**ABSTRACT:**

Cars are more than just a ton of steel and engine. **Web development** is the work involved in developing a Web Site for the Internet (World Wide Web) or an Intranet (a private network). This project includes creating a fully responsive webpage with the help of web development techniques. This project is about creating a webpage for a car dealership that buys and sells a wide variety of cars. **Front-end web development** is the development of the Graphical user interface of a website, through the use of HTML, **CSS**, and **JavaScript**, so that users can view and interact with that website. This webpage includes various components ranging from basic **HTML** tags to complex **Bootstrap (CSS framework)** codes. The goal of a front end Web developer is to create a webpage with which the users can interact. This project includes the use of various components such as carousel, cards, navbar, footer, etc which are required to create a complete webpage.

Keywords: Web Development, Cars, responsive webpage, Front end web development.

**Table of contents:**

1. Introduction
2. Existing Method
3. Proposed Method with Architecture
4. Methodology
5. Implementation
6. Conclusion

**Introduction :**

**Front-end web development** is the development of the graphical user interface of a website, through the use of HTML, CSS, and JavaScript, so that users can view and interact with that website. There are several tools and platforms, such as WordPress, Joomla, and Drupal, available that can be used to develop the front end of a website.

However, the basics of web development are stated below:

1. **Hyper Text Markup Language**

Hyper Text Markup Language (HTML) is the backbone of any website development process, without which a web page does not exist. Hypertext means that text has links, termed hyperlinks, embedded in it. When a user clicks on a word or a phrase that has a hyperlink, it will bring another web-page. A markup language indicates text can be turned into images, tables, links, and other representations. It is the HTML code that provides an overall framework of how the site will look. HTML was developed by Tim Berners-Lee. The latest version of HTML is called HTML5 and was published on October 28, 2014 by the W3 recommendation. This version contains new and efficient ways of handling elements such as video and audio files.

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

Cascading Style Sheets (CSS) controls the presentation aspect of the site and allows your site to have its own unique look. It does this by maintaining style sheets which sit on top of other style rules and are triggered based on other inputs, such as device screen size and resolution.

### JavaScript

JavaScript is an event-based imperative programming language (as opposed to HTML's declarative language model) that is used to transform a static HTML page into a dynamic interface. JavaScript code can use the Document Object Model (DOM), provided by the HTML standard, to manipulate a web page in response to events, like user input.

Using a technique called AJAX, JavaScript code can also actively retrieve content from the web (independent of the original HTML page retrieval), and also react to server-side events as well, adding a truly dynamic nature to the web page experience.

### Web Assembly

Web Assembly, supported by all the major browsers (i.e., from the major vendors Google, Apple, Mozilla and Microsoft), is the only alternative to JavaScript for running code in web browsers (without the help of plug-ins, such as Flash, Java or Silverlight; all being discontinued, as browsers are dropping plug-in support). Prior to its adoption, there was asm.js (a subset of JavaScript; and thus, strictly works in all browsers), that's also used as a compiler target with efficient support in browsers such as Internet Explorer 11; and for such browsers that do not support Web Assembly directly, it can be compiled to asm.js and those browsers supported that way. Generally speaking, programmers do not program in Web Assembly (or asm.js) directly, but use languages such as Rust, C or C++ or in theory any language, that compile to it.

**Existing Method:**

STEP-1 Start with HTML to design the basic structure of the webpage.

Hypertext Markup Language, or HTML, is a programming language used to describe the structure of information on a webpage. A webpage can contain headings, paragraphs, images, videos, and many other types of data. Front-end developers use the HTML element to specify what kind of information each item on a webpage contains — for instance, the “p” HTML element indicates a paragraph. Developers also write HTML language code to specify how different items relate to one another in the page’s overall structure or document structure. Every website we open in our web browser, from social networks to music services, uses HTML. A look under the hood of any website would reveal a basic HTML code page, written with an HTML structure editor, providing structure for all the page’s components, including its header element, footer element, main content, and other inline elements. HTML text lets us indicate the roles of different structural elements to search engines and other services that index the content and summarize it for other users. For instance, marking the caption of an image with the “fig caption” element and enclosing the image and its caption in the “figure” meta element helps a search engine understand that these two pieces of content are related and that the caption describes the associated image.

STEP-2 Add the required CSS to the elements for the styling.

