, Express, Angular, Node JS).

3. Python:

- This is increasingly becoming an all-purpose language.
- It is used heavily in Artificial Intelligence and Data Science for making computers "intelligent".
- It is used for desktop development as well as design management systems.
- Python allow users to write only a few lines of code to solve particular problem.
- Python is generally used in Web with Django and Flask frameworks.

4. Java:

- Java is perhaps the most used language in the world.
- It is used all around the world for banking ecosystems because of its high security.
- The world's leading mobile operating system Android depends on mobile apps which are written in Java.
- On the web, Java uses a multitude of frameworks such as Java EE, Play Framework, Vaadin, Struts, etc. However, the most famous one is Spring Model View Controller (MVC).

5. C#:

- This is the brainchild of Microsoft.
- C# itself is inspired from Java.
- C#, along with Java, is one of the two languages that is primarily used for coding enterprise-level development.
- C# to design desktop application with Visual Studio or writing web applications with the ASP.NET framework.

Model View Controller (MVC) framework

- Model View Controller (MVC) framework is an architectural pattern that is used to build software applications.

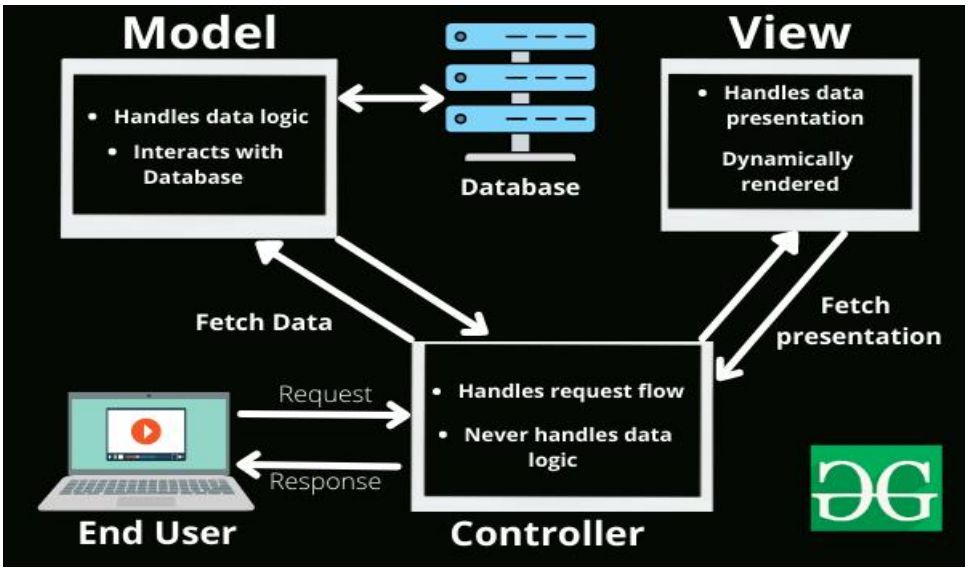
- The MVC framework separates an application into three main logical components:
  - Controller
  - Model
  - View

- Each of these components is built to handle specific development aspects of an application.

- MVC is one of the most frequently used industry-standard web development frameworks to create scalable and extensible projects.

- It is also used for designing mobile apps.

- **Controller:**
  - The controller is the component that enables the interconnection between the views and the model so it acts as an intermediary.
  - The controller doesn't have to worry about handling data logic, it just tells the model what to do.
  - It process all the business logic and incoming requests, manipulates data using the **Model** component and interacts with the **View** to render the final output.
- **View:**
  - The **View** component is used for all the User Interface (UI) logic of the application.
  - It generates a user interface for the user.
  - Views are created by the data which is collected by the model component but these data aren't taken directly but through the controller. It only interacts with the compiler.
- **Model:**
  - The **Model** component corresponds to all the data-related logic that the user works with.
  - This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data.
  - It can add or retrieve data from the database.
  - It responds to the controller's request because the controller can't interact with the database by itself.
  - The model interacts with the database and gives the required data back to the controller.



- **Advantages of MVC:**
  - Codes are easy to maintain and they can be extended easily.
  - The MVC model component can be tested separately.
  - The components of MVC can be developed simultaneously.
  - It reduces complexity by dividing an application into three units. Model, view, and controller.
  - It supports Test Driven Development (TDD).
  - It works well for Web apps that are supported by large teams of web designers and developers.
  - This architecture helps to test components independently as all classes and objects are independent of each other
  - *Search Engine Optimization* (SEO) Friendly.

- **Disadvantages of MVC:**
  - It is difficult to read, change, test, and reuse this model
  - It is not suitable for building small applications.
  - The inefficiency of data access in view.
  - The framework navigation can be complex as it introduces new layers of abstraction which requires users to adapt to the decomposition criteria of MVC.
  - Increased complexity and Inefficiency of data.

