

Major Project Report

On

“Online Industrial Training & Courses”

Submitted in partial fulfillment of the requirement for the degree of

.....

Session 20...-20...

SUBMITTED BY

NAME

Stream & Sem

Roll No.

SUBMITTED TO

University Name

University Logo

College name

(Approved by, Affiliated to University,)

CERTIFICATE

This is to certify that the project titled “**PROJECT NAME**” is the bonafide work carried out by **STUDENT NAME**, student of **STREAM** of **UNIVERSITY NAME** in fulfillment of the requirements for the award of the degree of **DEGREE NAME**.

The report submitted by the student is the work of student himself and he/she has fulfilled the requirements of completing his/her work during the training. His/her conduct was good during the entire training period. He/ She were sincere towards his work and did all the study with thorough understanding and dedication.

(-----)

Director

(Prof.-----)

Internal Guide

DECLARATION

I Beg **STUDENT NAME** hereby declare that the Project Report entitled
“**PROJECT NAME**” submitted by me to **UNIVESITY NAME** is
record of an original work done by me under the guidance of
TRAINER’S NAME.

This Project report has been submitted in partial fulfillment of the requirement
for the award of degree of **DEGREE NAME**. This is my original work and the
conclusions drawn therein are based on the material collected by myself. This
project report has not been submitted to any other university or institute for the
award of any degree or diploma.

Under the Guidance of:

Signature of the Student

PROF. NAME

STUDENT NAME

Uni.

Roll no.

Class: - stream &

sem

.....

(Project Supervisor)

ACKNOWLEDGEMENT

Heartfelt thanks to the following people...

A Few typewritten words of thanks can-not really express the sincerity of my gratitude. But I am trying to put into words my gratefulness towards all who have helped & encourage me in carrying out the project.

I would like to thank my trainer Mrs. ----- . Who helped me throughout the project .

I would then like to thank my faculty guide, Prof. -----, for all his valuable inputs and constant support towards me throughout my project and providing me an opportunity to learn outside the class room. It was a truly wonderful learning experience.

I would like to dedicate this project to my parents. Without their help and constant support this project would not have been possible. I would like to thank all my friends who did their valuable suggestions and support.

Last but not the least I would like to thank all the respondents who offered their opinions and suggestions and sometimes critical views throughout the survey which made me constantly update myself come out with a successful project.

Roll No:

Student Name

Stream & Semester

INDEX

<u>S.no</u>	<u>Topic</u>
1.	Introduction to Full Stack Development
2.	Introduction to Front End Development
3.	Introduction to Html
4.	Introduction to CSS
5.	Introduction to Java Script
6.	Introduction to Bootstrap
7.	METHODOLOGY
8.	Screenshots
9.	References

FULL STACK DEVELOPMENT

Full-stack Development refers to the full study of a computer system application, and full-stack developers straddle two distinct areas of web development: front-end and back-end.

The front end includes everything a customer, or web user, can see and interact with. In contrast, the back-end refers to all the servers, databases, and other internal architectures that drive the application; generally, the end user never interacts directly with this domain.

The easiest way to put the full stack into perspective is to imagine a restaurant. The front part includes the well-decorated and comfortable seating areas where visitors enjoy their food.

The kitchen and pantry make up the “back-end” and are usually hidden from view from the customer.

Front end developers work to optimize the visible parts of an application for web browsers and mobile devices. Front end platforms are usually built with HTML, CSS, and JavaScript.

Back end developers, in contrast, refine the software code that communicates with servers, databases, or other proprietary software that conveys information to front end interfaces.

The modern full stack developer is an experienced generalist who can build a minimal viable product — i.e., an application with enough functionality to please early customers and spark feedback for continued development — on their own. Companies rely on full stack professionals to spot errors between the front and back end and tackle tasks that straddle both disciplines. This versatility has become increasingly vital as newer apps begin to incorporate AI and other sophisticated technologies into their programming.

FRONT-END DEVELOPMENT

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant.

Everything you see on a website, like buttons, links, animations, and more, were created by a front end web developer. It is the front end developer's job to take the vision and design concept from the client and implement it through code. Everything on the page from the logo to the search bar, buttons, overall layout and how the user interacts with the page was created by a front end developer. Front end developers are in charge of the look and feel of the website.

Four Main Components of Front End Development-

- HTML
- CSS
- JavaScript
- BootStrap

Role of HTML in Web Designing

Hypertext Mark-up Language, or HTML, is a programming language used to describe the structure of information on a web page. The Hyper Text Markup Language or HTML is the standard mark-up language for documents designed to be displayed in a web browser. Together, HTML, CSS, and JavaScript make up the essential building blocks of websites, with CSS controlling a page's appearance, and JavaScript programming its functionality. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. You can think of HTML as providing the bones of a web page, while CSS provides the skin, and JavaScript provides the brains. A web page can contain headings, paragraphs, images, videos, and many other types of data. Front-end developers use HTML elements to specify what kind of information each item on a web page contains — for instance, the “p” element indicates a paragraph. Developers also write HTML code to specify how different items relate to one another in the overall structure of the page.

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 `` and `<input />` directly introduce content into the page.

Every website you open in your browser, from social networks to music services, uses HTML. A look under the hood of any website would reveal HTML code providing structure for all the page's components.

Different Versions of HTML

1. HTML 1.0

- The basic version of HTML has support for basic elements like text controls and images. This was the very basic version of HTML with less support for a wide range of HTML elements. It does not have rich features like styling and other things that were related to how content will be rendered in a browser.
- The initial version of HTML does not provide support for tables, font support, etc., as it provides us in the latest version.
- We would also like to discuss that W3C did not exist before HTML 2.0; hence it does not show details about HTML 1.

2. HTML 2

- HTML version 2.0 was developed in 1995 with basic intention of improving HTML version 1.0
- Now a standard got started to develop so as to maintain common rules and regulations across different browsers. HTML 2.0 has improved a lot in terms of the markup tags. In HTML 2.0 version concept of form came into force. Forms were developed, but still, they had basic tags like text boxes, buttons, etc.
- Also, the table came as an HTML tag. Now, in HTML tag 2.0, browsers also came with the concept of creating their own layers of tags that were specific to the browser itself. W3C was also formed. The main intention of W3C is to maintain standard across different web browsers so that these browsers understand and render HTML tags in a similar manner.

3. HTML 3.2

- It was developed in 1997. After HTML 2.0 was developed, the next version of HTML was 3.2
- With version 3.2 of HTML, HTML tags were further improved. It is worth noting that because of W3C standard maintenance, the newer version of HTML was 3.2 instead of 3.
- Now, HTML 3.2 has better support for new form elements. Another important feature what HTML 3.2 implemented was support for CSS. CSS stands for Cascading Style Sheet. It is CSS that provides features to make HTML tags look better on rendering it on browsers. CSS helps to style HTML elements.
- With the upgradation of browsers to HTML 3.2, the browser also supported for frame tags, although HTML specifications still do not support frame markup tags.

4. HTML 4.01

- It was developed in 1999. It extended the support of cascading styling sheets. In version 3.2, CSS were embedded in HTML page itself. Therefore, if the website has various web pages to apply to the style of each page, we must place CSS on each web page. Hence there was a repetition of the same block of CSS.
- To overcome this thing, in version 4.01 concept of an external styling sheet emerged. Under this concept, an external CSS file could be developed, and this external styling file could be included in HTML itself. HTML 4.01 provided support for further new tags of HTML.

5. HTML5

- This is the latest version of HTML. For a developer, it could be used in 2014. It came up with lots of HTML tags support. HTML5 provided support for new form elements like input elements of different types; geolocations support tags, etc.

Let us look at a few of the tags which were added to HTML5

- **Email** – New HTML5 tag, which was added, is the input element of type email. This is a form tag, although it could be used outside of a form tag also. This tag checks the validation of the input value. It checks whether the value inserted is a valid email.
- **Password** – This is another form tag that was added to receive a password from the user. Being the password type field, the user types in the field are not visible directly to the user but are represented by special symbols. These symbols save the password from getting revealed on the browser.
- **Audio tag** – This is a new audio tag that was implemented in HTML5. This tag helps to add audio to our web page. We can use this tag to embed an audio clip into a web page. This audio tag could be played on a webpage.
- **Semantic tags** – Semantic tags are also known as structural tags. Structural tags are the tags that provide structure to the HTML page. It helps it divide the HTML page into different structures. These structures get combined into an HTML page itself to form an HTML web page. Few of the important HTML semantic tags are fig caption, <header>, <footer>
- **Section tag** – This tag is used to semantic a section in an HTML page. A section tag represents a section on a web page.

6. W3C HTML Validator

An HTML validator is a web-based tool that is used to maintain or check whether a piece of HTML tag or HTML is valid. An HTML validator follows the standard of W3C to validate an HTML page. It follows the W3C standard.

Role of CSS in Web Designing

Cascading Style Sheets, commonly known as CSS, is an integral part of the modern web development process. It is a highly effective HTML tool that provides easy control over layout and presentation of website pages by separating content from design.

The standout advantage of CSS is the added design flexibility and interactivity it brings to web development. Developers have greater control over the layout allowing them to make precise section-wise changes.

As customization through CSS is much easier than plain HTML, web developers are able to create different looks for each page. Complex websites with uniquely presented pages are feasible thanks to CSS.