CSS stands for Cascading Style Sheets. It is the coding language that gives a website its look and layout. Before the development of CSS in 1996 by the World Wide Web Consortium (W3C), Web pages were extremely limited in both form and function. Early browsers presented a page as hypertext - plain text, images and links to other hypertext pages. There was no layout at all to speak of, merely paragraphs running across the page in a single column.

CSS allowed several innovations to webpage layout, such as the ability to:

* Specify fonts other than the default for the browser
* Specify color and size of text and links
* Apply colors to backgrounds
* Contain webpage elements in boxes and float those boxes to specific positions on the page

They put the "style" in style sheets, and for the first time, Web pages could be designed.

**How CSS works:**

To understand the basics of how CSS works, you must first understand a little about modern HTML. Web developers lay out pages according to the "box model." A Web page is a series of boxes, each containing a discrete element. These boxes are nested, one inside another.

For example, a page's header is a box, and it contains several smaller boxes comprising all the elements that make up a header: logo, navigation, social media buttons, shopping cart buttons, etc. Using CSS, a developer assigns styles to the "header" box. In this example, let's assume that the developer makes the text inside the header purple, Arial font and fifteen points high.

Here's where the "cascading" part of cascading style sheets comes into play. The font styles applied to the header cascade down to all the elements contained inside the header. Elements containing text such as navigation, links or calls to action will all be purple, Arial and fifteen points high.

STEP-3 Add the required JavaScript to make the page interactive.

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user. Common examples of JavaScript that you might use every day include the search box on Amazon, a news recap video embedded on The New York Times, or refreshing your Twitter feed.

Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one. To recap, JavaScript adds **behaviour** to web pages.

JavaScript allows users to interact with web pages. There are almost no limits to the things you can do with JavaScript on a web page – these are just a few examples:

* Show or hide more information with the click of a button
* Change the colour of a button when the mouse hovers over it
* Slide through a carousel of images on the homepage
* Zooming in or zooming out on an image
* Displaying a timer or count-down on a website
* Playing audio and video in a web page
* Displaying animations
* Using a drop-down hamburger menu

**Proposed Method with Architecture**

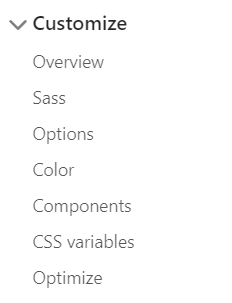
First and third step remains the same as per the previous section but in second step instead of using CSS for the styling we use it’s framework that is **Bootstrap**.

**BOOTSTRAP:**

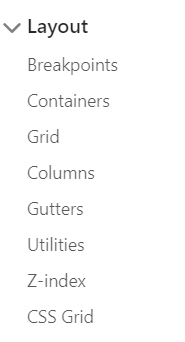
Quickly design and customize responsive mobile-first sites with Bootstrap, the world’s most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins. Along with CSS it’s library has pre-defined plugins for JavaScript.

To use **Bootstrap** first include the starter template.

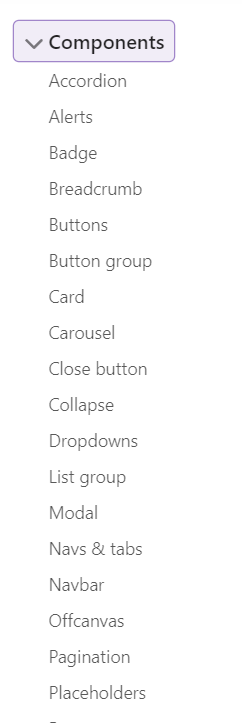
Then click and **Customise** option and choose your desired feature.



For using various layout properties such as containers, grid, columns, etc click on layout option.



And then for adding various components to your web page such as buttons, alerts, cards, etc click on the components option which gives a wide list to choose from.



