

BIRAJEE TRAVELS

*A Summer Internship Report submitted in partial fulfillment of the
requirements for the award of degree of*

**BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING**

**Submitted
by**

**PUSPENDRA BIRAJEE
22A91A05J7**



**A D I T Y A
U N I V E R S I T Y**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ADITYA UNIVERSITY

(Formerly Aditya Engineering College (A))

2024-2025

ADITYA UNIVERSITY

(Formerly Aditya Engineering College(A))

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Internship report entitled “**BIRAJEE TRAVELS**” is being submitted by **PUSPENDRA BIRAJEE (22A91A05J7)**. In partial fulfillment of the requirements for award of the B. Tech degree in COMPUTER SCIENCE AND ENGINEERING for the academic year 2024-2025.

Internship Coordinator

A. Phani Sridhar, M.Tech, (Ph.D)
Assistant professor
Department of CSE

Head of the Department

Dr. K. Swaroopa, M.Tech, Ph.D
Associate Professor
Department of CSE

DECLARATION

We hereby declare that the internship on “**JAVA FULL STACK**” is a genuine internship. This work has been submitted to the **ADITYA ENGINEERING COLLEGE**, Surampalem, permanently affiliated to **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA** in partial fulfillment of the **B. Tech** degree. We further declare that this internship work has not been submitted in full or part of the award of any degree of this or any other educational institutions.

by

PUSPENDRA BIRAJEE

(22A91A05J7)

Internship Completion Certificate



CERTIFICATE OF INTERNSHIP COMPLETION

Date: 12-08-2024

TO

Puspendra Birajee
Aditya Engineering College surampalem Andhra Pradesh India

This is to certify that **Puspendra Birajee** has successfully completed his **TWO MONTHS INTERNSHIP** program with **BrainOvision Solutions Pvt. Ltd.** He has worked on **JAVA FULL STACK** and was actively & diligently involved in the projects and tasks assigned to him. During the span, we found his punctual and hardworking person. His feedback and evolution proved that he is a quick learner. Congratulations and Best Wishes.

ROLE : **JAVA FULL STACK INTERN**
INTERN ID : **BV24MS0088**
START DATE : **06-06-2024**
END DATE : **11-08-2024**

Yours Faithfully


Ganesh Nag Daddi
Founder & CEO
Brainovision Solutions India Pvt Ltd




Dr. Buddha Chandrashekar
Chief Coordinating Officer – AICTE
All India Council for Technical Education

ACKNOWLEDGEMENT

First, I would like to thank the Director of Organization **BRAIN O VISION, Hyderabad** for giving me the opportunity to do an internship within the organization. I also would like all the people that worked with me in **BRAIN O VISION, Hyderabad** with their patience and openness they created an enjoyable working environment.

It is with immense pleasure that we would like to express my indebted gratitude to our internship coordinator **A. Phani Sridhar, Assistant Professor in CSE** who has guided us a lot and encouraged us in every step of the intern project work, his valuable moral support and guidance throughout the Intern project helped us to a greater extent.

I am grateful to **Dr. K. Swaroopa, Associate Professor and HOD** for inspiring us all the way and for arranging all the facilities and resources needed for my intern project work.

I wish to thank our **Dr. M.V. Rajesh, Associate Dean** and **Dr. Dola Sanjay, Dean School of Engineering** for their encouragement and support during the course of my intern project work.

I would like to extend my sincere thanks to **Dr. G. Suresh, Registrar, Dr. S. Rama Sree, Pro Vice-Chancellor, Dr. M.B. Srinivas, Vice-Chancellor, Dr. M. Sreenivasa Reddy, Deputy Pro Chancellor and Management**, Aditya University for unconditional support for providing me the best infrastructural facilities and state of the art laboratories during my intern project work.

Not to forget, **Faculty, Lab Technicians, Non-Teaching Staff and our Friends** who have directly or helped and supported us in completing my intern project work in time.