CSS works by creating rules. These rules are simultaneously applied to multiple elements within the site. Eliminating the repetitive coding style of HTML makes development work faster and less monotonous. Errors are also reduced considerably. Since the content is completely separated from the design, changes across the website can be implemented all at once. This reduces delivery times and costs of future edits.

Improved website loading is an underrated yet important benefit of CSS.

Browsers download the CSS rules once and cache them for loading all the pages of a website. It makes browsing the website faster and enhances the overall user experience.

TYPES OF CSS

Internal CSS

Internal or embedded CSS requires you to add **<style>** tag in the **<head>** section of your HTML document.

This CSS style is an effective method of styling a single page. However, using this style for multiple pages is time-consuming as you need to put CSS rules on every page of your website.

Inline CSS

Inline CSS is used to style a specific HTML element. For this CSS style, you'll only need to add the **style** attribute to each HTML tag, without using selectors.

This CSS type is not really recommended, as each HTML tag needs to be styled individually. Managing your website may become too hard if you only use inline CSS.

However, inline CSS in HTML can be useful in some situations. For example, in cases where you don't have access to CSS files or need to apply styles for a single element only.

Let's take a look at an example. Here, we add an inline CSS to the **<p>** and **<h1>** tag:

External CSS

With external CSS, you'll link your web pages to an external **.css** file, which can be created by any text editor in your device (e.g., **Notepad++**).

This CSS type is a more efficient method, especially for styling a large website. By editing one **.css** file, you can change your entire site at once

Role of JavaScript in Web Designing

In November 1996, Netscape submitted JavaScript to Ecma International, as the starting point for a standard specification that all browser vendors could conform to. This led to the official release of the first ECMAScript language specification in June 1997.

The standards process continued for a few years, with the release of ECMAScript 2 in June 1998 and ECMAScript 3 in December 1999. Work on ECMAScript 4 began in 2000.

Meanwhile, Microsoft gained an increasingly dominant position in the browser market. By the early 2000s, Internet Explorer's market share reached 95%. This meant that JScript became the de facto standard for client-side scripting on the Web.

JavaScript is the dominant client-side scripting language of the Web, with 97% of websites using it for this purpose. Scripts are embedded in or included from HTML documents and interact with the DOM. All major web browsers have a built-in JavaScript engine that executes the code on the user's device.

Related Technologies

Java

A common misconception is that JavaScript is the same as Java. Both indeed have a C-like syntax (the C language being their most immediate common ancestor language). They are also typically sandboxed , and JavaScript was designed with Java's syntax and standard library in mind.

In particular, all Java keywords were reserved in original JavaScript, JavaScript's standard library follows Java's naming conventions, and JavaScript's `Math` and `Date` objects are based on classes from Java 1.0.

The differences between the two languages are more prominent than their similarities. Java has static typing, while JavaScript's typing is dynamic. Java is loaded from compiled bytecode, while JavaScript is loaded as human-readable source code. Java's objects are class-based, while JavaScript's are prototype-based. Finally, Java did not support functional programming until Java 8, while JavaScript has done so from the beginning, being influenced by Scheme.

JSON

JSON, or JavaScript Object Notation, is a general-purpose data interchange format that is defined as a subset of JavaScript's object literal syntax.

WebAssembly

Since 2017, web browsers have supported WebAssembly, a binary format that enables a JavaScript engine to execute performance-critical portions of web page scripts close to native speed. WebAssembly code runs in the same sandbox as regular JavaScript code.

asm.js is a subset of JavaScript that served as the forerunner of WebAssembly.

Transpilers

JavaScript is the dominant client-side language of the Web, and many websites are script-heavy. Thus transpilers have been created to convert code written in other languages, which can aid the development process.

Role of BootStrap in Web Development

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

As of August 2021, Bootstrap is the tenth most starred project on GitHub

Bootstrap allows back-end developers the ability to pseudo-implement layouts without the need for a front-end developer. It gives people who don't know much about how CSS works an easy way to implement basic grid structures for modules and full-page layouts. It is a great tool for rapid prototype development.

Methodology

The goal of front-end development is to ensure the best possible user experience which includes efficiency, speed, and smooth functionality. In fact, a front-end developer acts as a mediator between a designer and a back-end developer.

Moreover, the tools and methodologies used for front-end development are continuously evolving and the developers need to constantly upgrade their skills.

Front End Development Steps

Step 1. Requirement Gathering

As in other spheres of software development, the front-end development process begins with collecting and defining business requirements. Gathering requirements for the project is the most important part. During this stage, the product owner describes the expectations of the project:

- what the goal of the project is;
- who the target audience is;
- how they will use the product.

It's important for the development team to understand the needs of the client because this information is critical for addressing the product owner's requests.

Step 2. Creating Prototype

Once the requirements are collected, the developers create a prototype that is presented to the client for evaluation. A prototype is an early version of the future application. It demonstrates a basic idea of what the app will look like and how it will work.

After receiving client feedback, the development team modifies the prototype and presents it to the client again for evaluation. Once the client approves the prototype, it is used as a requirement for building the actual application.

Step 3. Development

At the development stage, all the requirements from the previous phases are transformed into the actual system. Developers define the most relevant front-end frameworks, tools, and the best development practices to ensure that the product is developed in the most efficient way. They purchase and install the respective software and hardware and create the actual code on the basis of given specifications. At this stage the front-developer:

- develops user-facing features;
- builds reusable code for future use;
- ensures the technical feasibility of UI/UX design;
- optimizes application for maximum speed and scalability;
- assures that all user input is validated before submitting to the back-end;
- collaborates with other team members and the client.

Step 4. QA and Testing

Testing is normally carried out by Quality Assurance specialists. Once the development process is complete, the app is deployed in the testing environment and the QA specialists start testing the functionality of the app. Testing is performed to ensure that the app works according to the specified requirements.

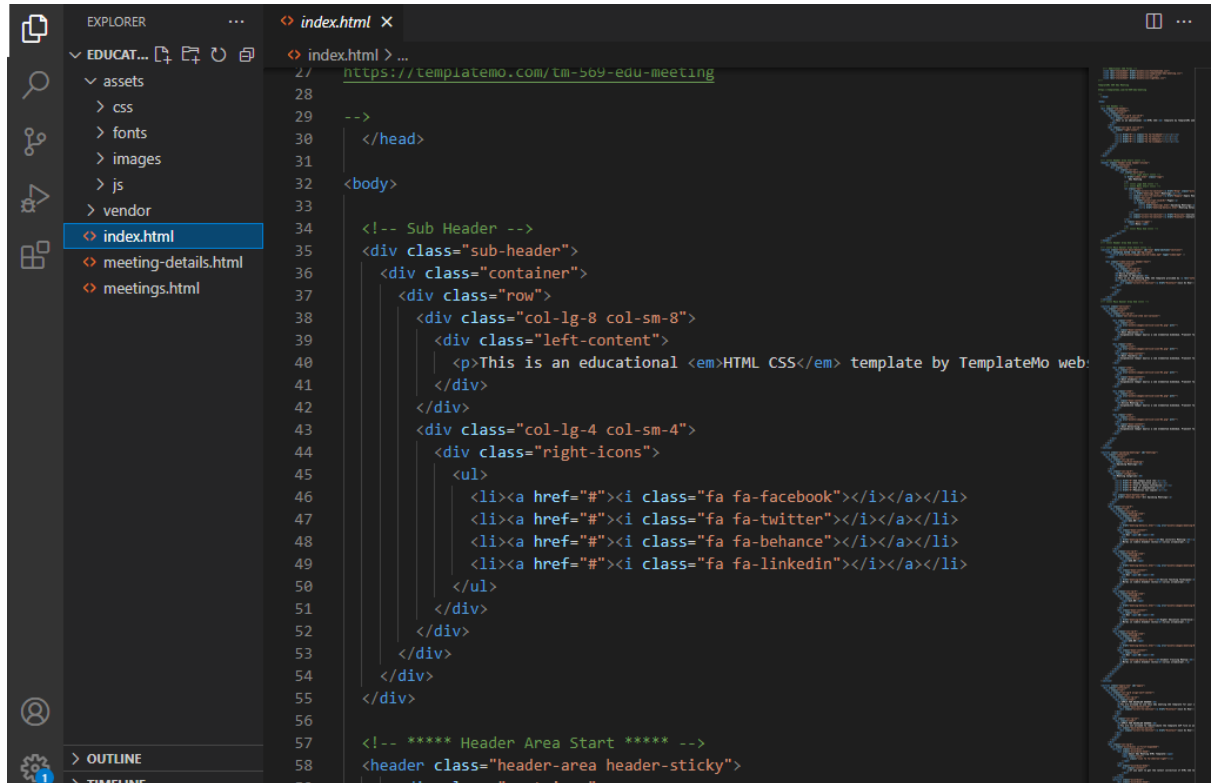
You may wonder how this step is related to front-end development. The thing is that once the QA team finds some bugs they report on the problems to developers. The developers fix the bug and send the app back to the QA team for retesting. This procedure continues until the app is stable and works flawlessly.

Step 5. Maintenance and Support

The development doesn't end with the app rollout. Bugs may appear at any time so the development team should constantly monitor the app performance to ensure it functions properly. The main challenge here is to ensure that code enhancement doesn't cause any other malfunctions.