And many more…

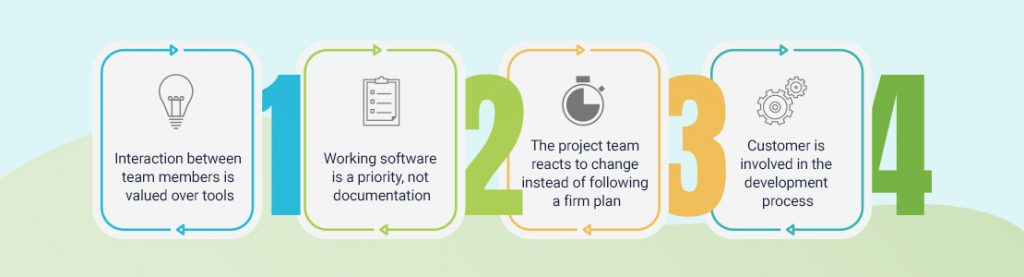
**Methodology**

There are a lot of methodologies used for Web Development, such as:

## **Agile**

Agile is another methodology widely used for web app development projects. The Agile method is often used for the projects with no definite requirements and limited short timeframes.

The key characteristics of the Agile methodology are the following:



There’s no pre-planning of the entire project, so the tasks are accomplished in an adaptive manner. The teams’ efforts are concentrated on small tasks that require urgent attention. Agile approach also supports quick modifications in the project’s scope and direction based on the market changes.

Agile is better suited to manage complex web development projects that involve variability. It allows teams to develop in short iterations that have a specified timeframe and deliverables but don’t have a specific order.

## **Scrum**

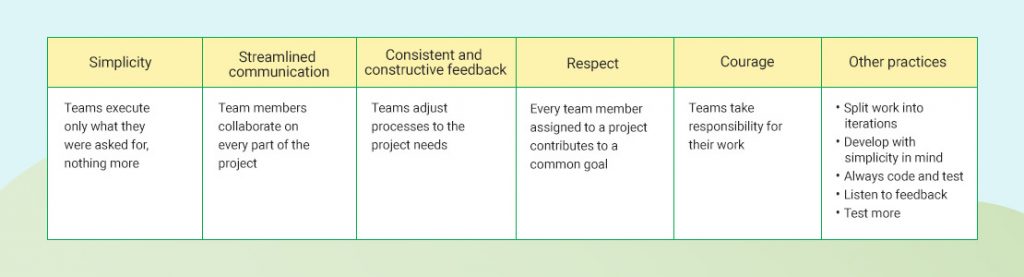
Scrum shares Agile’s fundamental philosophy that development teams should collaborate daily and intensively. Scrum uses an iterative approach to web development where the team is a key. This approach requires self-management and self-organization which means that it will be a good fit for smaller projects with experienced and organized team members.

Scrum combines the flexibility and iterative practices of the modern-day Agile method and the discipline and structure of more conventional methodologies. End goals are broken into smaller ones and the teams are supposed to use fixed-length iterations (typically they are two-week-long) to develop software and present it to the client. Meetings are crucial here and during each iteration, teams conduct daily meetings to track progress and get feedback. Such iterative approach allows quick changes and increases effectiveness for complex projects.

## **Extreme Programming**

Extreme Programming (XP) is one more Agile framework that aims to produce high-quality web apps and responsiveness to changing customer requirements. Just like many other Agile methodologies, XP is focused on regular releases in short iterations, enabling change when needed.

XP is more than just a sequence of steps to manage web development projects – it adheres to a set of values that are supposed to help teams work more efficiently. These values include:



This approach implies that the development team should first understand the project’s needs well enough to give the client feedback about the technical aspects of how the problem can be solved.

XP was designed to help development teams adjust to the fast-changing requirements. So, this approach will suit small teams working on a new system, especially when a client sets tight deadlines for a project.

## **Lean**

Lean is more than just a workflow methodology – it’s a mindset too. Originating from manufacturing, as a way to optimize the production line to minimize waste and maximize value to the customer, lean approach is widely used in various spheres, including web application development.

Lean means creating more value for customers with fewer resources involved. Lean’s basic principles are the following:

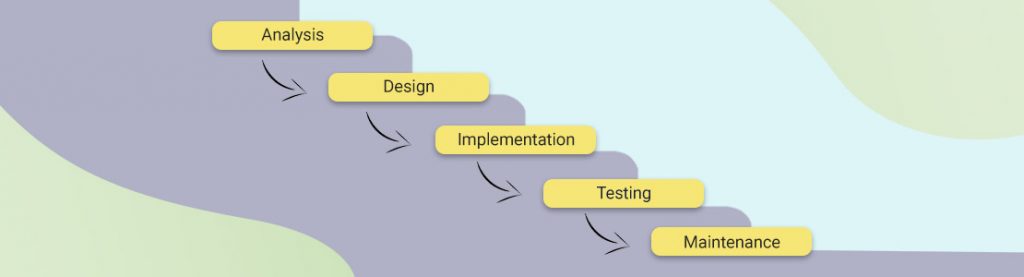


Waste is defined as anything capable of impairing code quality, hindering time and effort, and reducing the delivered business value. These may be issues like excessive code or functionality, development delays, vague requirements, and insufficient testing. The Lean methodology focuses on eliminating these issues, learning the required technologies, and identifying the real needs of the project.

