INDUSTRIAL TRAINING REPORT

On

Web Development

Made to demonstrate the project made in Industrial Training

Program of

B. Tech. in

ELECTRONICS AND COMMUNICATION ENGINEERING

SUBMITTED BY:

Manshi Shreya

2104122

ECE 3rd Year

Department of Electronics and Communication Engineering

National Institute of Technology Patna , Patna-800005

November 2023

राष्ट्रीय प्रौद्योगिकी संस्थान पटना
NATIONAL INSTITUTE OF TECHNOLOGY PATNA

CERTIFICATE

This is to certify that Manshi Shreya (Roll no. 2104122) has carried out the project entitled “Web Development” as his 5th semester industrial training under the supervision of B.C.Nagar , Assistant Professor, Department of Electronics and Communication Engineering, National Institute of Technology Patna and Kanchan Kumari, Senior Applications Engineer, Datex Technologies India Pvt. Ltd. , Patna. This project report is a bonafide work done by him for the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering.

Dr. B .C .NAGAR
(Project Supervisor)
U.G. Coordinator ,
NIT Patna

Dr. BHARAT GUPTA
(Head of Department)
Department of ECE,
NIT Patna

DATEX
Technologies Pvt. Ltd.

CIN - U74999BR2017PTC033626

INDUSTRIAL TRAINING
COMPLETION CERTIFICATE

This is to certify that Manshi Shreya has successfully completed his/her Training on Web Development with great efforts. Datex Technologies wishes good luck for your future endeavours. (Start Date: 01/06/23 – End Date: 13/07/23)

17.07.23

Date

DATEX TECHNOLOGIES PRIVATE LIMITED

Kanchan Kumari
(Director)

ACKNOWLEDGEMENT

Date: 28 November 2023

In the accomplishment of completion of my project on **Web Development** I would like to convey my special thanks of gratitude to my proffessors of National Institute of Technology Patna and to our supervisors of Datex Technology .

Your valuable guidance and suggestions helped me in various phases of the completion of this project. I will always be thankful to my friends in this regard for helping me in few tedious moment while completing this project.

I am ensuring that this project was finished by me and not copied.

Manshi Shreya

2104122

ECE-B

TABLE OF CONTENTS

Introduction

WEB DEVELOPMENT

Website is a collection of related web pages, including multimedia content, typically identified with a common domain name, and published on at least one web server. A website may be accessible via a public Internet Protocol (IP) network, such as the Internet, or a private local area network (LAN), by referencing a uniform resource locator (URL.) that identifies the site. Websites can have many functions and can be used in various fashions; a website can be a personal website, a commercial website for a company, a government website or a non-profit organization website. Websites are typically dedicated to a particular topic or purpose, ranging from entertainment and social networking to providing news and education. All publicly accessible websites collectively constitute the World Wide Web, while private websites, such as a company's website for its employees, are typically a part of an intranet.

CHAPTER 3- INTRODUCTION TO HTML

What is HTML

Stands for "Hypertext Markup Language." HTML is the language used to create webpages. "Hypertext" refers to the hyperlinks that an HTML page may contain. "Markup language" refers to the way tags are used to define the page layout and elements within the page.

Below is an example of HTML used to define a basic webpage with a title and a single paragraph of text.

```
<!doctype
<html>
<head>
<title>TechTerms.com</title>
</head>
<body>
<p>This Is an example of a paragraph in HTML.</p>
</body>
</html>
```

The first line defines what type of contents the document contains. "<!doctype html>" means the page is written in HTML5. Properly formatted HTML pages should include <html>, <head>, and <body> tags, which are all included in the example above. The page title, metadata, and links to referenced files are placed between the <head> tags. The actual contents of the page go between the <body> tags.

The web has gone through many changes over the past few decades, but HTML has always been the fundamental language used to develop webpages. Interestingly, while websites have become more advanced and interactive, HTML has actually gotten simpler. If you compare the source of an HTML5 page with a similar page written in HTML 4.01 or XHTML 1.0, the HTML5 page would probably contain less code. This is because modern HTML relies on cascading style sheets or JavaScript to format nearly all the elements within a page.