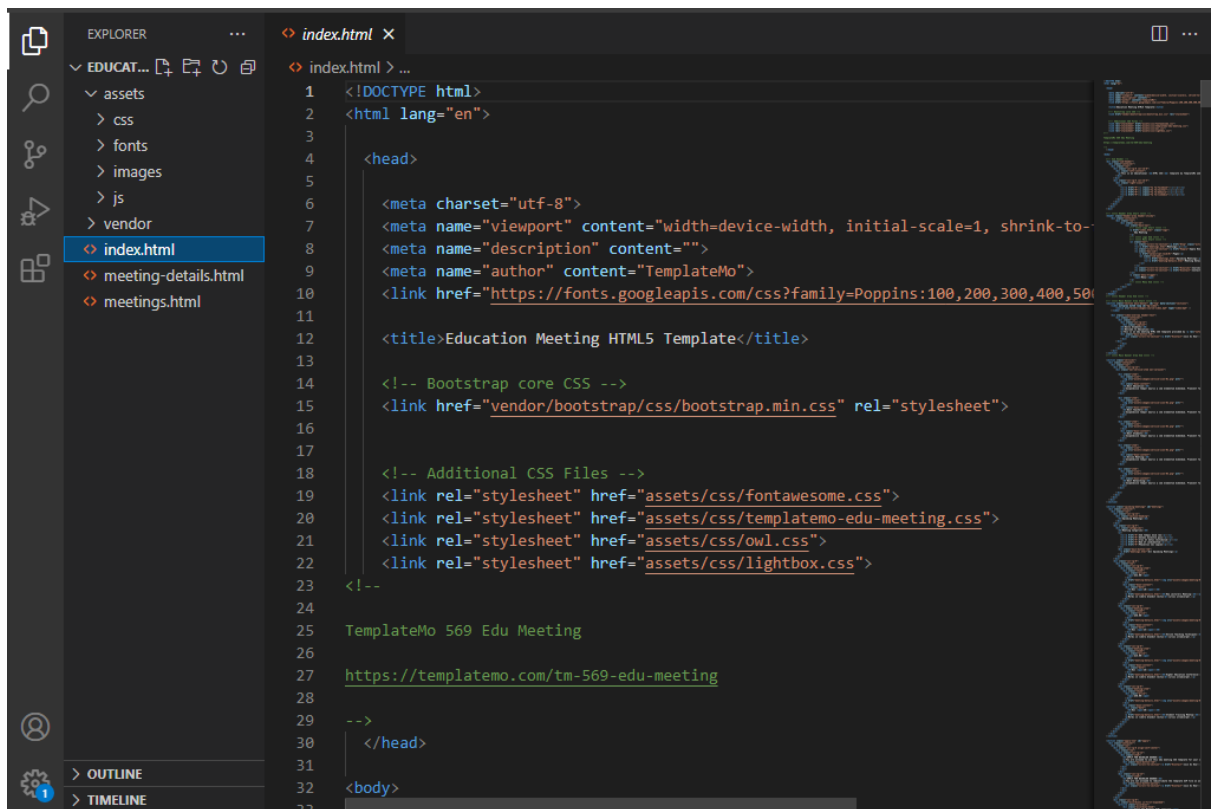
Coding Screenshots

1.



```
index.html X
index.html > ...
27 https://templatemo.com/tm-569-edu-meeting
28
29 -->
30 </head>
31
32 <body>
33
34 <!-- Sub Header -->
35 <div class="sub-header">
36   <div class="container">
37     <div class="row">
38       <div class="col-lg-8 col-sm-8">
39         <div class="left-content">
40           <p>This is an educational <em>HTML CSS</em> template by TemplateMo web
41         </div>
42       </div>
43       <div class="col-lg-4 col-sm-4">
44         <div class="right-icons">
45           <ul>
46             <li><a href="#"><i class="fa fa-facebook"></i></a></li>
47             <li><a href="#"><i class="fa fa-twitter"></i></a></li>
48             <li><a href="#"><i class="fa fa-behance"></i></a></li>
49             <li><a href="#"><i class="fa fa-linkedin"></i></a></li>
50           </ul>
51         </div>
52       </div>
53     </div>
54   </div>
55 </div>
56
57 <!-- ***** Header Area Start ***** -->
58 <header class="header-area header-sticky">
59   <div class="container">
```

2



```
index.html X
index.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5
6   <meta charset="utf-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
8   <meta name="description" content="">
9   <meta name="author" content="TemplateMo">
10  <link href="https://fonts.googleapis.com/css?family=Poppins:100,200,300,400,500
11
12  <title>Education Meeting HTML5 Template</title>
13
14  <!-- Bootstrap core CSS -->
15  <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
16
17  <!-- Additional CSS Files -->
18  <link rel="stylesheet" href="assets/css/fontawesome.css">
19  <link rel="stylesheet" href="assets/css/templatemo-edu-meeting.css">
20  <link rel="stylesheet" href="assets/css/owl.css">
21  <link rel="stylesheet" href="assets/css/lightbox.css">
22
23  <!--
24  TemplateMo 569 Edu Meeting
25
26  https://templatemo.com/tm-569-edu-meeting
27
28  -->
29  </head>
30
31  <body>
```

3.

```

56 <!-- ***** Header Area Start ***** -->
57 <header class="header-area header-sticky">
58   <div class="container">
59     <div class="row">
60       <div class="col-12">
61         <nav class="main-nav">
62           <!-- ***** Logo Start ***** -->
63           <a href="index.html" class="logo">
64             Edu Meeting
65           </a>
66           <!-- ***** Logo End ***** -->
67           <!-- ***** Menu Start ***** -->
68           <ul class="nav">
69             <li class="scroll-to-section"><a href="#top" class="active">Home</a></li>
70             <li><a href="meetings.html">Meetings</a></li>
71             <li class="scroll-to-section"><a href="#apply">Apply Now</a></li>
72             <li class="has-sub">
73               <a href="javascript:void(0)">Pages</a>
74               <ul class="sub-menu">
75                 <li><a href="meetings.html">Upcoming Meetings</a></li>
76                 <li><a href="meeting-details.html">Meeting Details</a></li>
77               </ul>
78             </li>
79             <li class="scroll-to-section"><a href="#courses">Courses</a></li>
80             <li class="scroll-to-section"><a href="#contact">Contact Us</a></li>
81           </ul>
82           <a class="menu-trigger">
83             <span>Menu</span>
84           </a>
85           <!-- ***** Menu End ***** -->
86         </nav>
87       </div>

```

4.

```

90 </div>
91 </header>
92 <!-- ***** Header Area End ***** -->
93
94 <!-- ***** Main Banner Area Start ***** -->
95 <section class="section main-banner" id="top" data-section="section1">
96   <video autoplay muted loop id="bg-video">
97     <source src="assets/images/course-video.mp4" type="video/mp4" />
98   </video>
99
100   <div class="video-overlay header-text">
101     <div class="container">
102       <div class="row">
103         <div class="col-lg-12">
104           <div class="caption">
105             <h1>Hello Students</h1>
106             <h2>Welcome to Education</h2>
107             <p>This is an edu meeting HTML CSS template provided by <a href="#">NoFoxy</a></p>
108             <div class="main-button-red">
109               <div class="scroll-to-section"><a href="#contact">Join Us Now!</a></div>
110             </div>
111           </div>
112         </div>
113       </div>
114     </div>
115   </div>
116 </section>
117 <!-- ***** Main Banner Area End ***** -->
118
119 <section class="services">
120   <div class="container">
121     <div class="row">
122       <div class="col-lg-12">

```

5.

```

141 </div>
142 </div>
143 </div>
144
145 <div class="item">
146   <div class="icon">
147     
148   </div>
149   <div class="down-content">
150     <h4>Best Students</h4>
151     <p>Suspendisse tempor mauris a sem elementum bibendum. Praesent fa
152   </div>
153 </div>
154
155 <div class="item">
156   <div class="icon">
157     
158   </div>
159   <div class="down-content">
160     <h4>Online Meeting</h4>
161     <p>Suspendisse tempor mauris a sem elementum bibendum. Praesent fa
162   </div>
163 </div>
164
165 <div class="item">
166   <div class="icon">
167     
168   </div>
169   <div class="down-content">
170     <h4>Best Networking</h4>
171     <p>Suspendisse tempor mauris a sem elementum bibendum. Praesent fa
172   </div>

```

6.

```

185 <div class="section-heading">
186   <h2>Upcoming Meetings</h2>
187 </div>
188 </div>
189 <div class="col-lg-4">
190   <div class="categories">
191     <h4>Meeting Categories</h4>
192     <ul>
193       <li><a href="#">Sed tempus enim leo</a></li>
194       <li><a href="#">Aenean molestie quis</a></li>
195       <li><a href="#">Cras et metus vestibulum</a></li>
196       <li><a href="#">Nam et condimentum</a></li>
197       <li><a href="#">Phasellus nec sapien</a></li>
198     </ul>
199   </div>
200   <div class="main-button-red">
201     <a href="meetings.html">All Upcoming Meetings</a>
202   </div>
203 </div>
204 <div class="col-lg-8">
205   <div class="row">
206     <div class="col-lg-6">
207       <div class="meeting-item">
208         <div class="thumb">
209           <div class="price">
210             <span>$22.00</span>
211           </div>
212           <a href="meeting-details.html">
215           <div class="date">
216             <h6>Nov <span>10</span></h6>
217           </div>