## **Waterfall**

Waterfall is the most conventional software development methodology. In fact, it has been one of the most popular approaches for web development projects for several decades due to its plan-driven approach.

The Waterfall approach requires a lot of structure and documentation. The process is divided into several stages that form a sequence:



The first stage is critical and requires a complete understanding of the project’s demands and scope by both the developers and the product owners.

Waterfall lacks flexibility which means that every phase should be fully completed before moving forward to the next stage. Once any modifications are needed or any errors are detected during the course of the project, Waterfall will require a full restart. As a result, projects managed under the Waterfall method might require much more time. On the other hand, it is great for ensuring that all deliverables meet expectations and it allows for easily measuring the progress since you see the full scope of the project in advance.

Out of all of the above mentioned methodologies I used **WATERFALL** methodology as:

#### **Stage 1. Discovery and project research**

Discovery means that the development team only investigates client’s business requirements and decide for themselves whether they can really solve a client’s problem. An outcome is getting to know a client's goal, business specifics, and initial requests.

If everything is good and the development team can help the client, the PM passes the tech documentation and specifications to the Lead Developer, makes the project aims and requirements clear.  
  
Afterward, the Lead Developer does the report to the PM, who, in his/her turn, discusses all the features and details with the client.

#### **Stage 2. Wireframes and prototypes creation**

Wireframing is designing the webpage before actually starting to code, it is basically creating the layout of the webpage and deciding the position of various components in the page.

#### **Stage 3. Design**

This is the time when the actual designing of the webpage starts and all the various styling is added to the page components.

#### **Stage 4. Development**

Initial installation and configuration go first. Then developers set up all necessary settings of modules. They make sure every website page was approved, and the client has gone through demo versions of every feature on the website. It was a little hint from the development side to the client: check and test all the options. There are no minor ones when it comes to website usage. Believe us, your website’s visitors will notice every teeny-tiny bug.

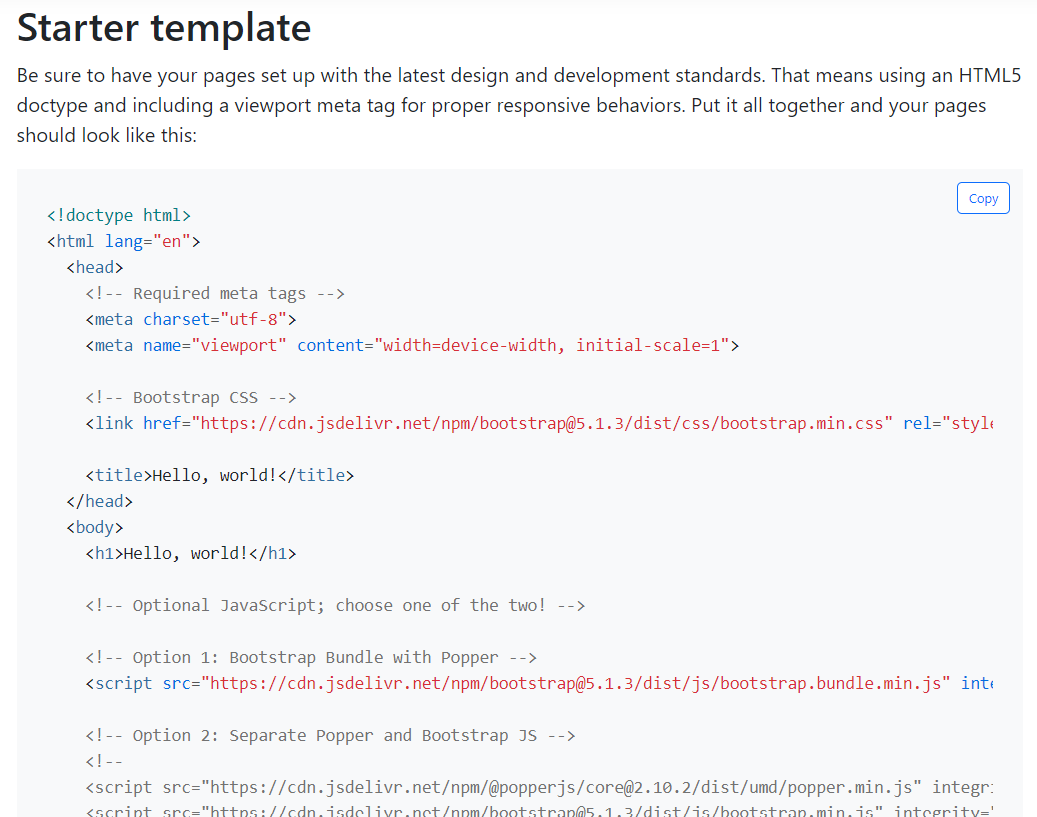
Depending on the project, the front-end and back-end development can go either in parallel or the back-end is followed by the front-end. A front-end developer implements all visual features and makes sure everything is pixel-perfect, and that a website is cross-browser compatible.

