INDUSTRIAL INTERNSHIP REPORT

Music Streaming Website

Submitted in partial fulfillment of the
Requirements for the award of

Submitted By

Name : Dhyey Bhatt

University Roll No : 21BT04014

Name of Branch : Computer Science and Engineering

Submitted To

GSFC University, Vadodara

Internship Institution: PhysicsWallah Private Limited

Internship Period: 3 Weeks

Date of Report Submission: 01-03-2023

DECLARATION

I hereby declare that the Industrial Internship Report **entitled ("Music Streaming Website**") is an authentic record of my own work as requirements of Industrial Internship during the period from **09/02/2022** to **25/02/2022** for the award of degree (name of degree), GSFC University, Vadodara, under the guidance of my Faculty Mentor Ms. Avneet Saluja.

(Signature of student)
(Name of Student : Dhyey Bhatt**)**
(University Roll No : 21BT04014**)**

Date: 01/03/2022

CERTIFICATE

This is to certify that Mr. **Dhyey Bhatt** has completed Industrial Internship during the period from **09/02/2022** to **25/02/2022** in our Organization / Industry as a Partial Fulfillment of Degree of . He / She was trained in the field of **Web Development** .

Signature & Seal of Host Company Mentor

ACKNOWLEDGEMENT

In completing this graduate project I have been fortunate to have help, support and encouragement from many people. I would like to acknowledge them for their cooperation.

First and foremost deeply thankful to Faculty Mentor Ms. Avneet Saluja, for her wonderful guidance during this project work in the field of Computer Science and Engineering, at GSFC University. I am also thankful for her continuous feedback and encouragement throughout this project work. Her broad knowledge and hardworking attitude has left me with very deep impressions and they will greatly benefit me throughout my life.

I would like to thank my team members for their consistent hard work and support throughout this project.

TABLE OF CONTENT

List of Figures

Abbreviations and Nomenclature

API	**Application Programming Interface**
DOM	**Document Object Model**
CX	**Customer Experience**
FAQ	**Frequently Asked Questions**
DFD	**Data Flow Diagram**

Chapter: 1 Introduction to Project

The objective of this project is to implement a Music CD store web application with user interface. The motivation of this project comes from my desire to learn the increasingly growing field of .NET, SQL server database designing, website designing and their growing popularity by taking up this case study. The word “design” in the context of a Web Application can mean many things. Its most popular usage probably refers to the visual and user interface (UI) design of a web site. This aspect is crucial because, the visitor is often more impressed with how a website looks and how easy it is to use than about which technologies and techniques are used behind the scenes, or what operating system the web server is running. If the site is hard to use and easy to forget, it just doesn’t matter what technologies was used to create it. Unfortunately, this truth makes many inexperienced programmers underestimate the importance of the way the invisible part of the site is implemented—the code, the database, and so on. The visual part of a site gets visitors interested to begin with, but its functionality makes them come back. A web site can sometimes be implemented very quickly based on certain initial requirements, but if not properly architected, it can become difficult, if not impossible, to change. Thus, performance is also a major thrust area in the Web application which is one of the main reasons why users get attracted to it. Growing user needs should be taken in to concern with new features to be included. Effective performance can be achieved by making proper Database design strategy. Also, easy navigation also needs to be accomplished while executing this project. These are the main motivations for the project.

Chapter: 2 Major components of Project / Internship

There are two Major Components : The Administrators and the Customers/Users.

The developer operate as the administrators. They can add, edit, update products or, delete songs thus they able to change the names of songs, change name of artists and, add or remove songs from playlist.

The members can search for song selection, update the playlist, remove songs from the their playlist and check out from the web site. The members is also able to update his information such as names, address and other data. The Member is only able to browse the online music player and add a songs to the playlist..

- **Administrators Detailed Attribute**

  - Admin register

    The administrator needs to register before they can have access to the core data of the web site.

  - Admin login

    The admin logs in and can view, add products, manage member.

  - Admin Edit

    The Admin can make changes to the page such as delete member , add a member or, upload new playlist.

  - Manage member

    The administrator has the authority to delete or add a customer.

- **Member Detailed Attribute**

  - Sign up

    This refers to registering as a customer. The registered member has a lot of privileges associated with the shop when one becomes a customer.

  - Login

    After the user has registered, the user becomes a member, and he or she can log in with their personal information.

  - View

    The customer can see all the songs in the catalog and able to look at the artists and some features on the homepage.

 Edit

The customer can make changes to their data displayed on the customer page.

 Update Playlist

This refers to putting or removing songs from a playlist.

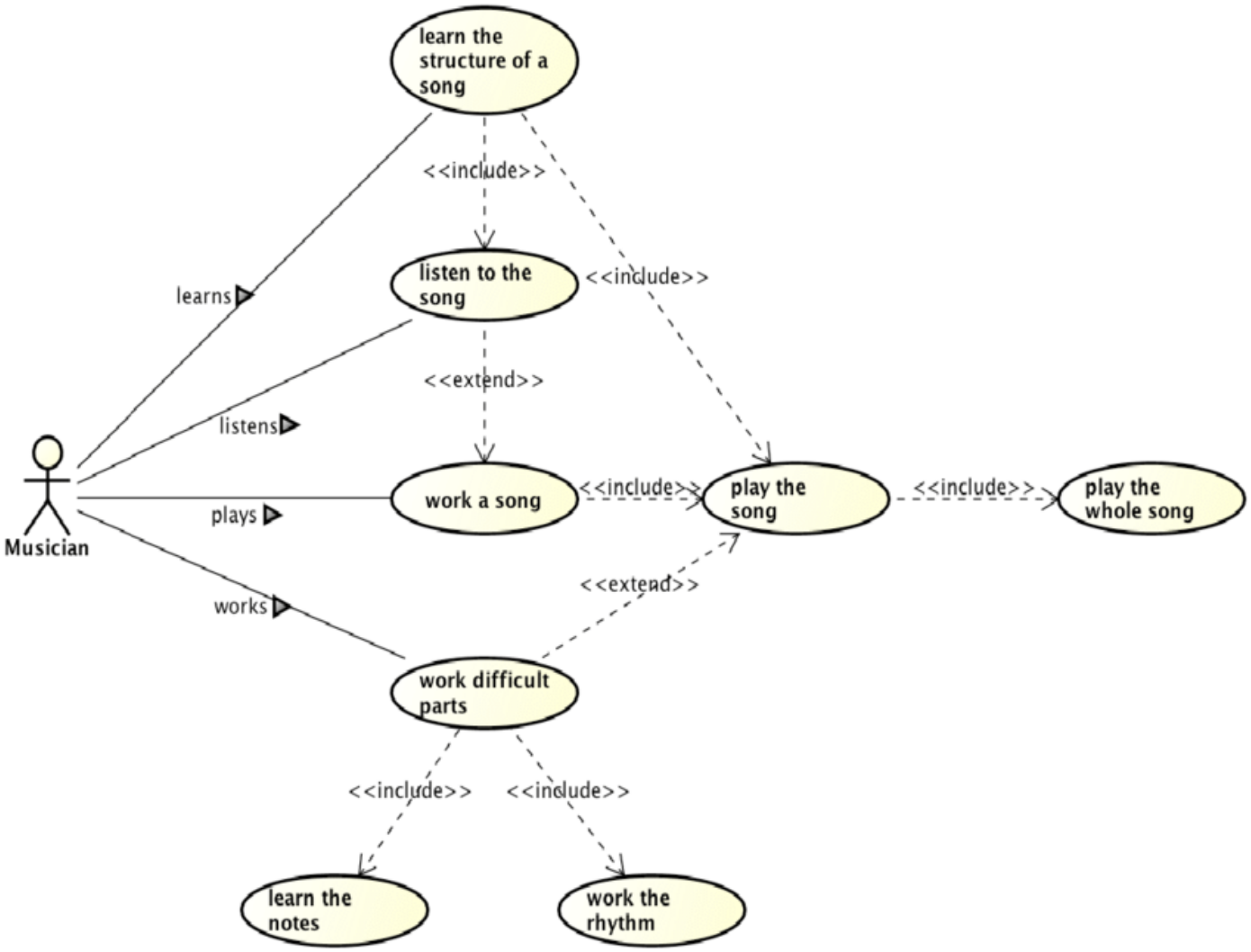


Figure 2.1 : Use case diagram of the Music Streaming Website

Chapter: 3 Methodology adopted to carry out a project / Internship

❖ DATA FLOW DIAGRAM

Data Flow Diagrams show the flow of data from external entities into the system, and from one process to another within the system. There are four symbols for drawing data flow diagram :

1. Rectangles representing external entities, which are sources or destinations of data.
2. Ellipses representing processes, which take data as input, validate and process it and output it.
3. Arrows representing the data flows, which can either, be electronic data or physical items.
4. Open-ended rectangles or a Disk symbol representing data stores, including electronic stores such as databases.