```

7.

```

212     <a href="meeting-details.html">
215         <div class="date">
216             <h6>Nov <span>10</span></h6>
217         </div>
218         <a href="meeting-details.html"><h4>New Lecturers Meeting</h4></a>
219         <p>Morbi in libero blandit lectus<br>cursus ullamcorper.</p>
220     </div>
221 </div>
222 </div>
223 <div class="col-lg-6">
224     <div class="meeting-item">
225         <div class="thumb">
226             <div class="price">
227                 <span>$36.00</span>
228             </div>
229             <a href="meeting-details.html">
232             <div class="date">
233                 <h6>Nov <span>24</span></h6>
234             </div>
235             <a href="meeting-details.html"><h4>Online Teaching Techniques</h4>
236             <p>Morbi in libero blandit lectus<br>cursus ullamcorper.</p>
237         </div>
238     </div>
239 </div>
240 <div class="col-lg-6">
241     <div class="meeting-item">
242         <div class="thumb">
243             <div class="price">

```

8.

```

275     </div>
276 </div>
277 </div>
278 </section>
279
280 <section class="apply-now" id="apply">
281     <div class="container">
282         <div class="row">
283             <div class="col-lg-6 align-self-center">
284                 <div class="row">
285                     <div class="col-lg-12">
286                         <div class="item">
287                             <h3>APPLY FOR BACHELOR DEGREE</h3>
288                             <p>You are allowed to use this edu meeting CSS template for your s
289                             <div class="main-button-red">
290                                 <div class="scroll-to-section"><a href="#contact">Join Us Now!</
291                             </div>
292                         </div>
293                     </div>
294                     <div class="col-lg-12">
295                         <div class="item">
296                             <h3>APPLY FOR BACHELOR DEGREE</h3>
297                             <p>You are not allowed to redistribute the template ZIP file on an
298                             <div class="main-button-yellow">
299                                 <div class="scroll-to-section"><a href="#contact">Join Us Now!</
300                             </div>
301                         </div>
302                     </div>
303                 </div>
304             </div>
305             <div class="col-lg-6">
306                 <div class="accordions is-first-expanded">

```

9.

The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with folders: assets, css, fonts, images, js, and vendor. The file explorer also shows files: index.html (selected), meeting-details.html, and meetings.html. The main editor area shows the index.html file with the following HTML code:

```

312     </span>
313   </div>
314   <div class="accordion-body">
315     <div class="content">
316       <p>If you want to get the latest collection of HTML CSS te
317     </div>
318   </div>
319 </article>
320 <article class="accordion">
321   <div class="accordion-head">
322     <span>HTML CSS Bootstrap Layout</span>
323     <span class="icon">
324       <i class="icon fa fa-chevron-right"></i>
325     </span>
326   </div>
327   <div class="accordion-body">
328     <div class="content">
329       <p>Etiam posuere metus orci, vel consectetur elit imperdie
330       Nulla malesuada elit lacus, ac ultricies massa varius sed.
331     </div>
332   </div>
333 </article>
334 <article class="accordion">
335   <div class="accordion-head">
336     <span>Please tell your friends</span>
337     <span class="icon">
338       <i class="icon fa fa-chevron-right"></i>
339     </span>
340   </div>
341   <div class="accordion-body">
342     <div class="content">
343       <p>Ut vehicula mauris est, sed sodales justo rhoncus eu. M

```

10.

The screenshot shows the VS Code editor with the file explorer on the left. The file explorer shows a project structure with folders: assets, css, fonts, images, js, and vendor. The file explorer also shows files: index.html (selected), meeting-details.html, and meetings.html. The main editor area shows the index.html file with the following HTML code:

```

367 <section class="our-courses" id="courses">
368   <div class="container">
369     <div class="row">
370       <div class="col-lg-12">
371         <div class="section-heading">
372           <h2>Our Popular Courses</h2>
373         </div>
374       </div>
375     </div>
376     <div class="col-lg-12">
377       <div class="owl-courses-item owl-carousel">
378         <div class="item">
379           
380           <div class="down-content">
381             <h4>Morbi tincidunt elit vitae justo rhoncus</h4>
382             <div class="info">
383               <div class="row">
384                 <div class="col-8">
385                   <ul>
386                     <li><i class="fa fa-star"></i></li>
387                     <li><i class="fa fa-star"></i></li>
388                     <li><i class="fa fa-star"></i></li>
389                     <li><i class="fa fa-star"></i></li>
390                     <li><i class="fa fa-star"></i></li>
391                   </ul>
392                 </div>
393                 <div class="col-4">
394                   <span>$160</span>
395                 </div>
396               </div>
397             </div>
398           </div>
399         </div>

```

11.

12.
24

13.

```

693     </div>
694 </section>
695
696 <section class="contact-us" id="contact">
697   <div class="container">
698     <div class="row">
699       <div class="col-lg-9 align-self-center">
700         <div class="row">
701           <div class="col-lg-12">
702             <form id="contact" action="" method="post">
703               <div class="row">
704                 <div class="col-lg-12">
705                   <h2>Let's get in touch</h2>
706                 </div>
707                 <div class="col-lg-4">
708                   <fieldset>
709                     <input name="name" type="text" id="name" placeholder="YOURNAME">
710                   </fieldset>
711                 </div>
712                 <div class="col-lg-4">
713                   <fieldset>
714                     <input name="email" type="text" id="email" pattern="^[^ @]*@[^ @]*\.[a-z]{2,4}$">
715                   </fieldset>
716                 </div>
717                 <div class="col-lg-4">
718                   <fieldset>
719                     <input name="subject" type="text" id="subject" placeholder="SUBJECT">
720                   </fieldset>
721                 </div>
722                 <div class="col-lg-12">
723                   <fieldset>
724                     <textarea name="message" type="text" class="form-control" id="message">

```

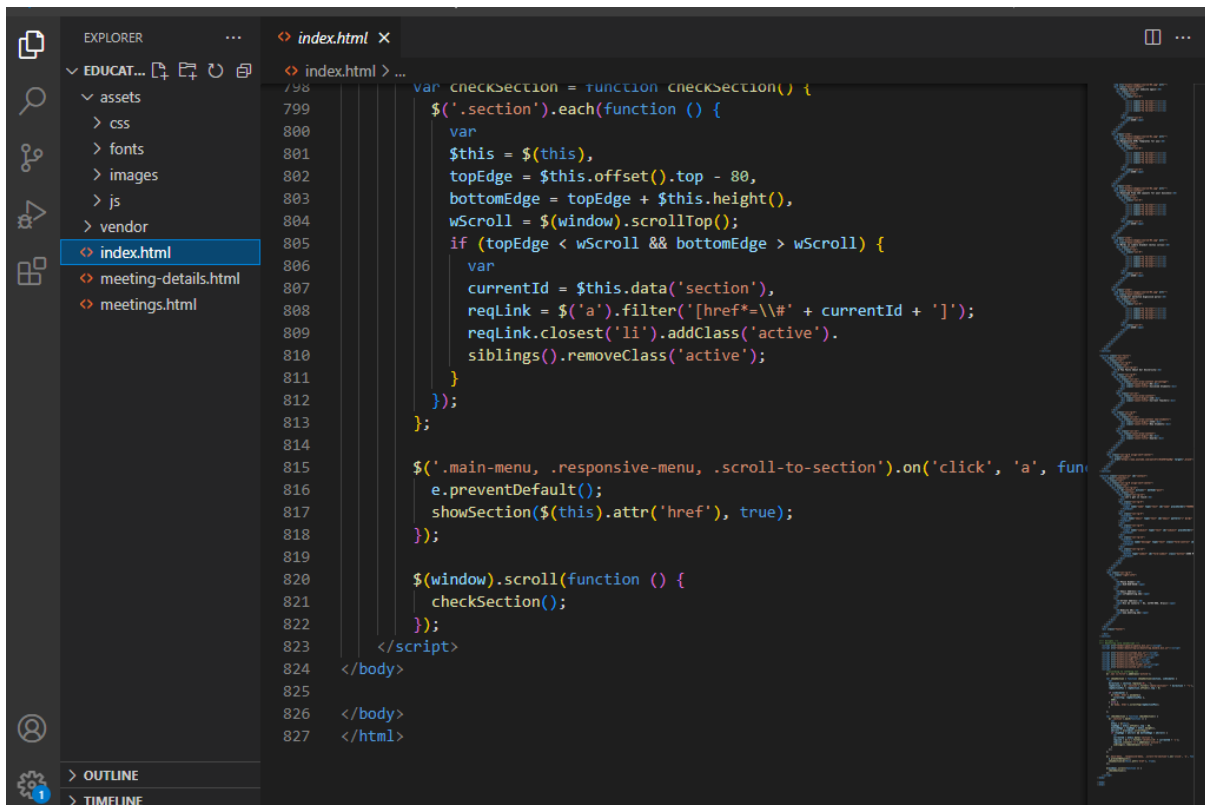
14.

```

762 </div>
763 </section>
764
765 <!-- Scripts -->
766 <!-- Bootstrap core JavaScript -->
767 <script src="vendor/jquery/jquery.min.js"></script>
768 <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
769
770 <script src="assets/js/isotope.min.js"></script>
771 <script src="assets/js/owl-carousel.js"></script>
772 <script src="assets/js/lightbox.js"></script>
773 <script src="assets/js/tabs.js"></script>
774 <script src="assets/js/video.js"></script>
775 <script src="assets/js/slick-slider.js"></script>
776 <script src="assets/js/custom.js"></script>
777
778 //according to loftblog tut
779 $('.nav li:first').addClass('active');
780
781 var showSection = function showSection(section, isAnimate) {
782   var
783     direction = section.replace(/#/, ''),
784     reqSection = $('.section').filter('[data-section="' + direction + '"]'),
785     reqSectionPos = reqSection.offset().top - 0;
786
787   if (isAnimate) {
788     $('body, html').animate({
789       scrollTop: reqSectionPos },
790       800);
791   } else {
792     $('body, html').scrollTop(reqSectionPos);
793   }

```

15.