#### **Stage 5. Quality assurance**

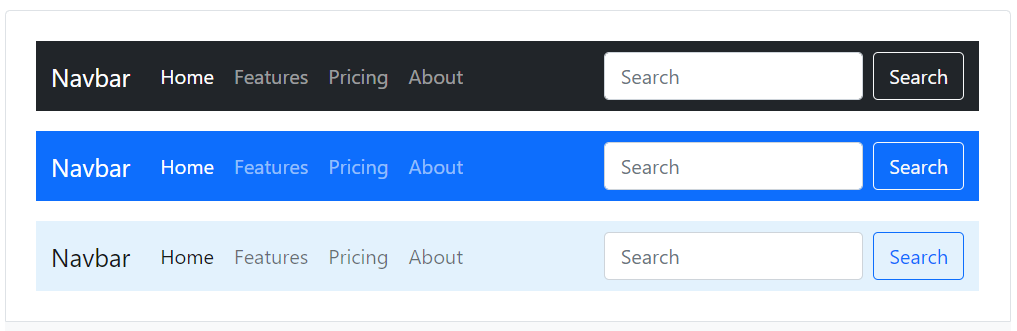
Never ever omit the testing stage. After the integration test, we move to the functional and UI tests and end up with manual tests.

**Implementation:**

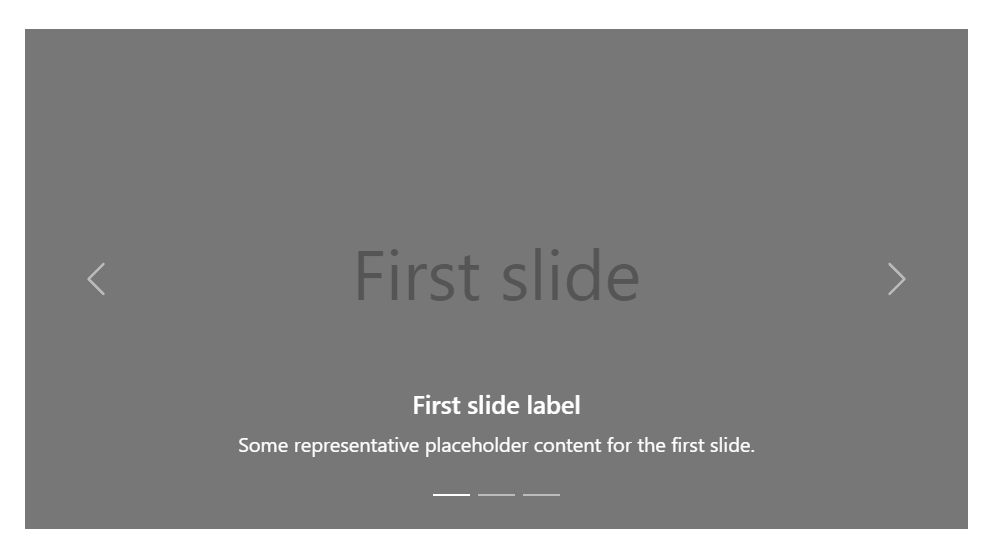
Started to code with a starter template of bootstrap.