Figure 3.1 : Context Level Diagram

Figure 3.2 : First Level Diagram

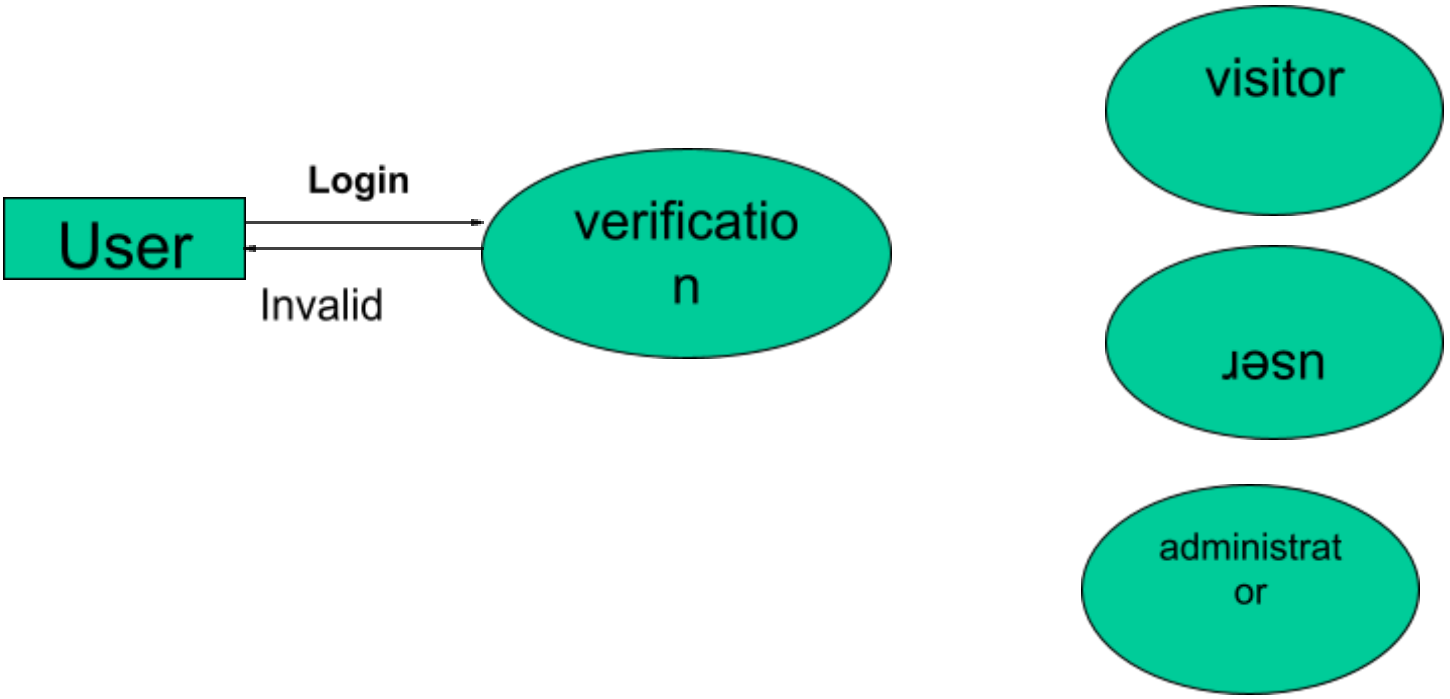


Figure 3.3 : DFD for Administrator