CHAPTER 4- INTRODUCTION TO CSS

What is CSS

CSS, stands for Cascading Style Sheet is a computer language to describe presentation (for example width, height, color, background color, alignment etc.) of HTML and XML (and XML based languages like XHTML, SVG) web documents.

CSS is a standard specified and maintained by **World Wide Web Consortium**.

Ways to attach **CSS** to an HTML web page.

1. Writing CSS code in a separate file (**CSS files are saved with .css extension**) and including that CSS file in head section of an HTML page using **<link>** element.

Advantages of CSS

1. **Separation of content form presentation** : Writing CSS code in another CSS file and attaching it to an HTML page, you can separate content from presentation. So, as an author, you need not be concerned about presentation and concentrate on content only.
2. **Consistency** : CSS can provide a consistent presentation for all of the pages of a web site.
3. The **Increment in accessibility** : If a particular page (or a number of pages) needs a different look and formatting, with a change of a single line, that can be achieved by calling more than one CSS for the same page.

---

4. **Save of bandwidth** : Since CSS separates content form style, it makes a web document lightweight, causing saving of bandwidth and in turn faster loading of the page.

Ease of contribution : Content Management Systems (for example WordPress) uses CSS, so that people without bothering how their content will look, can submit their content. This has caused an exponential increase in **User Generated Content**

CHAPTER 5-INTRODUCTION TO JAVASCRIPT

What is JavaScript ?

JavaScript is a cross-platform, object-oriented scripting language developed by Netscape. JavaScript was created by Netscape programmer **Brendan Eich**.

It was first released under the name of LiveScript as part of Netscape Navigator 2.0 in September 1995. It was renamed JavaScript on December 4, 1995. As JavaScript works on the client side, It is mostly used for client-side web development.

JavaScript is designed for use on web pages and closely integrated with HTML. JavaScript can create applications which run in the browsers such as IE, Opera, FireFox, Google Chrome and other. Netscape submitted JavaScript to ECMA International for standardization resulting in the standardized version named ECMAScript.

JavaScript and Java

JavaScript and Java are similar in some ways but fundamentally they are different. Java is a programming language developed by Sun Microsystems, Inc. and JavaScript is a scripting language developed by Netscape. Java is a server-side and static type language. JavaScript is a client-side, dynamically typed language. Java programs are compiled on the server and run on almost every platform without distribution of source code whereas scripts written in JavaScript are placed inside a HTML document and interpreted by the browser. The syntax, reserved-words of JavaScript and Java are also different.

JavaScript compares to Java

JavaScript	**Java**
JavaScript is used for front-end web development (for example field level validation in a HTML form).	Java is used as a back-end language within a web environment.
Interpreted (not compiled) by the client.	Compiled bytecodes downloaded from the

	server, executed on the client.
Object-oriented. No distinction between types of objects. Inheritance is through the prototype mechanism, and properties and methods can be added to any object dynamically.	Class-based. Objects are divided into classes and instances with all inheritance through the class hierarchy. Classes and instances cannot have properties or methods added dynamically.
Variable data types are not declared (loose typing).	Variable data types must be declared as Java maintains strong type checking.
Cannot automatically write to hard disk.	Cannot automatically write to hard disk.

CHAPTER 6-INTRODUCTION TO BOOTSTRAP

WHAT IS BOOTSTRAP?

Twitter Bootstrap is a front end framework to develop web apps and sites fast. In modern web development, there are several components which are required in almost all web projects. Bootstrap provides you with all those basic modules - Grid, Typography, Tables, Forms, Buttons, and Responsiveness. Besides, there are a plethora of other useful front- end components like Dropdowns, Navigation, Modals, Typehead, Pagination, Carousal, Breadcrumb, Tab, Thumbnails, Headers etc. With these, you can make a web project up and running quickly and easily.

Moreover, since the entire framework is module based, you can customize it with your own bit of CSS or even go for a complete overhaul after getting started.

It is based on the several best practices and we believe it is a very good point to start learning modern day web development with HTML, and JavaScript/Jquery once you know the basics.