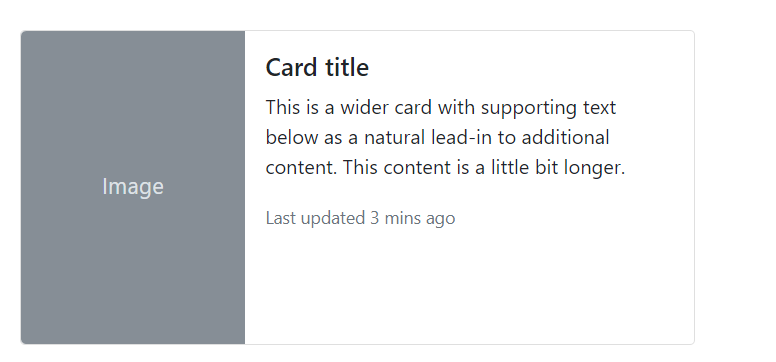
Then added a navbar and edited it’s contents as per requirements.



Then added a carousel with custom options.



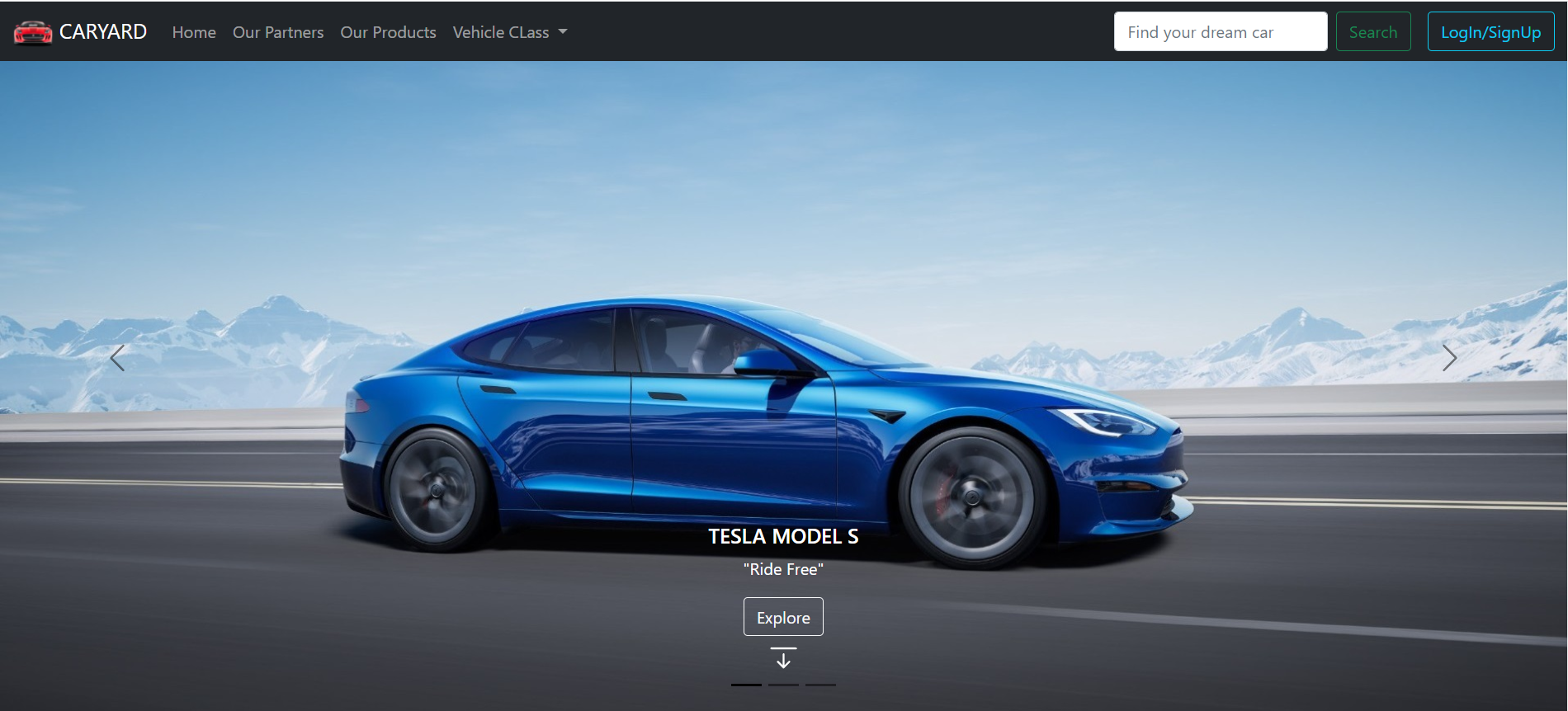
Later added various cards and also some custom items for different sections of the webpage.