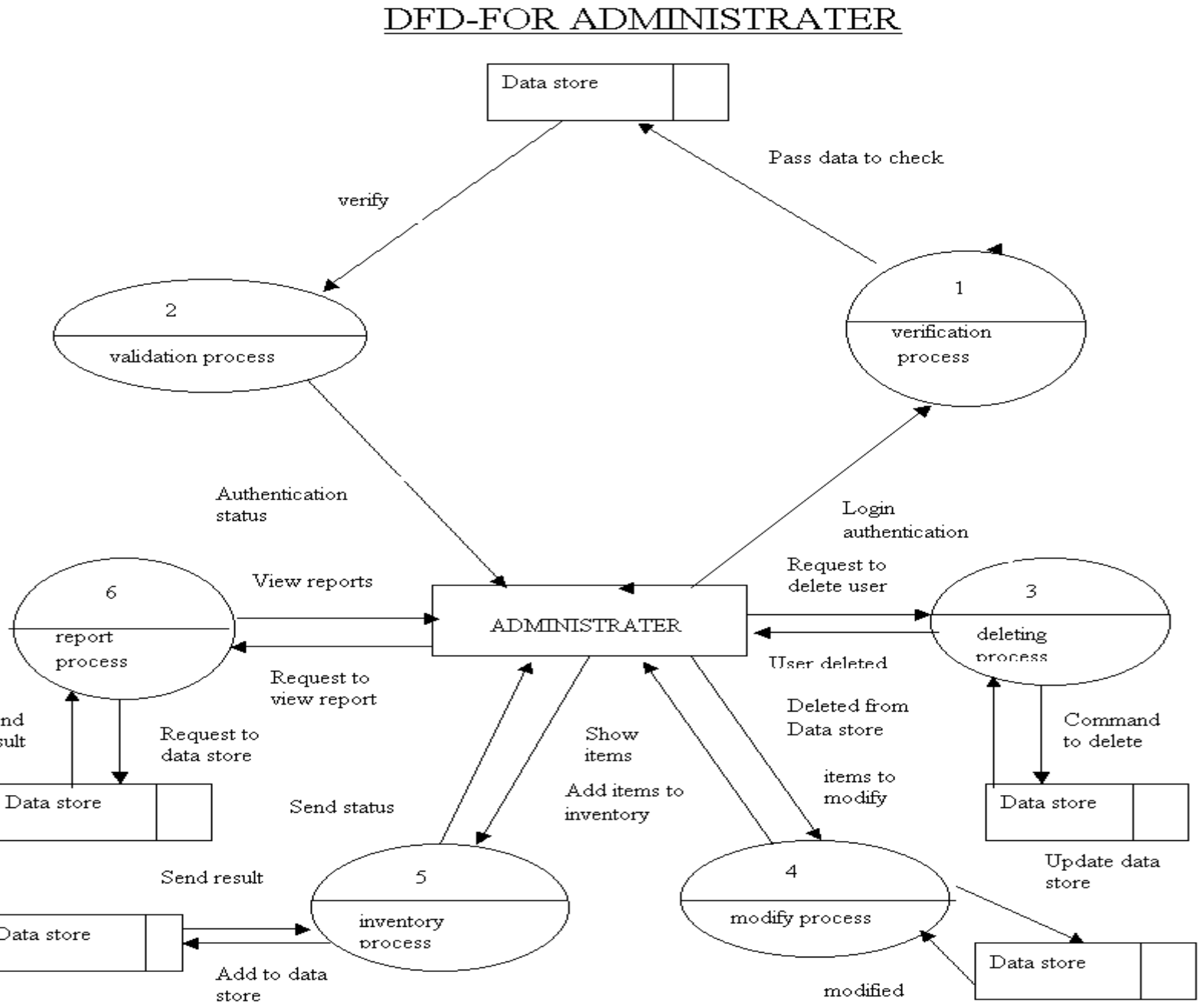


Figure 3.4 : DFD For User

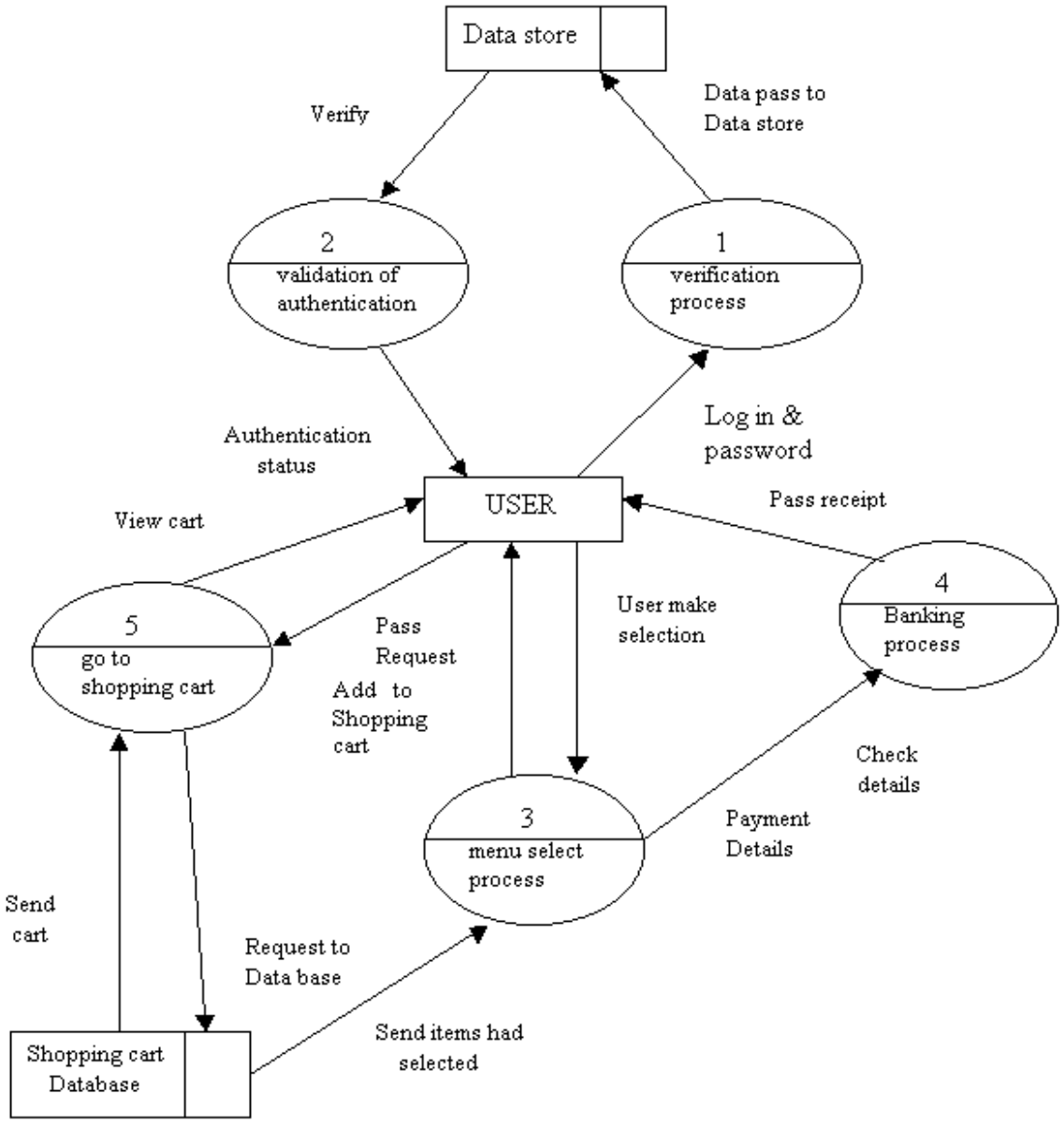
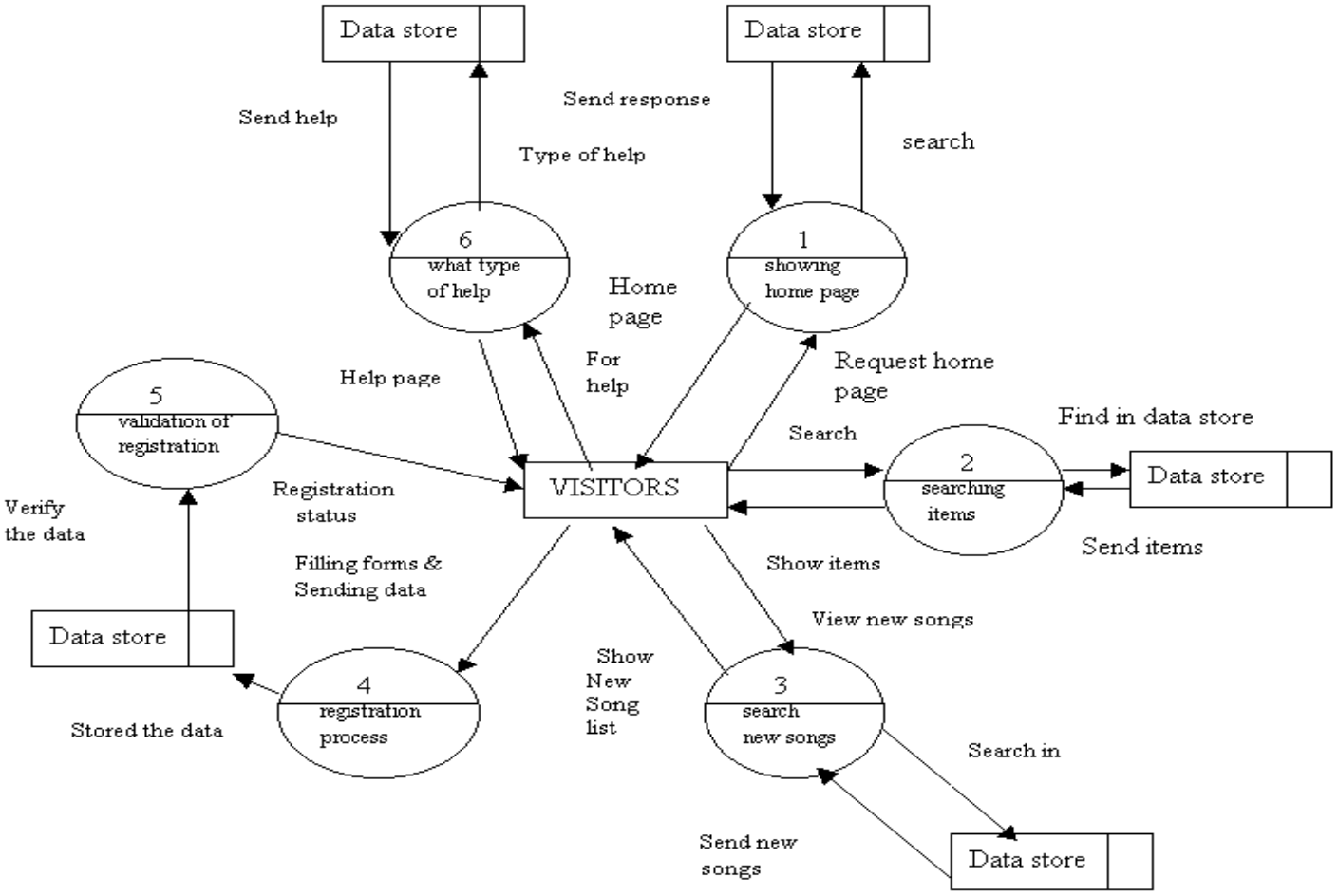