- **Popular MVC Frameworks:**
  - Some of the most popular and extensively used MVC frameworks are listed below.
    - Ruby on Rails
    - Django
    - CherryPy
    - Spring MVC

- Catalyst
- Rails
- Zend Framework
- Fuel PHP
- Laravel
- Symphony

Communication Between Front-End and Back-End

- Front-end and back-end must use a mechanism to communicate with each other.
- In order to get data from web services, we need to send a request from front-end and receive a response from the back-end.
- A browser and server can only send and receive data which is in the textual form.
- The most common data formats use to send and receive information between front-end and back-end are:
    - *JavaScript Object Notation (JSON)*
    - *Extensible Markup Language (XML)*
- **JavaScript Object Notation (JSON)**
    - JSON is a form of syntax which is used to store and transfer data on the Internet.
    - In JSON data are in plain text.
    - Objects in JavaScript can be easily transformed into JSON, which are then used by the server. The server can then convert JSON into an object of JavaScript.
    - JSON uses ordered lists and key−value pairs to manage data.
    - Due to its convenience, JSON receives support from all common programming languages.
    - The example of JSON format is presented in the following:

```
var adam =
{
“occupation”: “software developer”,
“city” : “Austin, TX”,
“country”: “USA”
};
```

    - ***Note:***
        - This program generates an object which can be accessed by “adam”.
        - The curly brackets represents the “value” of our object.

- An object can entail multiple properties by the use of key−value pair, where commas demarcate them.

- A single variable can also store information for multiple people. To do this, square brackets are to entail more than one object. For example, for two employees, see the following program:

```
var employees = [{
“name”: “Adam”,
“city” : “Austin, TX”,
“country”: “USA”
},
{
“name”: “William”,
“city” : “Fort Worth, TX”,
“country”: “USA”
}]
```

- **Extensible Markup Language (XML)**

  - Extensible Markup Language (XML) is commonly used to describe data.
  - Its format is flexible which assists to generate various formats for information while it is utilized to exchange data on enterprise networks and public networks.
  - XML bears some similarity to HTML as both make the use of markup. However, while HTML is used for displaying the contents of a web page, XML is used for displaying the data.
  - The terms self-defining and self-describing are sometimes associated with XML because the data’s structure is embedded in the data and therefore there are no requirements for any pre-building of the structure needed for the storage of data.
  - The most fundamental component of a document written in XML is known as element.
  - To define elements, tags are used.
  - All elements have an opening tag and a closing tag.
  - The root element is the outermost element, which encompasses all the elements of an XML document.
  - Hence, hierarchy is supported in XML.
  - The following is a basic XML example:

```
<?xml version=“1.0” standalone=“yes”?>
<conversation>
        <greeting>Hello World!</greeting>
        <response>Hello Friend!</response>
</conversation>
```

Object Relational Mapping (ORM) with Hibernate

- The connection between relational models and object models often results in complexities because of their unique approaches.
- While object-oriented programming (OOP) languages such as Java use an interconnected object graph to represent data, relational database management systems work with data in a tabular format.
- To solve this issue, ORM came into existence.
- ORM allows developers to access and modify objects while it saves them from thinking about the relation of objects with data sources.
- By using the OOP concept of abstraction, ORM maps details of RDBMS or XML data sources with a single or multiple objects where the updated modifications in the linked interfaces are kept hidden from the programmers.
- Example
  - There is a wide range of ORMs in different object-oriented programing languages. Let's consider the following MySQL query:
    - ***$value = SELECT * FROM collection WHERE day = 'Monday'***
    - Note: The above query retrieves a value from the table collection in which columns (day) are equal to Monday.
  - In an ORM, this query would look something like this:
    - value = collection.query(day = 'Monday')

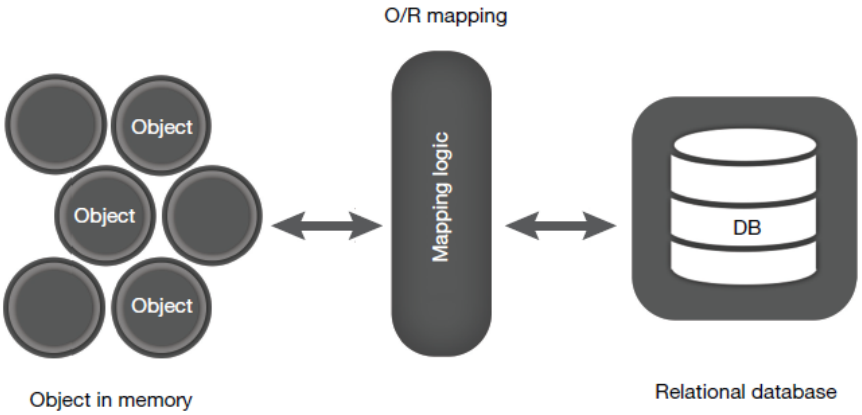


Figure 1.5 ORM mapping.

- **Hibernate**

  - Hibernate is an Object-Relational Mapping (ORM) solution for JAVA.
  - It is an open source persistent framework created by Gavin King in 2001.
  - It is a powerful, high performance Object-Relational Persistence and Query service for any Java Application.
  - Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieves the developer from 95% of common data persistence related programming tasks.
  - Hibernate sits between traditional Java objects and database server to handle all the works in persisting those objects based on the appropriate O/R mechanisms and patterns.

- **Hibernate Advantages**

  - Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.
  - Provides simple APIs for storing and retrieving Java objects directly to and from the database.
  - If there is change in the database or in any table, then you need to change the XML file properties only.
  - Abstracts away the unfamiliar SQL types and provides a way to work around familiar Java Objects.
  - Hibernate does not require an application server to operate.
  - Manipulates Complex associations of objects of your database.
  - Minimizes database access with smart fetching strategies.
  - Provides simple querying of data.