Though there are criticisms, that all Bootstrap made projects looks same and you can make a website up without much of HTML+CSS knowledge, we need to understand that Bootstrap is a generic framework and like any other generic stuff, you need to customize it to look it exclusively. And you need to delveloped when you are on your way to customizing it and that is not feasible without a well understanding of HTML+CSS.

There are of course very good front-end frameworks available beside bootstrap and it's completely a developer's choice who wants which. But it is definitely worth trying.

CHAPTER 7- RSPONSIVE DESIGN

What Is Responsive Web Design?

Responsive Web design is the approach that suggests that design and development should respond to the user's behavior and environment based on screen size, platform and orientation.

The practice consists of a mix of flexible grids and layouts, images and an intelligent use of CSS media queries. As the user switches from their laptop to iPad, the website should automatically switch to accommodate for resolution, image size and scripting abilities. In other words, the website should have the technology to automatically *respond* to the user's preferences. This would eliminate the need for a different design and development phase for each new gadget on the market.

Almost every new client these days wants a mobile version of their website. It's practically essential after all: one design for the BlackBerry, another for the iPhone, the iPad, netbook, Kindle — and all screen resolutions must be compatible, too. In the next five years, we'll likely need to design for a number of additional inventions. When will the madness stop? It won't, of course.

In the field of Web design and development, we're quickly getting to the point of being unable to keep up with the endless new resolutions and devices. For many websites, creating a website version for each resolution and new device would be impossible, or at least impractical.

The Concept Of Responsive Web Design

Ethan Marcotte wrote an introductory article about the approach, Responsive Web Design, for A List Apart. It stems from the notion of responsive architectural design, whereby a room or space automatically adjusts to the number and flow of people within it:

Transplant this discipline onto Web design, and we have a similar yet whole new idea. Why should we create a custom Web design for each group of users; after all, architects don't design a building for each group size and type that passes through it?

Like responsive architecture, Web design should automatically adjust. It shouldn't require countless custom-made solutions for each new category of users. Obviously, we can't use motion sensors and robotics to accomplish this

the way a building would. Responsive Web design requires a more abstract way of thinking. However, some ideas are already being practiced: fluid layouts, media queries and scripts that can reformat Web pages and mark-up effortlessly (or automatically).

But responsive Web design is **not only about adjustable screen resolutions and automatically resizable images**, but rather about a whole new way of thinking about design.

Adjusting Screen Resolution

With more devices come varying screen resolutions, definitions and orientations. New devices with new screen sizes are being developed every day, and each of these devices may be able to handle variations in size, functionality and even color. Some are in landscape, others in portrait, still others even completely square. As we know from the rising popularity of the iPhone, iPad and advanced smartphones, many new devices are able to switch from portrait to landscape at the user's whim.

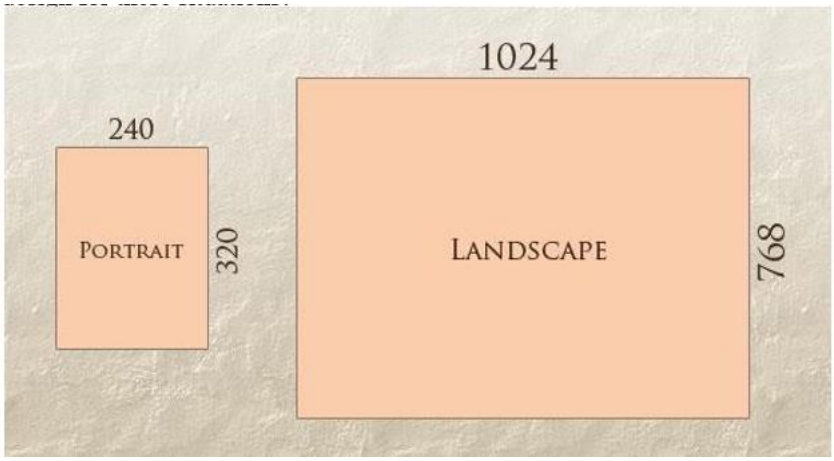

Fig 7.1 Different dimension of views

In addition to designing for both landscape and portrait (and enabling those orientations to possibly switch in an instant upon page load), we must consider the hundreds of different screen sizes. Yes, it is possible to group them into major categories, design for each of them, and make each design as flexible as necessary. But that can be overwhelming, and who knows what the usage figures will be in five years? Besides, many users do not maximize their browsers, which itself leaves far too much room for variety among screen sizes.