Figure 3.5 : DFD For Visitor

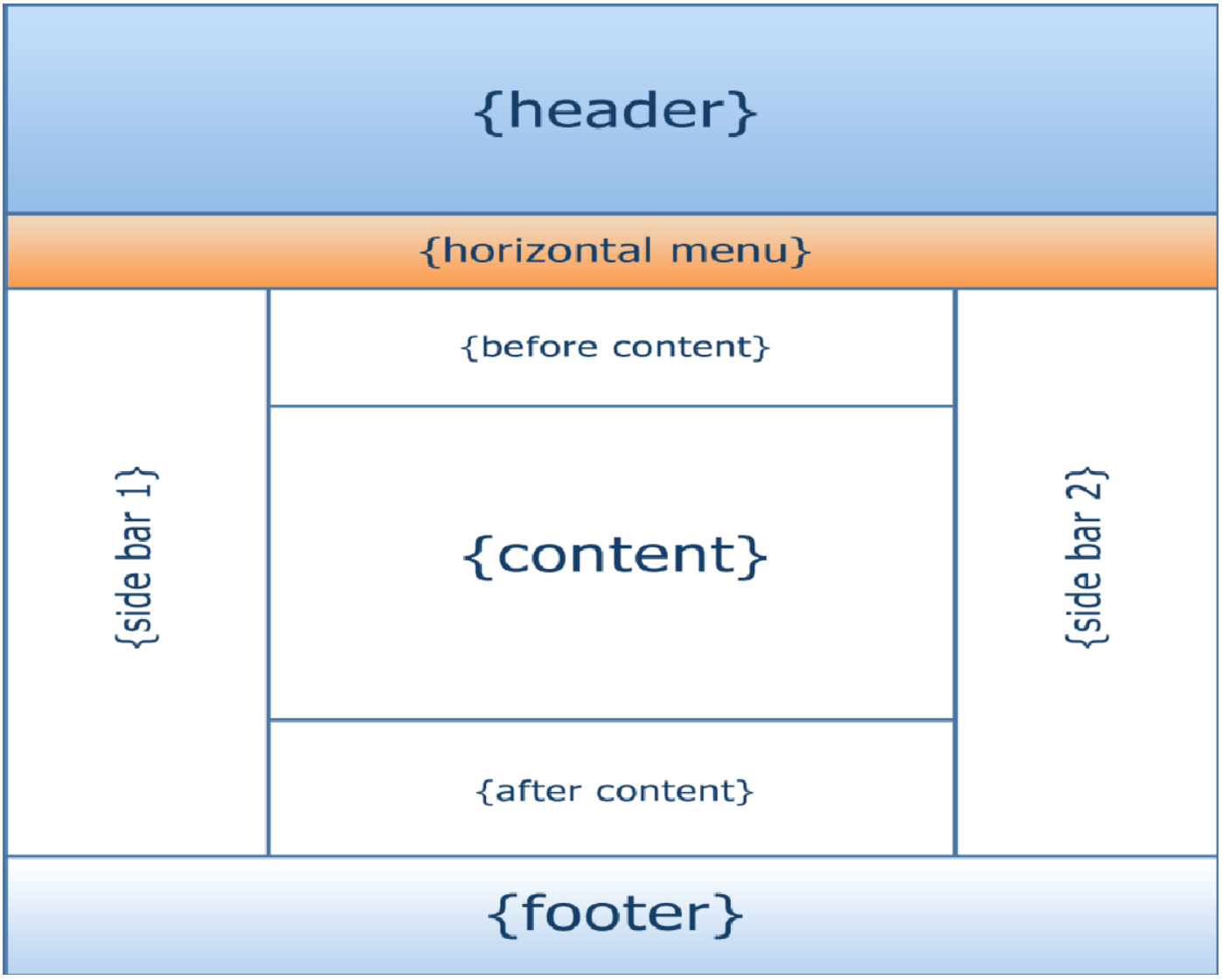


Chapter: 4 Tools and Technology used

1. **Html/Html5**

HTML means Hypertext Markup Language. This language is used in creating web pag-es. This language also supports other languages such CSS, PHP, JAVASCRIPT, etc. in creating interactive and responsive pages on the pages. HTML5 is just an updated ver-sion of the HTML. It supports new features, new attributes, new HTML elements, full CSS3 support, video and audio, 2D/3D graphics that help users and also help web developers to create new features easily on the website.

Figure 4.1 : HTML Structure



The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It is frequently assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997 A form of HTML, known as HTML5, is used to display video and audio, primarily using the <canvas> element, in collaboration with JavaScript.

2. **Cascading Style Sheets(CSS)**

CSS is simply referred to as Cascading Style Sheets.CSS is used to define styles for web pages, including the design, layout, and variations in the display for different de-vices and screen sizes./3/ The general structure of CSS

Basic syntax:

selector{property: value}

Example:
p{text-align: center;
color: black;
font-family: arial}

- CSS can be used in a separate style sheet or used in the webpage

Figure 4.2 : External CSS Structure

```
<! DOCTYPE html>
<html lang = "en-US">
<head>
<meta charset = "UTF-8">
<link rel = "stylesheet"  type = "text/css"  href =
"myStyle.css" />
</head>
```

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML) CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts[1] This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.[4]

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.[5]

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

Syntax:-

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties.

A style sheet consists of a list of *rules*. Each rule or rule-set consists of one or more *selectors*, and a *declaration block*.

Selector

In CSS, selectors declare which part of the markup a style applies to by matching tags and attributes in the markup itself.

Selectors may apply to the following:

- all elements of a specific type, e.g. the second-level headers h2
- elements specified by attribute, in particular:
- id: an identifier unique within the document, denoted in the selector language by a hash prefix e.g. #id

- class: an identifier that can annotate multiple elements in a document, denoted by a dot prefix e.g. .classname (the phrase "CSS class", although sometimes used, is a misnomer, as element classes—specified with the HTML class attribute—is a markup feature that is distinct from browsers' CSS subsystem and the related W3C/WHATWG standards work on document styles; see RDF and microformats for the origins of the "class" system of the Web content model)
    - elements depending on how they are placed relative to others in the document tree.

Classes and IDs are case-sensitive, start with letters, and can include alphanumeric characters, hyphens, and underscores. A class may apply to any number of instances of any element. An ID may only be applied to a single element.

Pseudo-classes are used in CSS selectors to permit formatting based on information that is not contained in the document tree. One example of a widely used pseudo-class is :hover, which identifies content only when the user "points to" the visible element, usually by holding the mouse cursor over it. It is appended to a selector as in a:hover or #elementid:hover. A pseudo-class classifies document elements, such as :link or :visited, whereas a *pseudo-element* makes a selection that may consist of partial elements, such as ::first-line or ::first-letter.[6] Note the double-colon notation for pseudo-elements versus single-colon notation for pseudo-class.

Selectors may be combined in many ways to achieve great specificity and flexibility.[7] Multiple selectors may be joined in a spaced list to specify elements by location, element type, id, class, or any combination thereof. The order of the selectors is important. For example, div .myClass {color: red;} applies to all elements of class myClass that are inside div elements, whereas .myClass div {color: red;} applies to all div elements that are inside elements of class myClass. This is not to be confused with concatenated identifiers such as div.myClass {color: red;} which applies to div elements of class myClass.

The following table provides a summary of selector syntax indicating usage and the version of CSS that introduced it

3. **Javascript**

   Javascript is a high-level language which could be used independently or inculcated into the webpage. It can be used to, handle requests and responses and also add dynamic be-havior and also store information on a website.

Figure 4.3 : Javascript Structure

```
<script ="javascript" type="text/javascript">

JavaScript code

</script>
```

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles.

JavaScript's dynamic capabilities include runtime object construction, variable parameter lists, function variables, dynamic script creation (via eval), object introspection (via for...in and Object utilities), and source-code recovery (JavaScript functions store their source text and can be retrieved through toString()).

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about APIs that are specific to Web pages, please see Web APIs and DOM.

The standards for JavaScript are the ECMAScript Language Specification (ECMA-262) and the ECMAScript Internationalization API specification (ECMA-402). As soon as one browser implements a feature, we try to document it. This means that cases where some proposals for new ECMAScript features have already been implemented in browsers, documentation and examples in MDN articles may use some of those new features. Most of the time, this happens between the stages 3 and 4, and is usually before the spec is officially published.

Do not confuse JavaScript with the Java programming language — JavaScript is *not* "Interpreted Java". Both "Java" and "JavaScript" are trademarks or registered trademarks of Oracle in the U.S. and other countries. However, the two programming languages have very different syntax, semantics, and use.