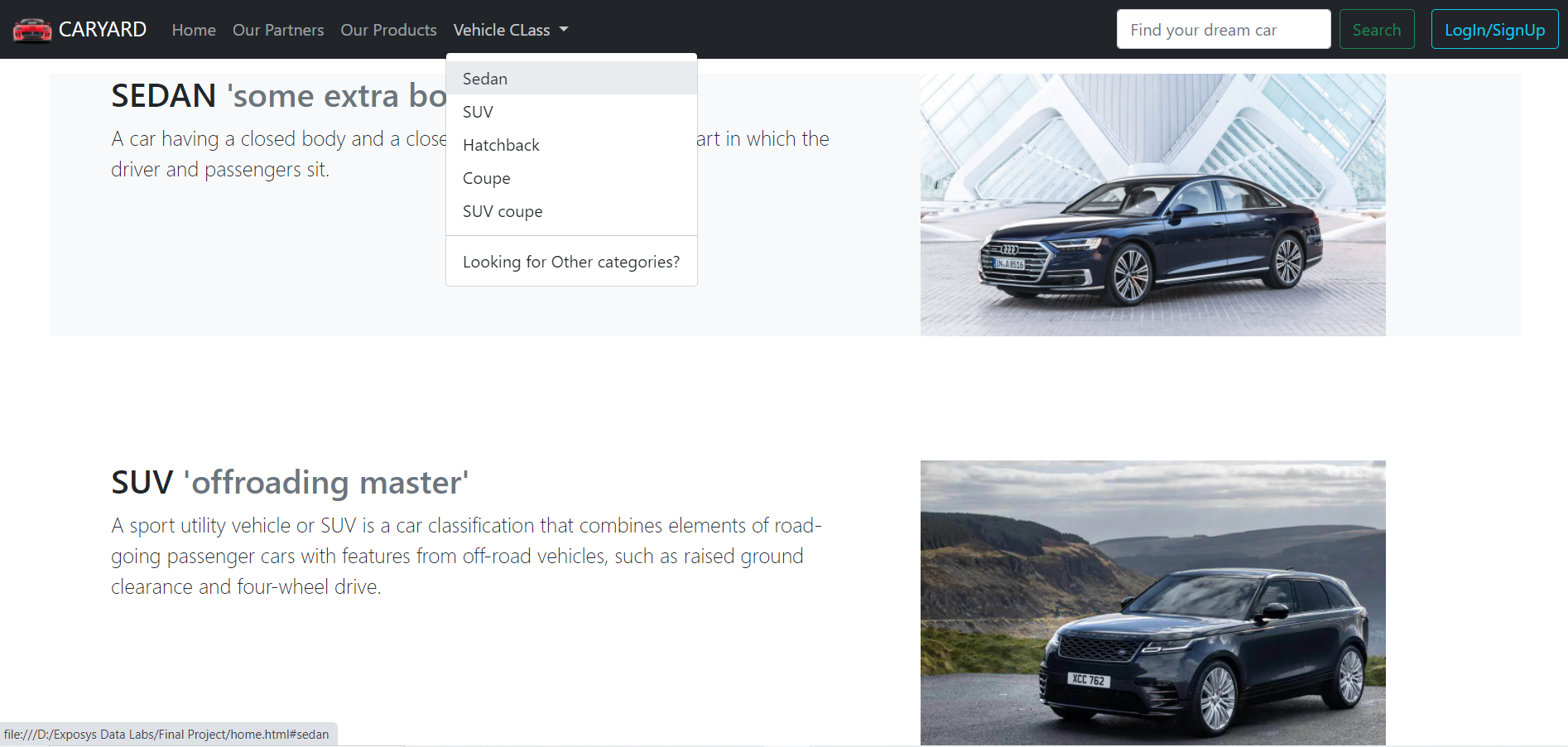


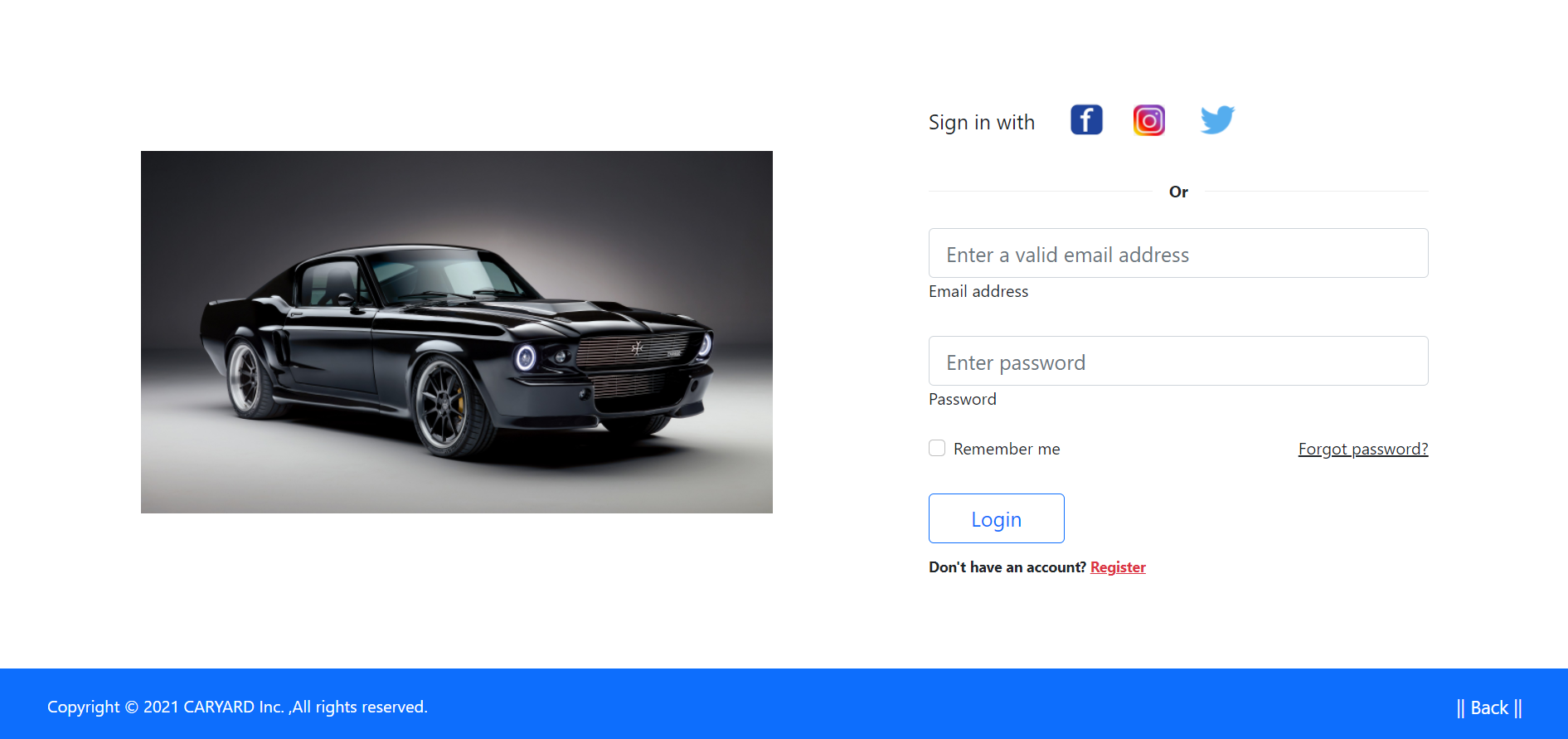
Ended the page with a footer.

And did the same for all the pages that are then interlinked with each other.

**Final Project Glimpses**







**Conclusion**

The main aim of this project was to design a multi-page responsive webpage which can be used in all the devices of different screen ratios.

The approach used is as follows:

Used HTML to build the structure of the webpage, then used Bootstrap (CSS framework) for various components and some CSS wherever required, at the end used the <scripts> to include JavaScript.