CHAPTER 8- Project: creating dynamic and responsive website for a restaurant

Introduction :

In this section I will talk about the projects I have created during my training . I have used HTML, CSS, Javascript and also Bootsrap framework .There are total of three projects-

1. Koo's landing page
2. To Do App
3. Resume Builder

Here are screenshots of my website in desktop browser .

Koo's Landing Page

I have tried to create a copy of the website of Koo landing page which is an Indian microblogging and social networking service, owned by Bangalore-based Bombinate Technologies , it empowers people to express their thoughts in a language of their choice, by making its style different from the existing one .It is a very simple project using HTML and CSS. The original webpage and new page both are attached. The new styling techniques had made it more appealing and eye - catchy. This landing page website, designed to showcase a particular website Koo and its features.

Features:

My product landing page has a header element with a corresponding id="header". Also attached company logo would make a good image . Within the #header element there is a nav element. There are three clickable elements inside the nav element, where we can add link for the next section. On clicking any them you will be directed to the corresponding section of the landing page.

It has a background image containing smartphones of latest models. A footer section showing copyright and privacy policies.

Technologies Used-

HTML5: For the structure of the web app page.

CSS: For styling the page beautifully.

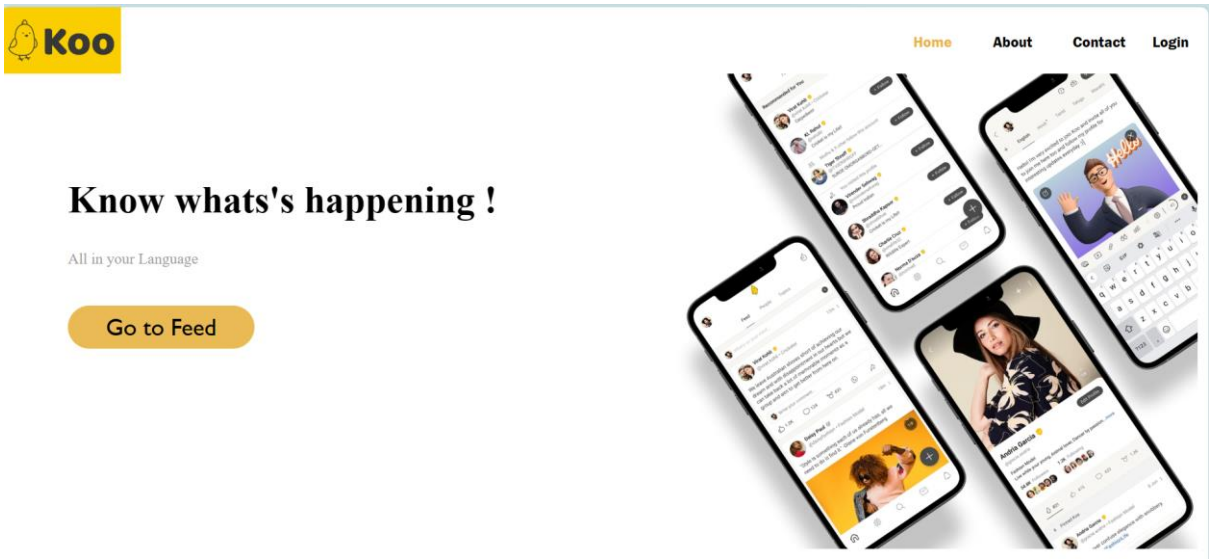

Fig. 8.1 Koo ‘s Landing Webpage

Future Aspects-

Mobile App Landing Page Template can have more features like link to Google Play, AppStore, Web App, cookie consent, automatic dark theme, privacy policy page.