JavaScript is a compact, object-based scripting language for developing client and server Internet applications. Netscape Navigator interprets JavaScript statements embedded in an HTML page, and Livewire enables you to create server-based applications similar to Common Gateway Interface (CGI) programs. This book describes the JavaScript language and its use in Navigator. For information on developing server-based JavaScript applications, see the *Livewire Developer's Guide.*

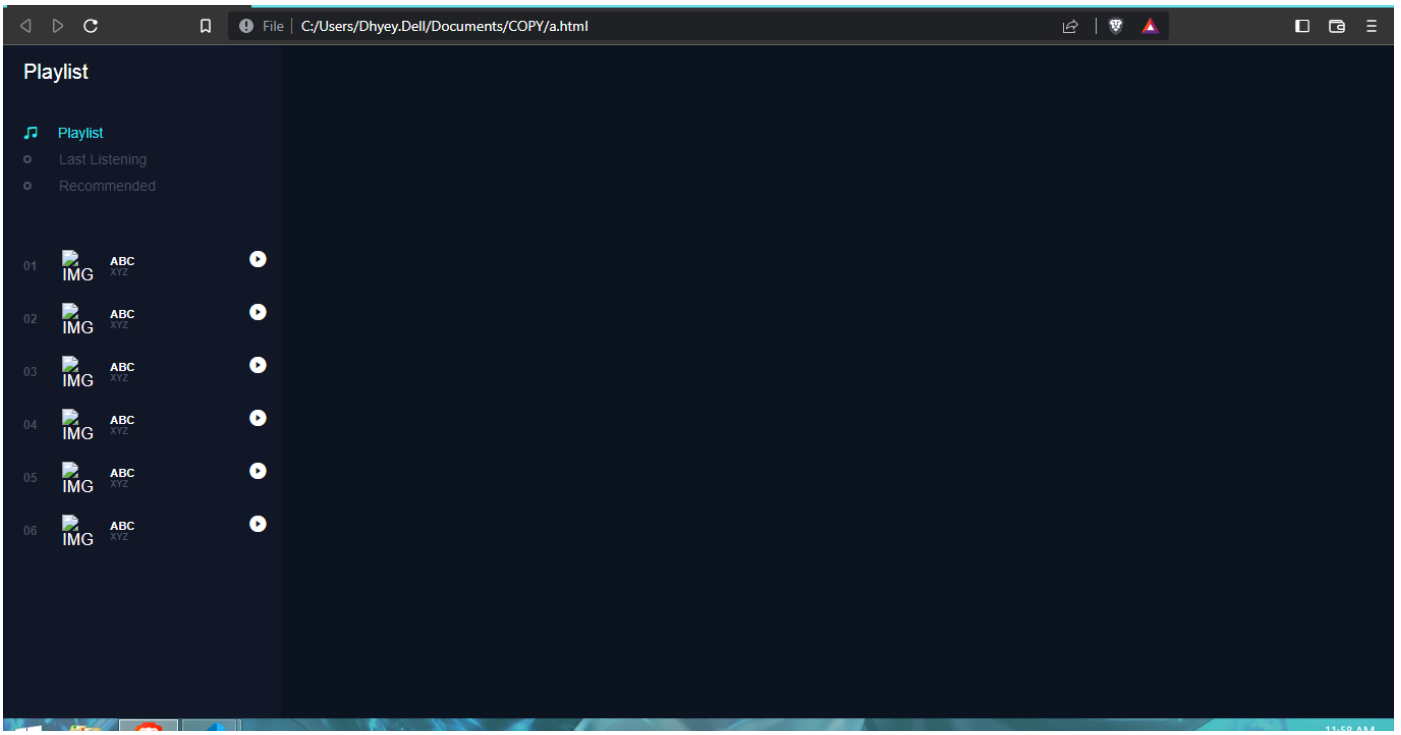
Client-side JavaScript statements embedded in an HTML page can respond to user events such as mouse-clicks, form input, and page navigation. For example, you can write a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code. Without any network transmission, the HTML page with embedded JavaScript can check the entered data and alert the user with a dialog box if the input is invalid.

<SCRIPTLANGUAGE="*JavaScriptVersion*">
JavaScript*statements...*
</SCRIPT>

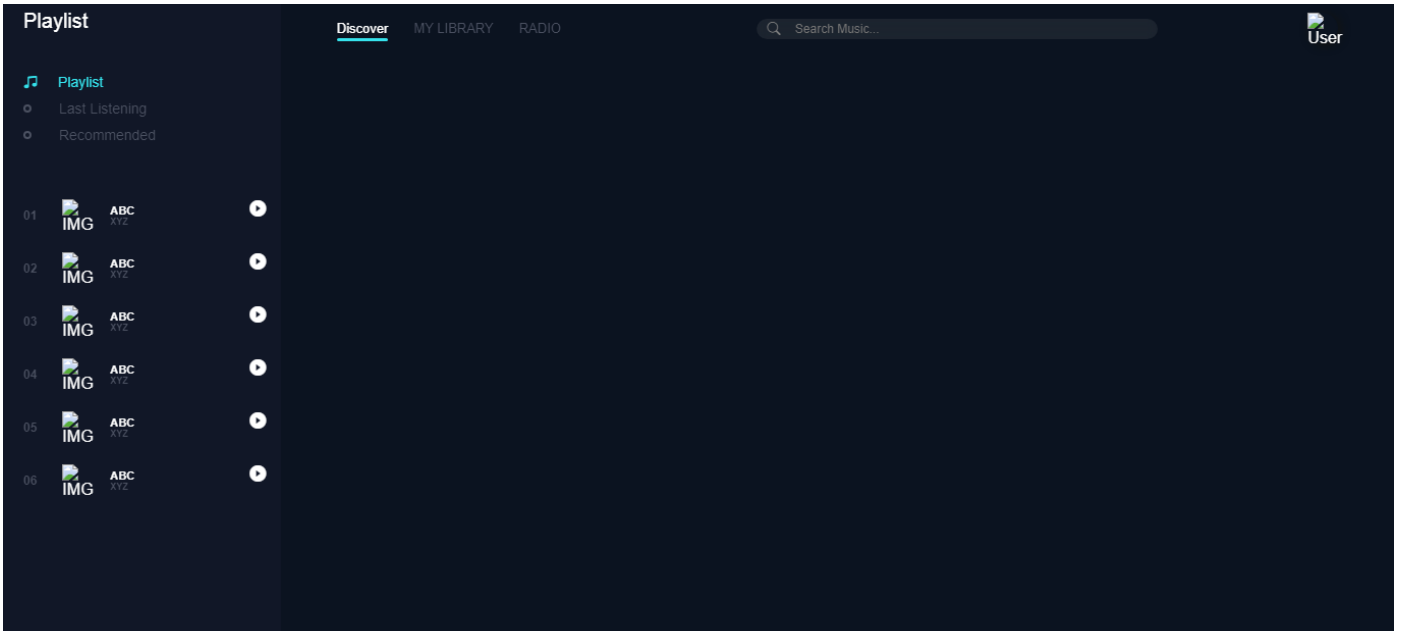
Since browsers typically ignore unknown tags, non-JavaScript-capable browsers will ignore the beginning and ending SCRIPT tags. All the script statements in between are enclosed in an HTML comment, so they are ignored too. Navigator properly interprets the SCRIPT tags and ignores the line in the script beginning with the double-slash (//).

Chapter: 5 Snapshots

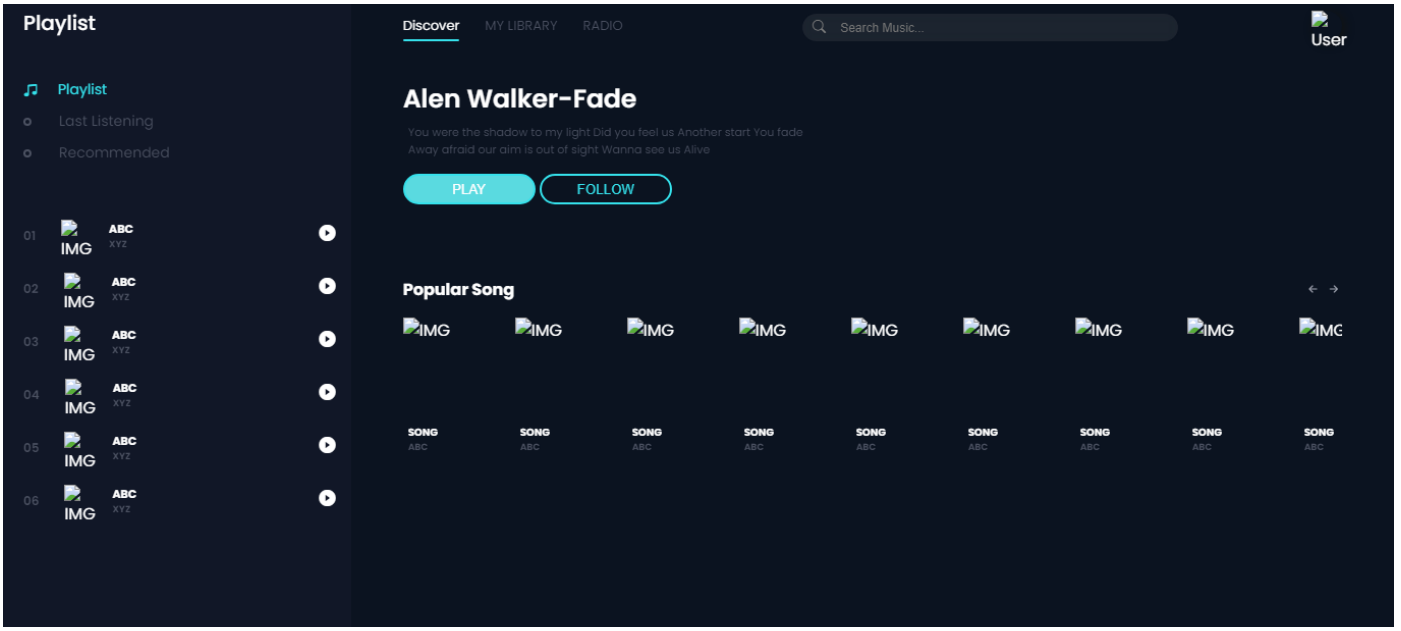
1. Side Bar :