The screenshot shows a code editor with the following structure:

- EXPLORER** (Left Panel):
 - EDUCAT...
 - assets
 - css
 - fonts
 - images
 - js
 - vendor
 - index.html (Selected)
 - meeting-details.html
 - meetings.html
- index.html** (Main Editor):

```
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827
```

```
var checkSection = function checkSection() {  
    $('.section').each(function () {  
        var  
        $this = $(this),  
        topEdge = $this.offset().top - 80,  
        bottomEdge = topEdge + $this.height(),  
        wScroll = $(window).scrollTop();  
        if (topEdge < wScroll && bottomEdge > wScroll) {  
            var  
            currentId = $this.data('section'),  
            reqLink = $('a').filter('[href*="#'+ currentId + ']');  
            reqLink.closest('li').addClass('active').  
            siblings().removeClass('active');  
        }  
    });  
};  
  
$('.main-menu, .responsive-menu, .scroll-to-section').on('click', 'a', function (e) {  
    e.preventDefault();  
    showSection($(this).attr('href'), true);  
});  
  
$(window).scroll(function () {  
    checkSection();  
});
```

```
</script>  
</body>  
</body>  
</html>
```
- OUTLINE** (Bottom Left):
 - 1
- TIMELINE** (Bottom Left):

Project Screenshots

Figure: 1

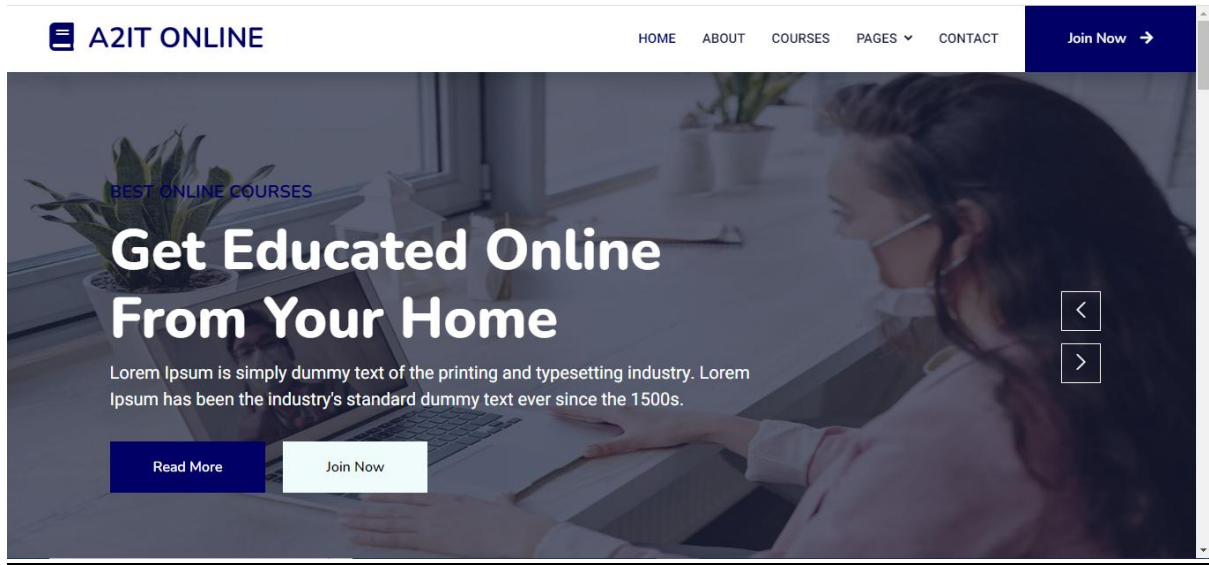


Figure:2

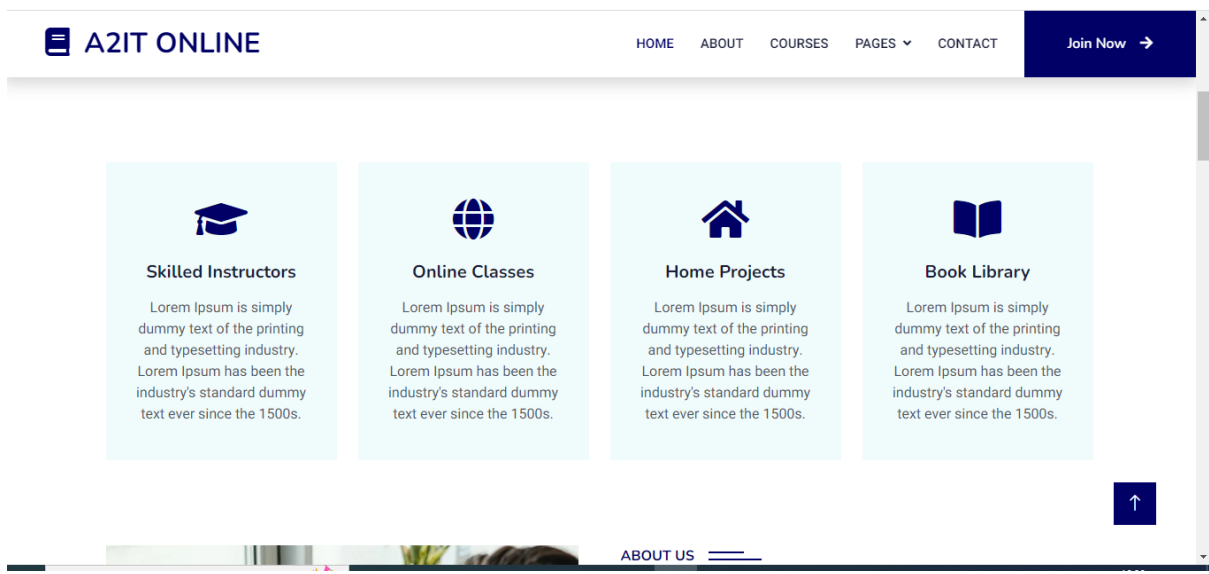


Figure: 3

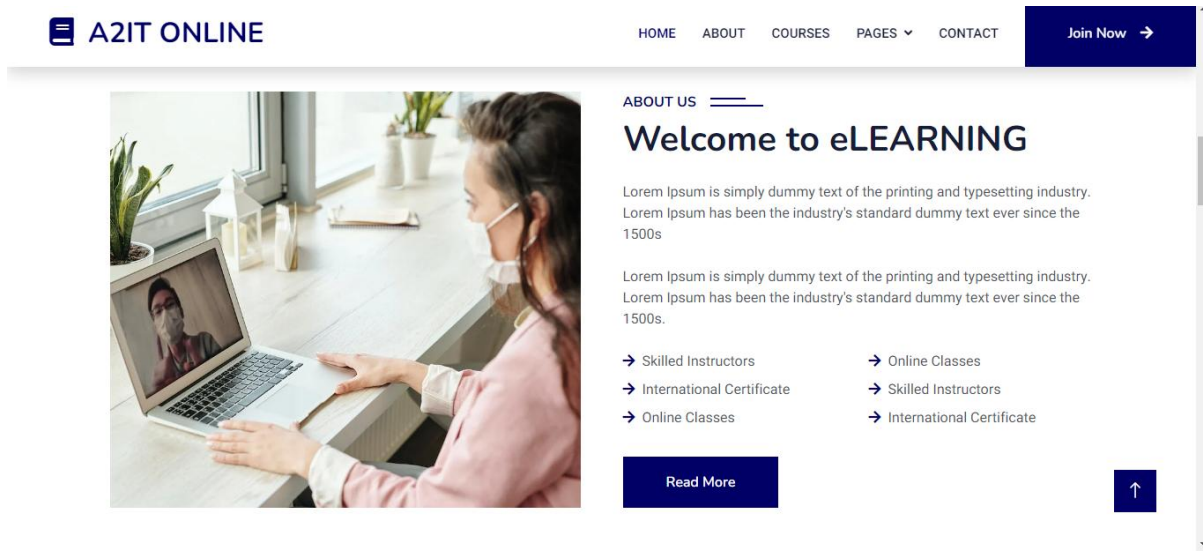


Figure: 4

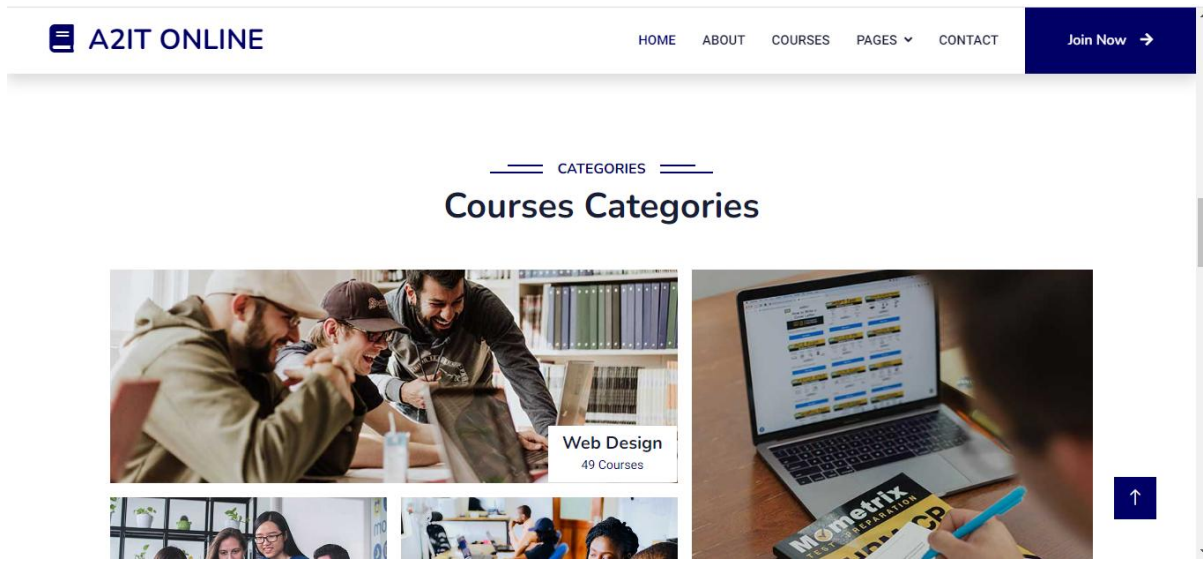


Figure:5

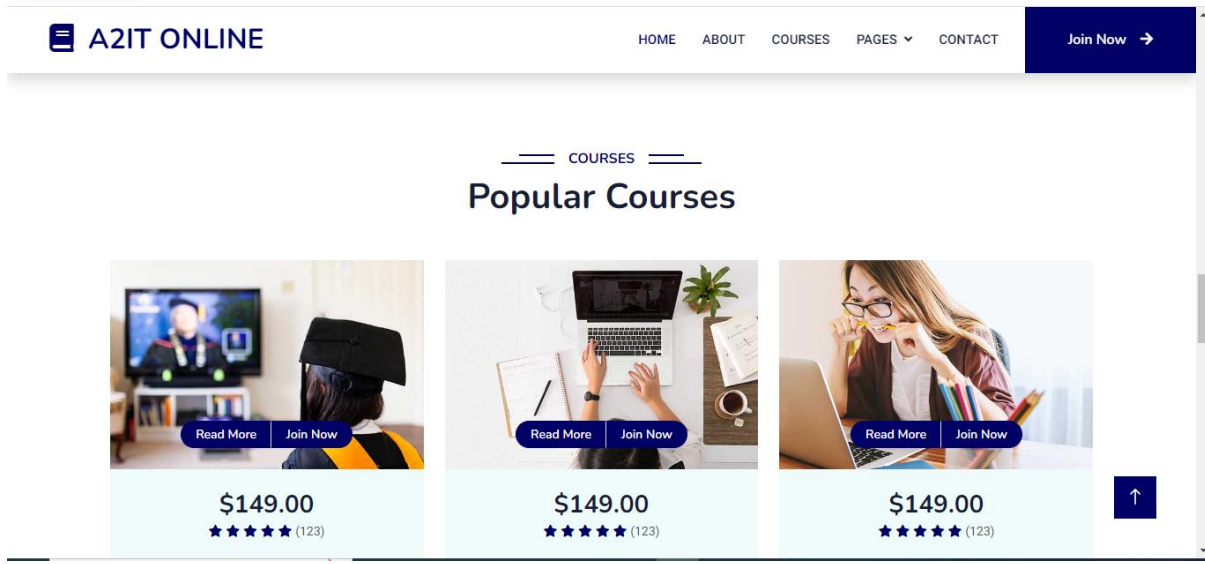


Figure:6

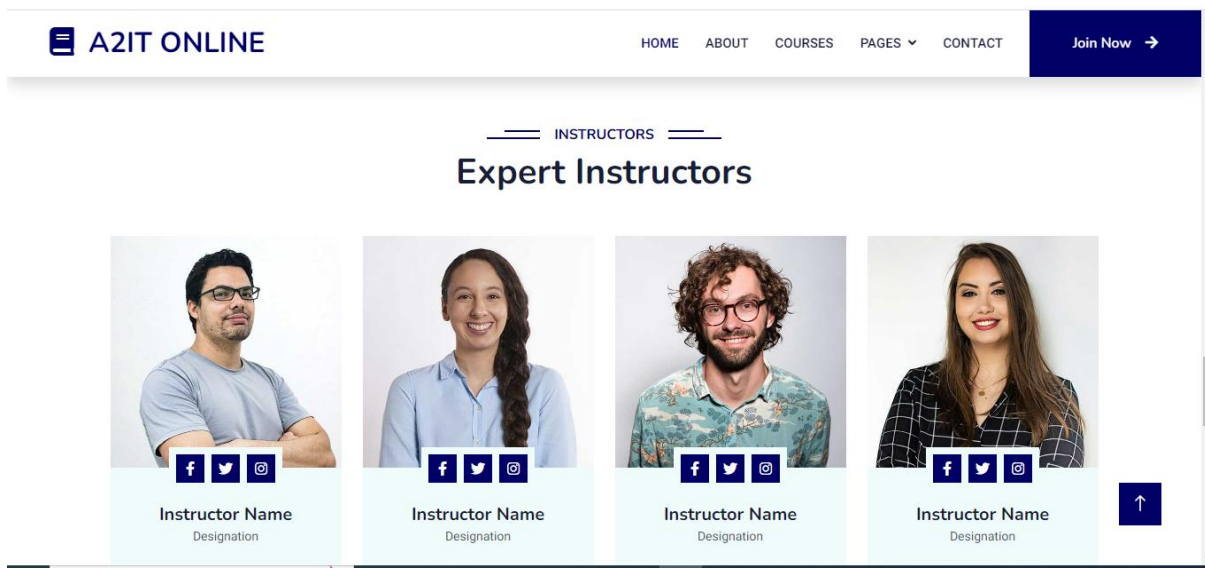