To Do List App

It is a simple and elegant To Do List web app built using HTML, Tailwind CSS and JavaScript. The app allows users to create, manage, and save todos, all while enjoying a beautiful user interface design. The To-Do List Web App allows users to manage multiple to-do lists with their tasks. The task then saved in your browser's localStorage, so it won't be lost when you reload the page. However, clearing your browser's data will also delete the saved to-do list.

It contains following things-

HTML Structure: The basic HTML structure for your To-Do app is created. It includes an input field for adding tasks, a button to submit tasks, and a container to display the tasks.

CSS for body styling: It sets the background color, font family, and centers the content both horizontally and vertically.

JavaScript Functionality: Used various functions like addTask() function which T is called when the "Add Task" button is clicked.Retrieve Input and **Task List Elements**: Retrieves the input field and the task list container from the HTML document.Create a **New Task**: It creates a new li element with the text content set to the value entered in the input field. **Append the Task**: Appends the new task to the task list container**. Clear Input Field**: Resets the input field to an empty string after adding the task.Optional: **Delete Button**: Adds an optional delete button to each task. Clicking the delete button removes the associated task. **Local Storage**: Use localStorage to save tasks, allowing them to persist between page reloads.Load tasks from local storage when the page is loaded.These enhancements can be added to make our To-Do app more feature-rich and user-friendly.

Features

We can add new todos with a user-friendly interface, mark to do’s as completed, delete completed todos or those no longer needed, save to do’s to the local storage for persistent data.

Technologies Used-

HTML5: The structure of the web app.

CSS3 with Tailwind CSS: For styling the app beautifully.

JavaScript: To handle the interactive functionality of the app.

Local Storage: To save todos for persistent data across browser sessions.

Fig.8.2.1 To Do Notes Website

Future Aspects

We can add buttons like cross which can cut the to do ‘ s when completed, implement features like reminder at the scheduled time , and show the pending status.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>NotesApp</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="cont">
        <h1>Task-3 Notes App</h1>
        <div class="ipBox">
            <input type="text" id="Title" placeholder="Heading"><textarea name="" id="txtar" cols="30" rows="10" placeholder="Description"></
            textarea>
            <button id="btn">Add</button>
        </div>
        <div id="notes">
            <div id="note">
                <button class="del">delete</button>
                <div id="tit"></div>
                <div id="text"></div>
            </div>
        </div>
    </div>
  <script src="practice.js"></script>
</body>
</html>
```

Fig 8.2.2 Screenshot of VS code of the HTML page.

```
const addTitle= document.getElementById('Title');
const addText= document.getElementById('txtar');
const addBtn= document.getElementById('btn');
const addNotes= document.getElementById('notes');
showNotes(); // this will make notes history remain there even after refreshing
function addYourNotes(){
    let notes=localStorage.getItem('notes'); // array ko le rha h in the form of string from ocal storage
    if(notes===null){
        notes=[];
        // agar notes arr mila to thik nhi to fir ek arr khud bna lo }
    else{
        // but agr present ho to us string ko obj me convert kro
        notes=JSON.parse(notes);
    }
  if(addText.value==''){
        alert('Add your notes');
        return;
    }//upar koi v if hoga hi satisfy uske baad object creation
    const noteObj= {
        title:addTitle.value,
        text: addText.value,
    }
 addTitle.value='';
    addText.value='';
    //update notes
    notes.push(noteObj);
    localStorage.setItem('notes',JSON.stringify(notes));        //storage me dalne ke liye
    showNotes();   // only callong here will make the history gone on reloading , so call at top also.
    // console.log(note);

function showNotes(){
    let notesHTML='';
    let notes=localStorage.getItem('notes');  // it is because earlier it was a global scope and now it is local scope
```

Fig 8.2.3 Javascript code -1

```
  if(notes===null){
      return;  // agr koi notes mhi h to return
  }
  else{
      notes=JSON.parse(notes);   // convert string to arr or obj
  }
  for(let i=0;i<notes.length;i++){
/iterate each note and print in the box of delete.