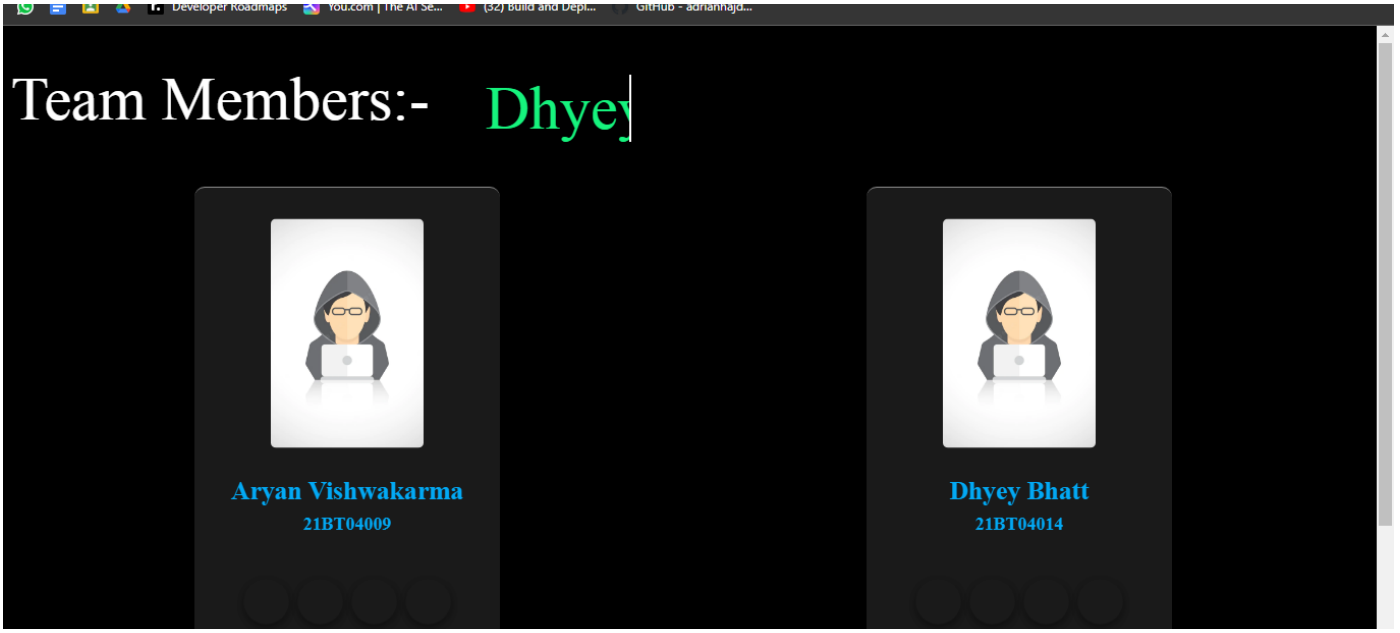
2. Navigation Bar :



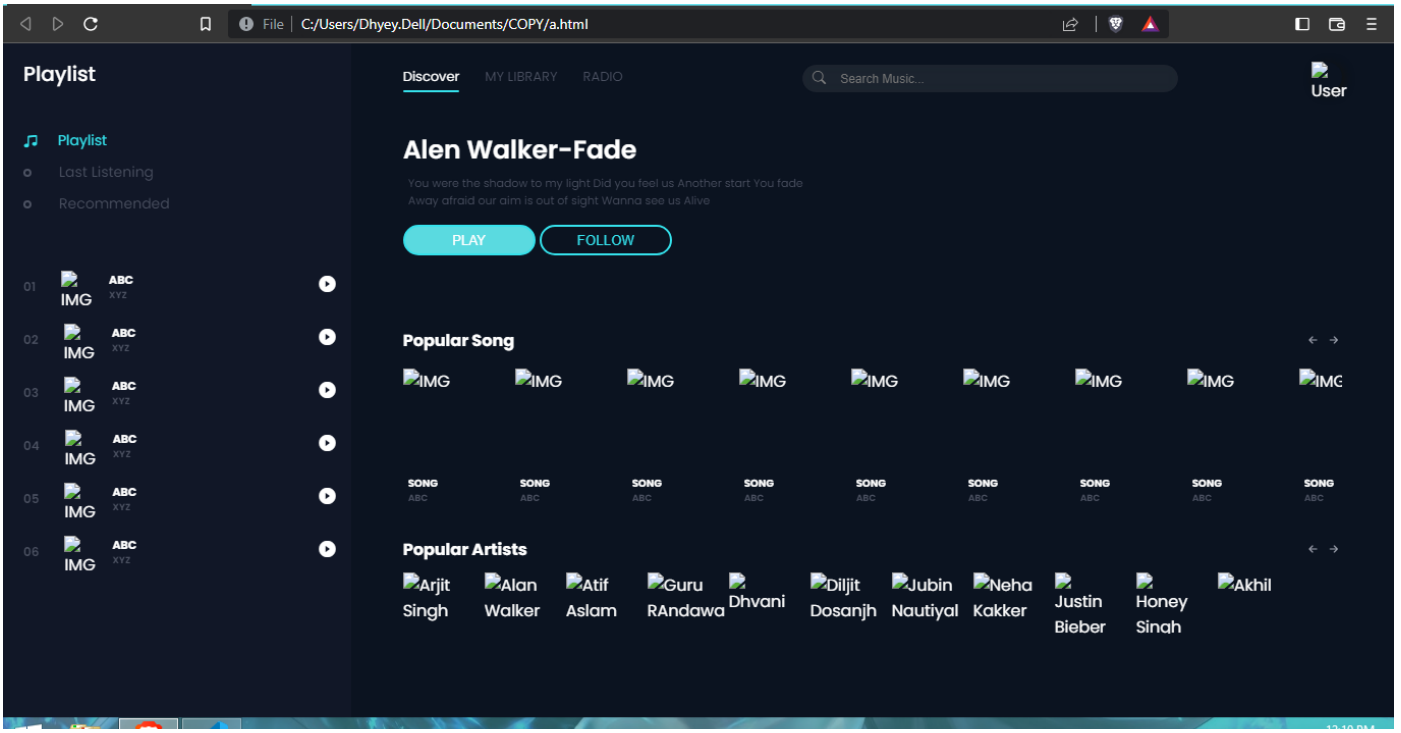
3. Popular Song Section :