Figure: 7

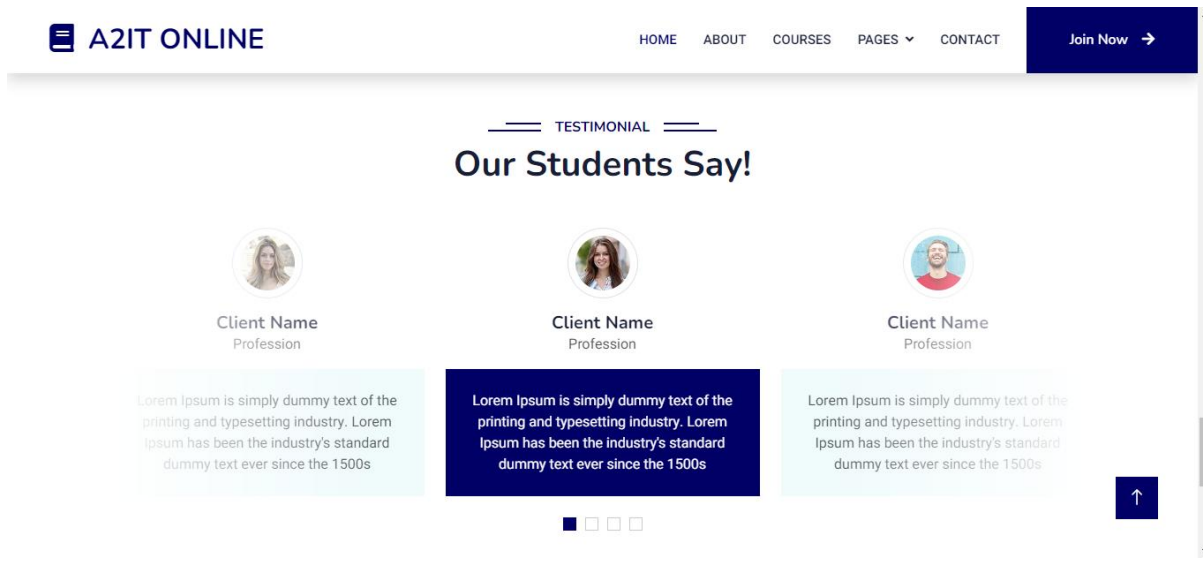


Figure: 8

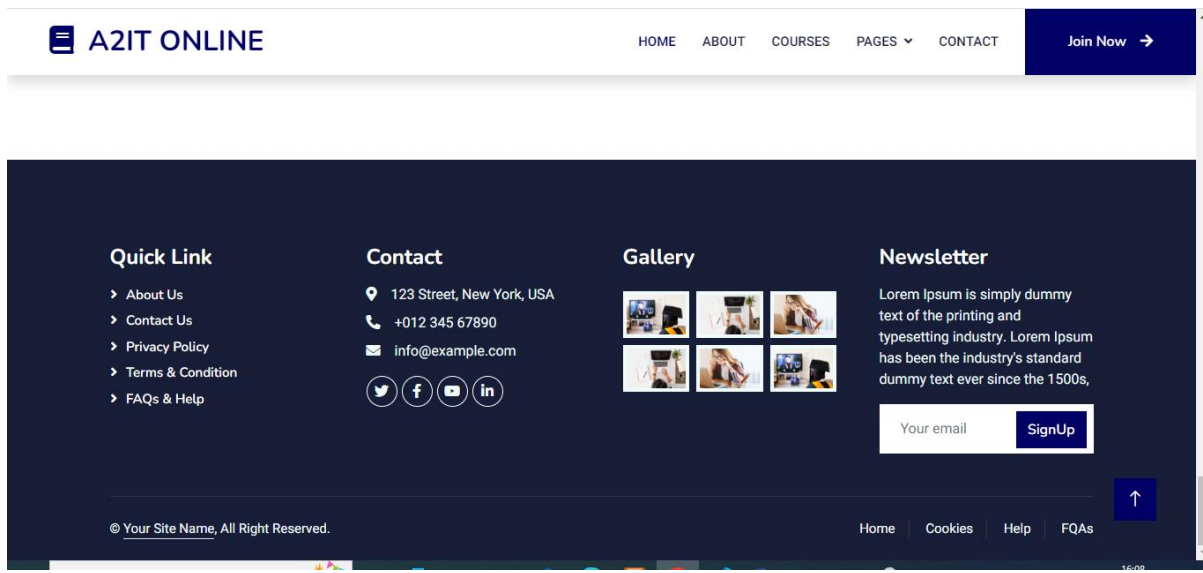


Figure: 9

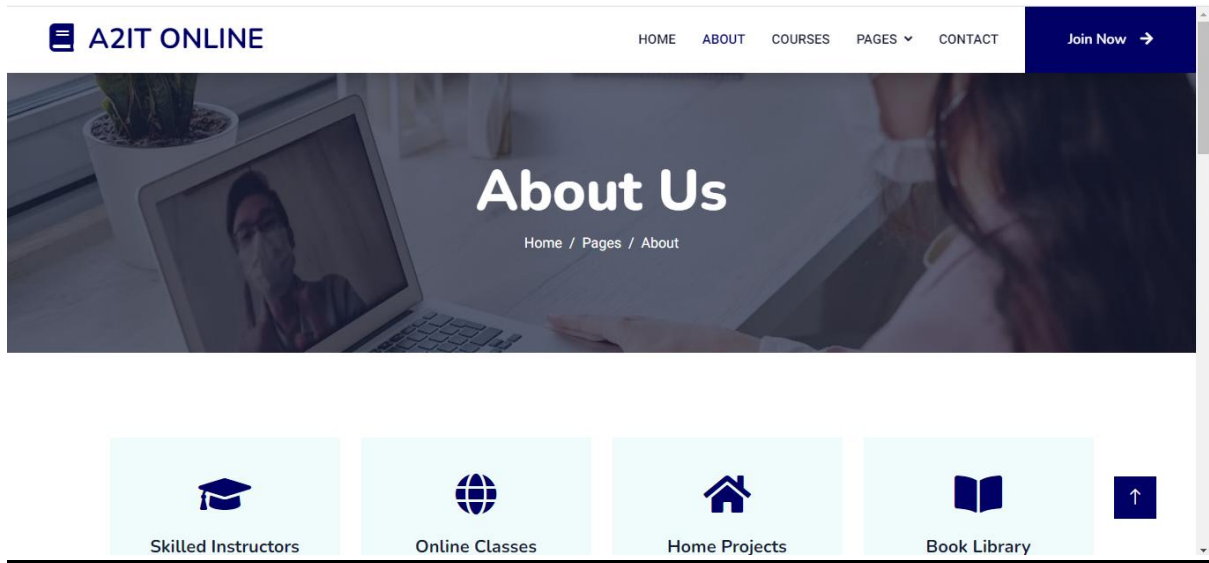


Figure: 10

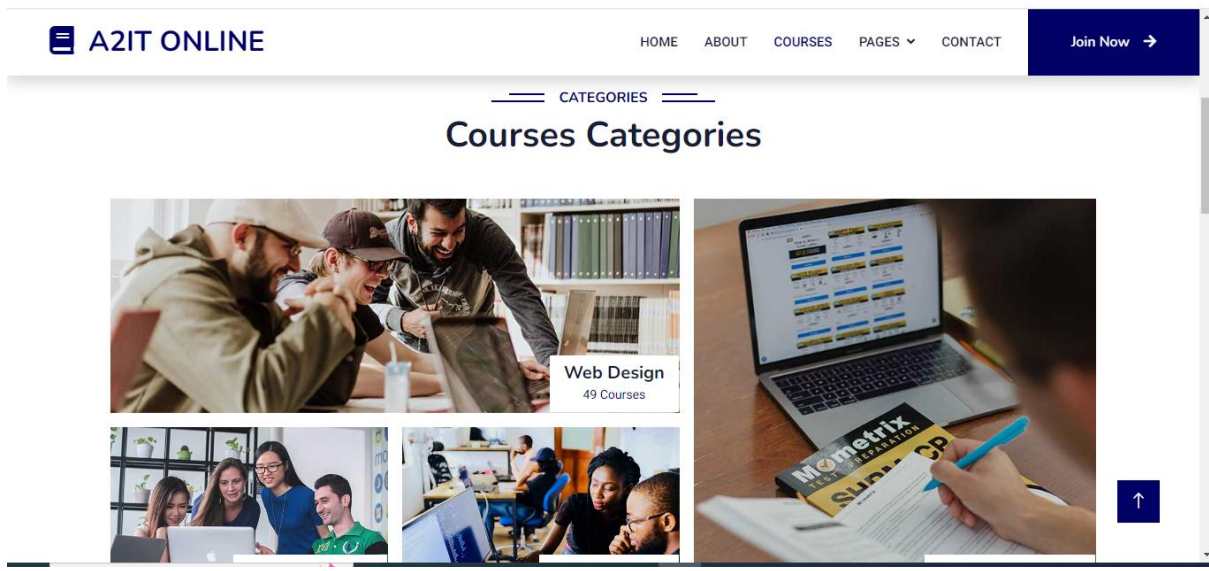


Figure:11

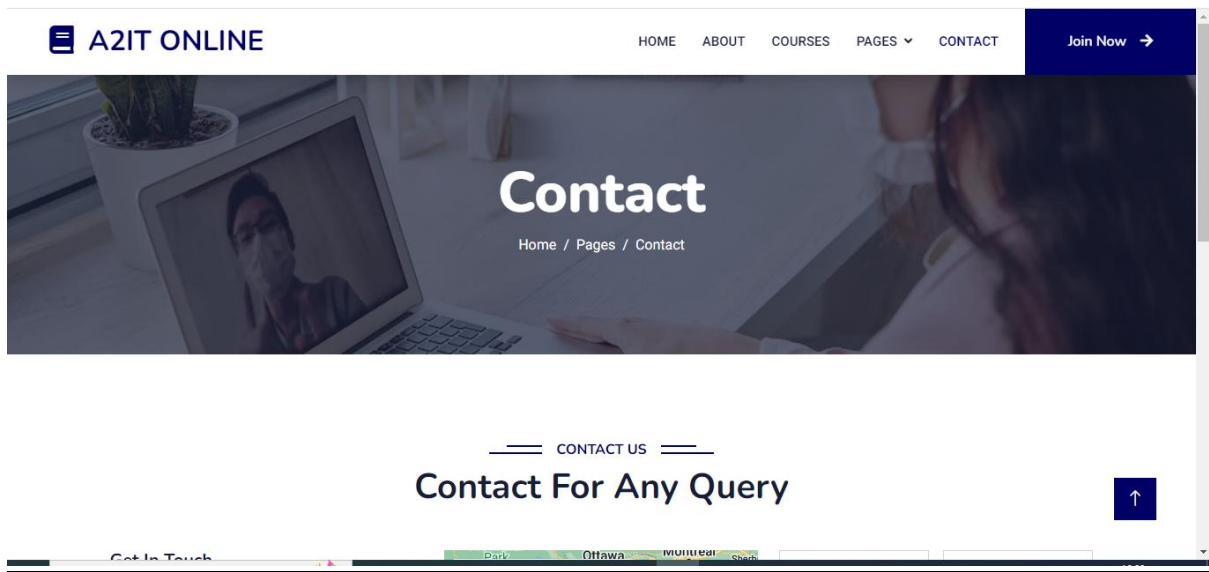


Figure:12

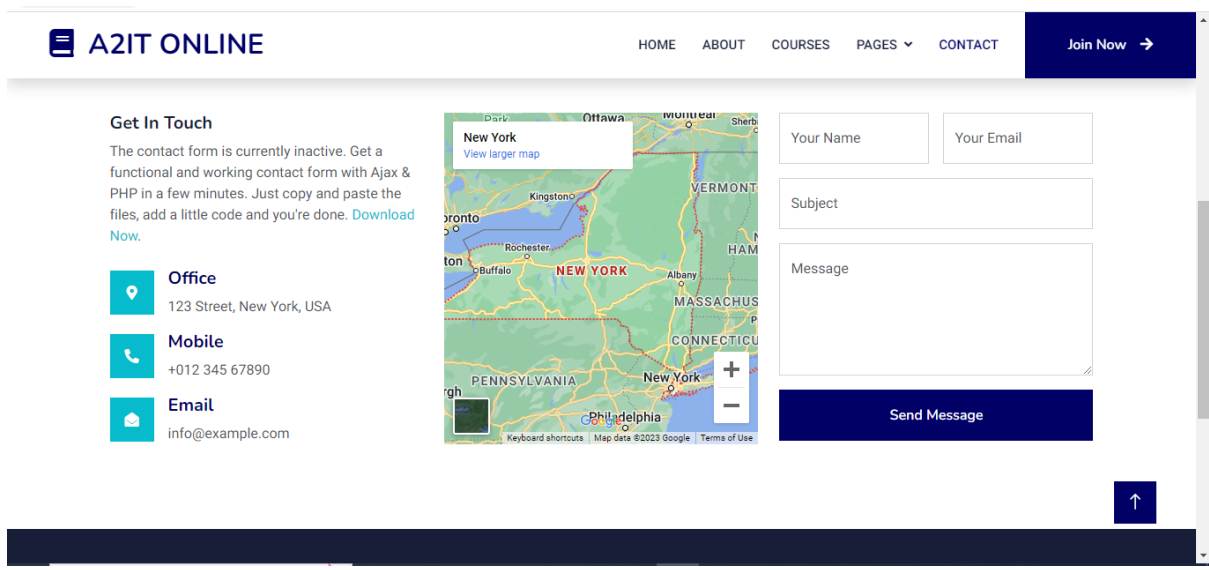


Figure:13

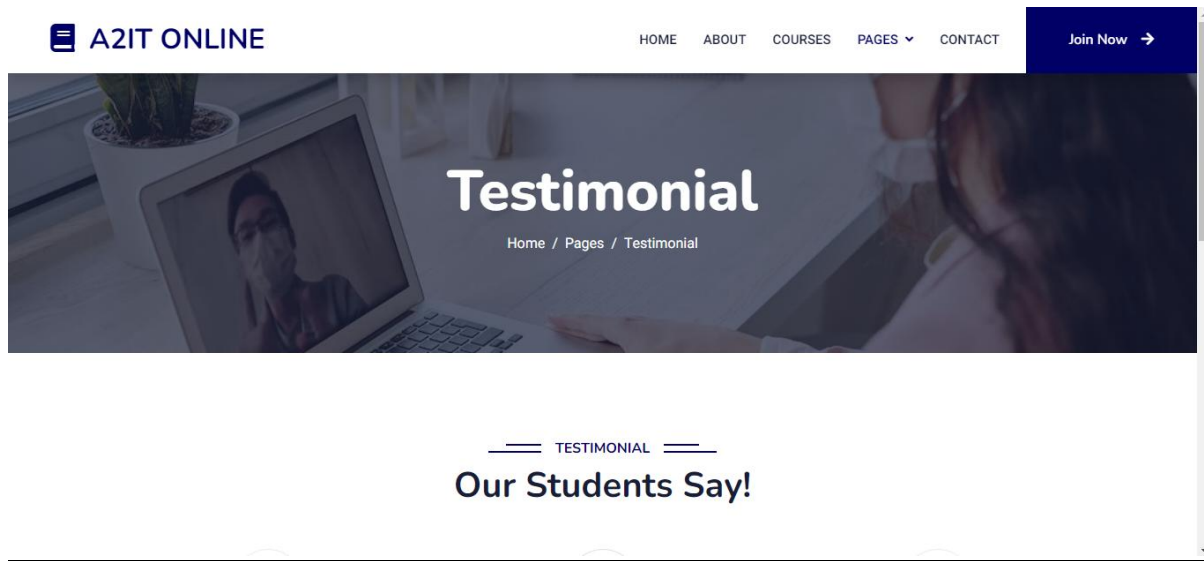


Figure:14

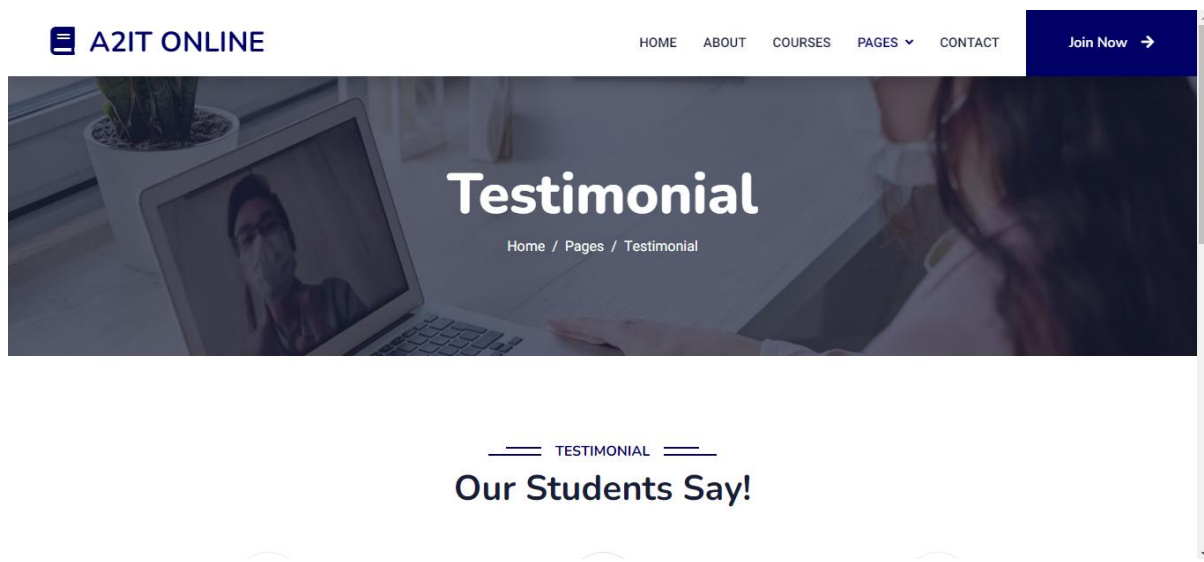
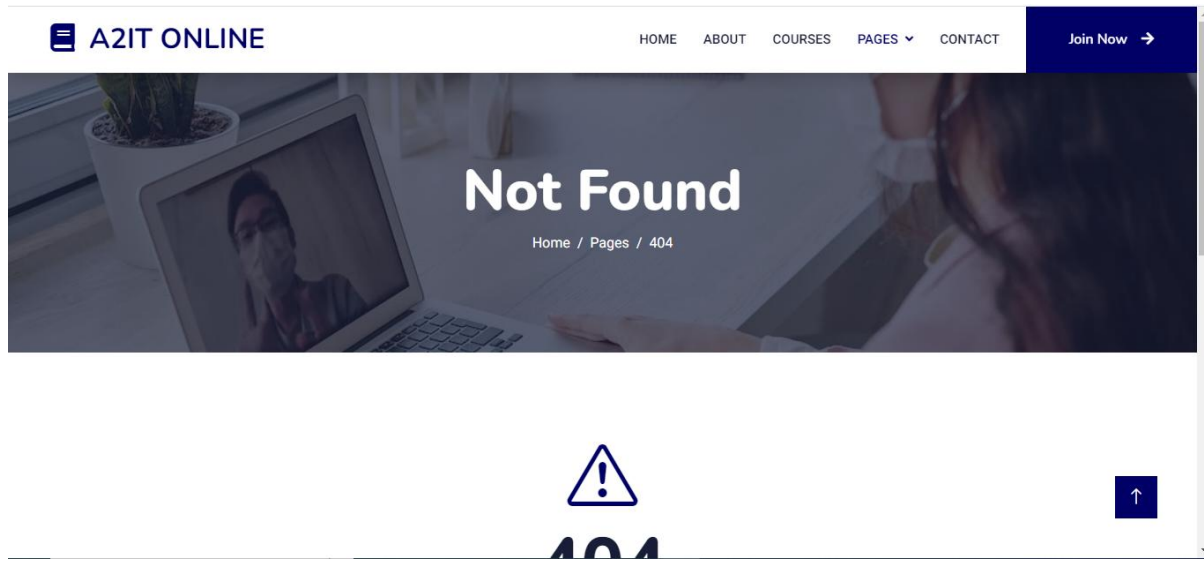


Figure:15



CONCLUSION

Technology has made significant progress over the years to provide consumers a better Industrial Technology Services and will continue to do so for years to come. With the rapid growth of products and brands, people have speculated that Industrial Technology Services will overtake in-Technology. While this has been the case in some areas, there is still demand for brick and mortar Technology in market areas where the consumer feels more comfortable seeing and touching the product being bought. However, the availability of online shopping has produced a more educated consumer that can shop around with relative ease without having to spend a large amount of time. In exchange, Industrial Technology Services has opened up doors to many small retailers that

would never be in business if they had to incur the high cost of owning a brick and mortar store. At the end, it has been a win-win situation for both consumer and sellers.

REFERENCES:

- ❖ <https://www.w3schools.com>
- ❖ <http://www.google.com>
- ❖ <https://getbootstrap.com/docs/4.0/getting/started/download/>
- ❖ <https://docs.djangoproject.com>