      notesHTML+= // for deletion
      `<div id="note">
                      <button class="del" id=${i} onclick="deleteNote(${i})">delete</button>
                      <div id="tit">${notes[i].title===""  ? 'Note':notes[i].title}</div>
                      <div id="text">${notes[i].text}</div>
                  </div>`
  }
                  addNotes.innerHTML= notesHTML;
  }
 function deleteNote(ind){
      let notes= localStorage.getItem('notes');
      if(notes===null){
          return;
      }
      else{
          notes=JSON.parse(notes);   // string ko obj ya arr me convert
      }
      notes.splice(ind,1);     //  del notes from index ind till count 1
      localStorage.setItem('notes',JSON.stringify(notes));    // update local storage by converting arr into string
      showNotes();            // print updated notes
  }
  addBtn.addEventListener('click',addYourNotes);
```

Fig 8.2.4 Javascript code-2

Resume Builder

Creating a resume is a bit tedious task for any working professional from any industry. One has to keep it short, simple, and with the latest work experience. Resume builder allows user to create a modern professional resume easily.It empowers you to easily create your job-winning online resume, showcase your academic publications, and create online courses or knowledge bases to grow your audience.Build your resume without registration, Save as json file and upload and use in future, Export as PDF (selectable text).A static single-page application resume-builder developed using HTML, CSS,JS.Resume content can be edited just like a normal document editor (cut,copy,undo etc).

Entire sections can be added, reordered, removed just by cut,copy,pasting method.I have created a user interface where users can input their information, dynamically generate a resume based on that input, and potentially allow users to download the generated resume.

HTML Structure:

Create the basic HTML structure with form elements to capture the user's information. Use appropriate input types for different pieces of information (text, textarea, etc.).Include a button to trigger the resume generation.

CSS Styling:

Create a stylesheet (styles.css) to style your HTML elements for visually appealing layout and better presentation.

JavaScript Functionality:

Write JavaScript code (script.js) to handle form submission, process user input, and dynamically generate the resume.

Testing and Refinement:

Test the resume builder in different browsers to ensure compatibility.Ensure the layout is responsive and works well on various devices.Refine the code and styling based on testing feedback.

Enhancements (Optional):

Allow users to dynamically add multiple entries for education, experience, etc.Implement a feature to download the generated resume as a PDF.

Features-

1. Real Time UI Update - The resume PDF is updated in real time as you enter your resume information, so you can easily see the final output.

2. Modern Professional Resume Design - The resume PDF is a modern professional design that adheres to U.S. best practices and is ATS friendly to top ATS platforms such as Greenhouse and Lever. It automatically formats fonts, sizes, margins, bullet points to ensure consistency and avoid human errors.

3. Privacy Focus - The app only runs locally on your browser, meaning no sign up is required and no data ever leaves your browser, so it gives you peace of mind on your personal data.

4.Additional links- If you would like to add more than the predefined columns in work experience , skills and education in the config file, then you can use the Add field to add as many as you wish.

5. Successful Track Record- OpenResume will help the users to land interviews and offers from top companies. It can be used at work and liken by recruiters and hiring managers.

Technologies Used-

JavaScript - It helps in adding and deleting features according to the user's requirement.You can start over,preview and save it also.

CSS Framework Tailwind CSS - Tailwind speeds up development by providing helpful CSS utilities and removing the need to context switch between tsx and CSS files.

PDF Renderer- Html2pdf And HtmlCanvas, it creates PDF files and is used by the resume builder to create a downloadable PDF file.

We have used a function in JavaScript window.location which can be used-to get current page address (URL), to reload the same page

```html
<!-- TEMPLATE -->
<div class="container  mt-2 border rounded" id="resume-template">
    <div class="row  mx-3 my-3">
        <!--COL-1 ==>  4 of 12 grid columns [Personal template]-->
        <div class="col-md-4 background text-center ">
            <img src="/img/blank-profile-picture.webp" id="profile-img-template" class="profile-pic text-center"
                alt="">