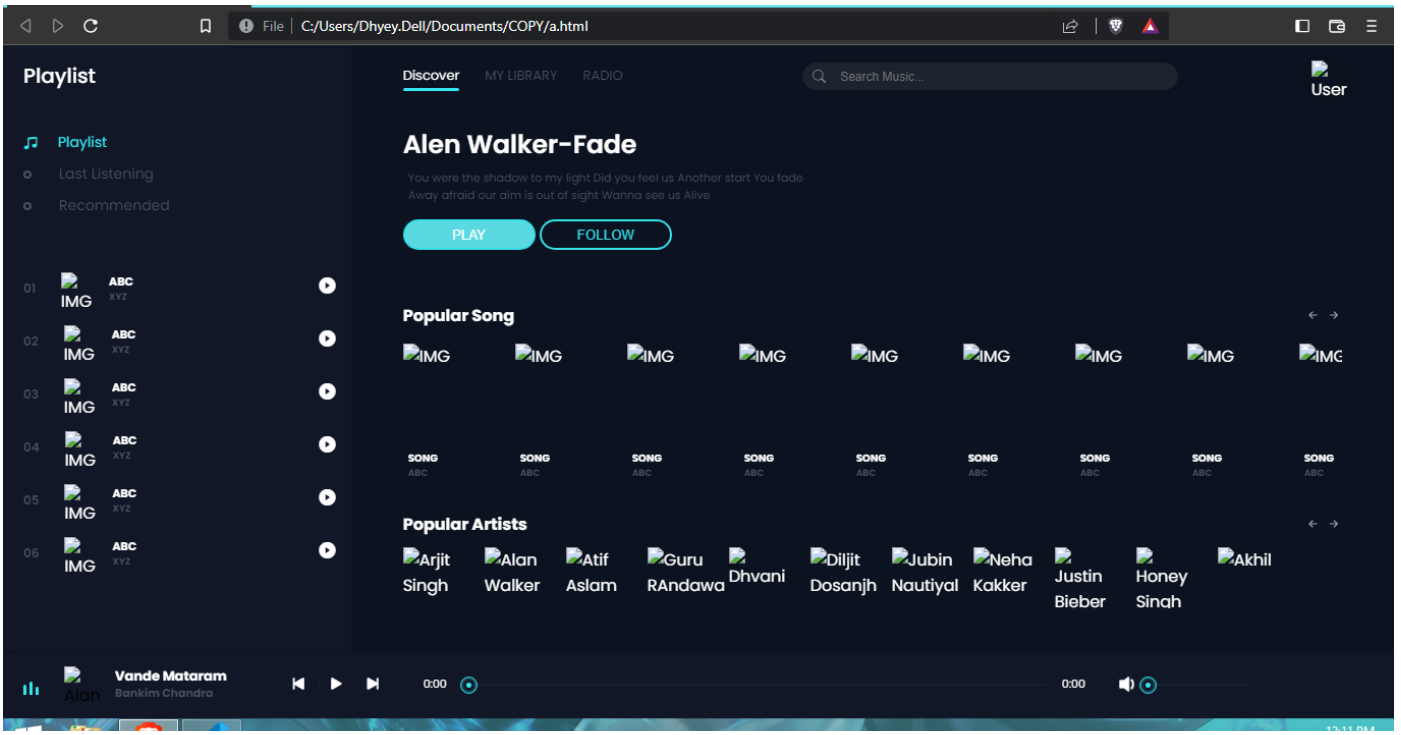
4. Contact us Page :



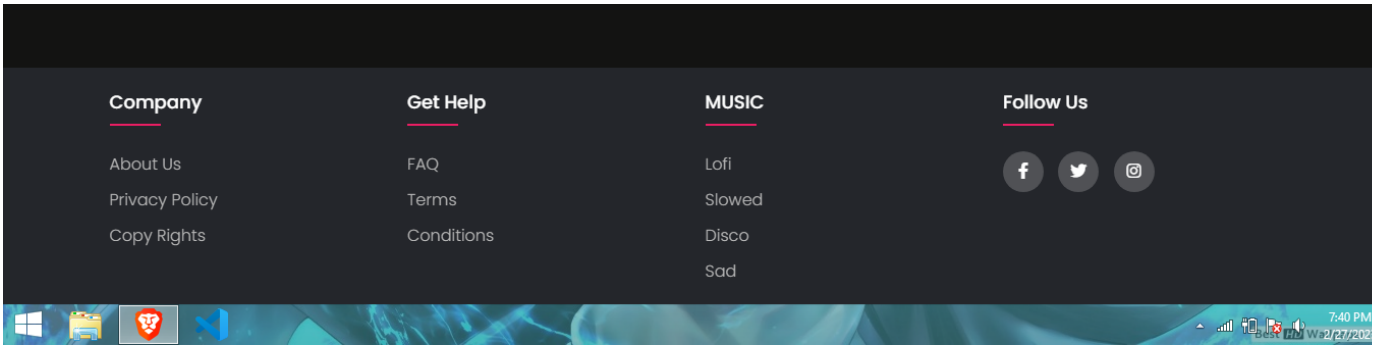
5. Popular Artists :



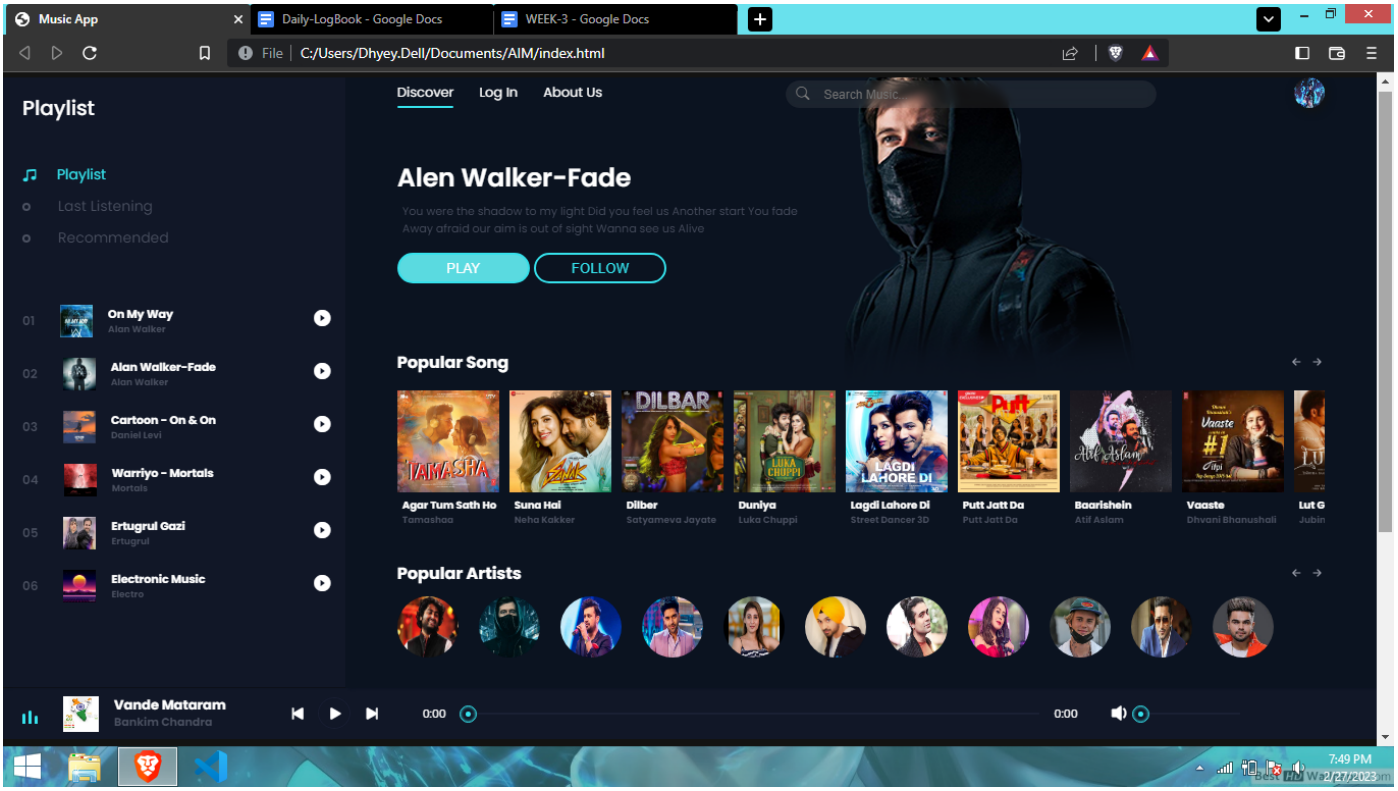
6. Master Play :



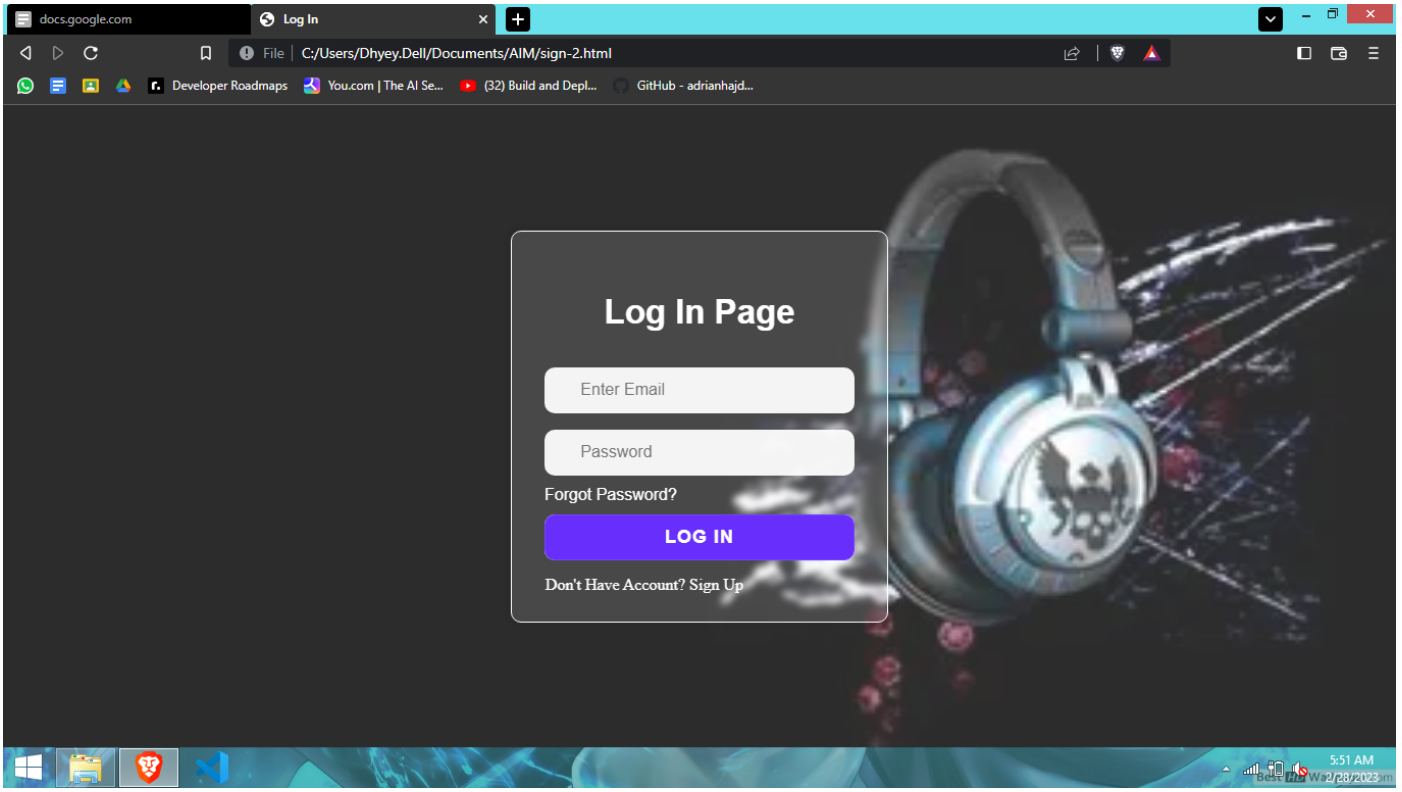
7. Footer :



8. Home Page :



9. Log In/Sign Up Page:-



Chapter: 6 Observations

- **Solution Assurance** – This will encompass solution architects from the customer side as well as solution architects from the various implementation partners. It will be responsible for governance and will provide the steer if any cross-team design issues arise.

- **Functional Design –** This team will be responsible for the design and preparation of the functional and integration specification documents.

- **Technical Design & Development –** This team will be responsible for the technical design and preparation of technical specification documents. Plus also the subsequent development based on the functional/ integration specification documents. This team will comprise of technical architects, back-end technical consultants and Integration experts.

- **Quality Assurance & Control –** Will be responsible for putting the test plan, monitoring strategy, defect resolution SLAs, quality standard of the deliverables, etc. They must liaise with the functional team to understand the functionality and to write the test scripts.

- **Project Management –** Will be responsible for building the project plan and keeping it updated in line with the project status, stake holder updates, co-ordination across teams, status review meetings, RAID Logs, dependency logs, monitoring the various project gates etc. The PM should have prior experience in managing ‘integration’ projects – otherwise, it just doesn’t work.

- **Team Leads/ Project Leads/ Scrum Masters –** To manage a combination of resources in a specific part of the overall project – It could be based on systems, functional area etc.

Chapter: 7 Results and Discussions

❖ Further Improvement

Although this application meets all the feature requirements as planned, there are still some aspects needed to improve. The main improvement would be some CSS styling. Color design should also be taken into account to enhance the user experience.

Despite of all the fundamental and useful implemented functionalities in the application, these features in the following list can be considered for further development:

- User sign up using Gmail or Facebook
- Email notification after successful payment
- Each product has a review and rating section

Eventually, in order for this e-commerce application to switch from development mode to production mode, testing and deployment have to be handled appropriately as the development process may have some unseen bugs.

As changes are always necessary in future it applies to software development also but these changes should be appreciable in nature. These appreciable changes will make the software to fight for its survival in the competitive market. Hence it is necessary to think about the future enhancements at present.

The system 'Online Music Store' will fulfill the entire requirement of the clients. The system is developed according to the present requirements of the company. The system is developed as easy as possible for the sake of end users.

One drawback of my system is that the client cannot view, search and purchase music according to a particular language option .By the next time I would like to add this facility.

By the next time I would like to add two more modules: Purchase Module and Accounting Module. Purchase Module deals with purchasing activities of music related items. Accounting Module deals with all accounting activities such as billing, ledger preparation, balance sheet preparation, profit and loss account preparation etc.

In the present system transaction is through a particular bank or through money orders. In future I would like to make it through credit cards. Credit card validation techniques are needed for that.

Chapter: 8 Conclusion & Future scope

The website was successfully developed at the end. A fully functional end-to-end e-commerce application featuring an online bookstore was released and aimed to help the startup develop their business strategy in general. This application was meant to solve the problem that is mentioned in the first section of this thesis: to help the book retailer startup become more widely known and gain more potential customers. People and book lovers also have one more source to expand their knowledge.

This project helps in understanding the creation of an interactive web page and the technologies used to implement it.

Although the application still has some drawbacks and needs more further improvements, both in styling issue and new features, it is a combination of one of the most widely used web stack technology with one of the most emerging business ideas nowadays.

As stated in the introduction the minimum target at the beginning of this project was to show the core functionalities in a user friendly web page. During the implementation the team reached their limits and succeeded in replacing these limits several times. This section will illustrate what the team reached and which problems occurred during the project. It will describe where the planning was realistic and will also give recommendations for similar or further projects. The core functionality was reached in the following parts. It is possible to insert a new customer, change the details of a customer and delete a customer. Also the product part is implemented in the same way. A new product can be inserted, changed and deleted. Furthermore the order section is completely implemented. It is possible to fill in one order for several products. This order also can be changed and deleted. The team did not succeed in implementing all the artist and song functionality which was planned. Reasons for that are the lack of experience and time as well as the wish to implement additional features in order to enrich the application from a different angle. The additional features that were reached are creating the overviews of the entities, and error checking. These offer extra functionality to the clerks and the managers of the Music Store. The team was intent on achieving a well designed ‘Insert new order’ window. Through this challenge a window was designed in a way that enables the user to simultaneously select products and customer, and view the details of the selected, in the same window. The application also provides error checking: this means every input is tested of validity.

References

1. About Javascript
   Available at: https://developer.mozilla.org/en-US/docs/Web/JavaScript

2. About CSS
   Available at: https://en.wikipedia.org/wiki/CSS

3. About HTML
   Available at: https://www.w3.org/html/wiki/Educational_materials/What_is_HTML

4. About Project
   Available at: Create A Music Website Using HTML CSS JAVASCRIPT | Add Music In HTM…