            <div class="container">
                <h3 id="full-name-template">John Doe</h3>
                <table class="table table-borderless">
                    <tbody>
                        <tr class="text-start">
                            <th scope="row">DOB</th>
                            <td id="dob-template" class="text-start">15-12-1997</td>
                        </tr>
                        <tr class="text-start">
                            <th scope="row">Phone</th>
                            <td id="phone-template" class="text-start">9998887776</td>
                        </tr>
                        <tr class="text-start">
                            <th scope="row">Email</th>
                            <td id="email-template" class="text-start">example@domain.com</td>
                        </tr>
                        <tr class="text-start">
                            <th scope="row">Address</th>
                            <td id="address-template" class="text-start">New York</td>
                            <!-- <td colspan="2">Larry the Bird</td> -->
                        </tr>
```

Fig 8.3.1 HTML code of Resume Builder

```javascript
  if(notes===null){
      return;  // agr koi notes mhi h to return
  }
  else{
      notes=JSON.parse(notes);   // convert string to arr or obj
  }
  for(let i=0;i<notes.length;i++){
/iterate each note and print in the box of delete.

      notesHTML+= // for deletion
      `<div id="note">
                      <button class="del" id=${i} onclick="deleteNote(${i})">delete</button>
                      <div id="tit">${notes[i].title==="" ? 'Note':notes[i].title}</div>
                      <div id="text">${notes[i].text}</div>
                  </div>`
  }
                  addNotes.innerHTML= notesHTML;
  }
  function deleteNote(ind){
      let notes= localStorage.getItem('notes');
      if(notes===null){
          return;
      }
      else{
          notes=JSON.parse(notes);   // string ko obj ya arr me convert
      }
      notes.splice(ind,1);     //  del notes from index ind till count 1
      localStorage.setItem('notes',JSON.stringify(notes));    // update local storage by converting arr into string
      showNotes();            // print updated notes
  }
  addBtn.addEventListener('click',addYourNotes);
```

Fig 8.3.2 Javascript code of Resume Builder

```
function generateResume() {
  console.log("generateResume() <<");
  let fullName = document.getElementById("full-name").value;
  let fullNameTemplate = document.getElementById("full-name-template");
  fullNameTemplate.innerHTML = fullName;
  fileName = fullName;

  let dob = document.getElementById("dob").value;
  let dobTemplate = document.getElementById("dob-template");
  dobTemplate.innerHTML = dob;

  let phone = document.getElementById("phone").value;
  let phoneTemplate = document.getElementById("phone-template");
  phoneTemplate.innerHTML = phone;

  let email = document.getElementById("email").value;
  let emailTemplate = document.getElementById("email-template");
  emailTemplate.innerHTML = email;

  let address = document.getElementById("address").value;
  let addressTemplate = document.getElementById("address-template");
  addressTemplate.innerHTML = address;

  let linkedin = document.getElementById("linkedin").value;
  let linkedinTemplate = document.getElementById("linkedin-template");
  linkedinTemplate.innerHTML = linkedin;

  let github = document.getElementById("github").value;
  let githubTemplate = document.getElementById("github-template");
```

Fig 8.3.3 Javascript code of Resume Builder

Resume Builder

Build your resume with this free online resume maker in few minutes

Enter Your Personal Details

Name

Manshi Shreya

DOB

12-09-2003

Phone Number

+917982431033

E-Mail

shmanshi1@gmail.com

Your email will be secured.

Address

Purandaha, Deoghar , Jharkhand, 814112

Profile Picture

Choose File | No file chosen

Links

LinkedIn

www.manshi//20.s.com

Github

manshi//github20.com

Professional Details

Objective

To be successful in life

Skills

Enter your skills | Add

- Communication
- Leadership
- Web development
- Machine Learning - Python

Experience

N/A

Add Experience | Delete Experience

Education

B.tech - IIT Kharagpur

12th- D.A.V Public School Deoghar

10th- D.A.V Public School Deoghar

Add Education | Delete Education

Reset | Generate Resume

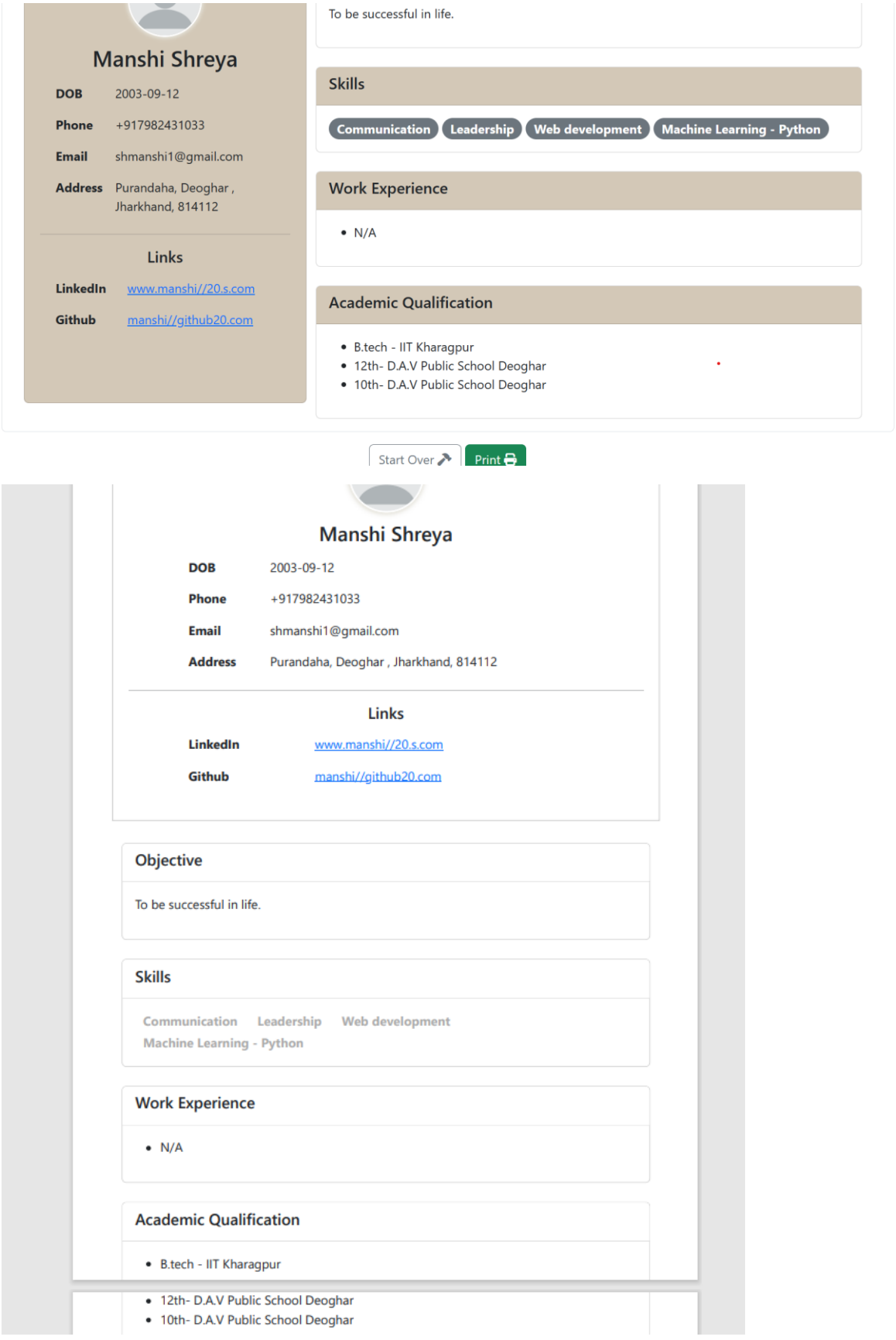

Fig. 8.3.4 Resume builder website image

CONCLUSION

In a nutshell, this internship has been an excellent and rewarding experience. I can conclude that there have been a lot I've learnt from my work at the training & research centre. Needless to say, the technical aspects of the work I've done are not flawless and could be improved provided enough time. As someone with no prior experience in JavaScript whatsoever I believe my time spent in training and discovering new languages was well worth it and contributed to finding an acceptable solution to an important aspect of web design and development. Two main things that I've learned the importance of time-management skills and self-motivation. Although I have often stumbled upon these problems at University, but they had to be approached differently in a working environment

REFERENCES

1.https://www.yhills.com

2.www.w3schools.com

3.www.tutotialpoint.com

4.www.youtube.com

5.www.wikipedia.org

****THANK YOU****