ABSTRACT

The advent of digital technology has revolutionized education, ushering in an era where traditional classroom settings are no longer the sole source of knowledge dissemination. Learning platforms have emerged as powerful tools that have transformed the way individuals' access, engage with, and acquire knowledge. This abstract explores the significant impact of learning platforms on modern learning, shedding light on their benefits, challenges, and the evolving landscape of education.

Learning platforms encompass a wide range of digital tools and resources, including online courses, virtual classrooms, interactive tutorials, and educational apps. These platforms have democratized education by providing accessible and flexible learning opportunities to individuals across the globe. They offer personalized learning experiences, catering to diverse learning styles and paces, thus breaking down the barriers to education that were once dictated by geographical location or socioeconomic factors.

However, the proliferation of learning platforms is not without its challenges. Issues related to digital equity and accessibility need to be addressed to ensure that all learners, regardless of their background, have equal opportunities to benefit from these platforms. Additionally, the sheer volume of available content can be overwhelming, necessitating effective curation and guidance to help learners navigate the digital learning landscape.

The ongoing evolution of learning platforms continues to shape the future of education. Advancements in artificial intelligence and data analytics are enabling platforms to provide even more personalized learning experiences. The integration of augmented and virtual reality is redefining the possibilities of immersive education. Furthermore, the emergence of block chain technology holds promise for the secure and verifiable validation of skills and credentials earned through these platforms.

Learning Objectives/Internship Objectives

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more.
- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
- Some internships are used to allow individuals topper form scientific research while others are specifically designed to allow people to gain first-hand experience working.
- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landingthe position.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

1 st WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	03/06/24	Monday	Introductory Session. Overview to Company Profile & Total Internship Schedule
	04/06/24	Tuesday	Introduction of Git and GitHub tools
	05/06/24	Wednesday	Version Control System
	06/06/24	Thursday	Workflow
	07/06/24	Friday	Branching & Merging
	08/06/24	Saturday	Overview about Git & GitHub

2 nd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	10/06/24	Monday	Introduction of HTML
	11/06/24	Tuesday	Page Structure, Head and Body
	12/06/24	Wednesday	Elements and Tags
	13/06/24	Thursday	Attributes, Heading and Paragraphs
	14/06/24	Friday	Links, Images and Lists
	15/06/24	Saturday	Iframes and Forms

3rdWEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	17/06/24	Monday	Introduction of CSS
	18/06/24	Tuesday	Structure and Basic Styles
	19/06/24	Wednesday	Inline, Internal, External-CSS
	20/06/24	Thursday	Basic Styles
	21/06/24	Friday	Box Model, Height and Width
	22/06/24	Saturday	Display and Position

4thWEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	24/06/24	Monday	Transforms and Animations
	25/06/24	Tuesday	Transforms and Animations
	26/06/24	Wednesday	Media Queries
	27/06/24	Thursday	Media Queries
	28/06/24	Friday	Various Margins and Paddings of CSS
	29/06/24	Saturday	Overview about CSS

5thWEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	01/07/24	Monday	Introduction to JAVASCRIPT
	02/07/24	Tuesday	Operators and Controls Statements
	03/07/24	Wednesday	Function and Objects
	04/07/24	Thursday	Array and Strings
	05/07/24	Friday	Events, Data Methods and Hoisting
	06/07/24	Saturday	Events, Data Methods and Hoisting

6thWEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	08/07/24	Monday	Events, Data Methods and Hoisting
	09/07/24	Tuesday	JavaScript in the browser
	10/07/24	Wednesday	Document Object Model
	11/07/24	Thursday	Walking the DOM
	12/07/24	Friday	Events & Other DOM Properties
	13/07/24	Saturday	Callback, Promise

7thWEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	15/07/24	Monday	Introduction to JAVA
	16/07/24	Tuesday	Java Syntax
	17/07/24	Wednesday	Java Comments
	18/07/24	Thursday	Java Variables and Data Type
	19/07/24	Friday	Java Operators
	20/07/24	Saturday	Java Conditional and Looping Statement

8thWEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	22/07/24	Monday	Java Methods
	23/07/24	Tuesday	Java Classes
	24/07/24	Wednesday	Objected Oriented Programming
	25/07/24	Thursday	Java File Handling
	26/07/24	Friday	Brief Introduction of Travel Agency
	27/07/24	Saturday	Overall view on all topics

INDEX

SNo no.	Contents	Page
1.	Introduction	
1.1	HTML	1
1.2	Key Resources	2
1.3	CSS	3
1.4	JavaScript	4
1.5	Java	5
2.	Birajee Travel	6
3.	Web Development	
3.1	History	7
3.2	Web Designing	8
4.	Frontend Developers	
4.1	Frontend Developer Role	9
4.2	Required Skills	10
5.	Backend Developers	
5.1	Introduction	11
5.2	Programming Languages	12
5.3	Frameworks	13
5.4	Databases	13
5.5	Web server	14
6.	Full Stack Developers	15

7.	Web Development Tools	
7.1	Introduction	16
7.2	Development Tools	17
8.	Web Security	18
9.	Responsive Web Design Ability	19
10.	Gallery	
10.1	Birajee Travel	20
10.2	Footer	21
11.	Summary	22
12.	About the Company	23
13.	Opportunities	24
14.	Training	25
15.	Challenge faced	26
16.	Conclusion	27

1. INTRODUCTION

1.1. What is HTML?

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behaviour (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, , , <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <search>, <output>, <progress>, <video>, , , and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case-insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. However, the convention and recommended practice is to write tags in lowercase.

1.2. Key resources

1.2.1. HTML Introduction

If you're new to web development, be sure to read our HTML Basics article to learn what HTML is and how to use it.

1.2.2 HTML Tutorials

For articles about how to use HTML, as well as tutorials and complete examples, check out our HTML Learning Area.

1.2.3. HTML Reference

In our extensive HTML reference section, you'll find the details about every element and attribute in HTML.

1.2.4. Multimedia and embedding

This module explores how to use HTML to include multimedia in your web pages, including the different ways that images can be included, and how to embed video, audio, and even entire other webpages.

1.2.5. HTML tables

Representing tabular data on a webpage in an understandable, accessible way can be a challenge. This module covers basic table markup, along with more complex features such as implementing captions and summaries.

1.2.6. HTML forms

Forms are a very important part of the Web — these provide much of the functionality you need for interacting with websites, e.g. registering and logging in, sending feedback, buying products, and more. This module gets you started with creating the client-side/front-end parts of forms.

Use HTML to solve common problems

Provides links to sections of content explaining how to use HTML to solve very common problems when creating a web page: dealing with titles, adding images or videos, emphasizing content, creating a basic form, etc.

1.2.7. HTML reference

HTML consists of elements, each of which may be modified by some number of attributes. HTML documents are connected to each other with links.

1.2.8. HTML element reference

Browse a list of all HTML elements.

1.2.9. HTML attribute reference

Elements in HTML have attributes. These are additional values that configure the elements or adjust their behavior in various ways.

1.2.10. Global attributes

Global attributes may be specified on all HTML elements, even those not specified in the standard. This means that any non-standard elements must still permit these attributes, even though those elements make the document HTML5-noncompliant.

Inline-level elements and block-level elements

HTML elements are usually "inline-level" or "block-level" elements. An inline-level element occupies only the space bounded by the tags that define it. A block-level element occupies the entire space of its parent element (container), thereby creating a "block box".

Guide to media types and formats on the web

The <audio> and <video> elements allow you to play audio and video media natively within your content without the need for external software support.

1.2.11. HTML content categories

HTML is comprised of several kinds of content, each of which is allowed to be used in certain contexts and is disallowed in others. Similarly, each context has a set of other content categories it can contain and elements that can or can't be used in them. This is a guide to these categories.

1.3. What is CSS?

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; rather, everything is now CSS without a version number.

After CSS 2.1, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to develop and release recommendations separately per module. Instead of versioning the CSS specification, W3C now periodically takes a snapshot of the latest stable state of the CSS specification and individual modules progress. CSS modules now have version numbers, or levels, such as CSS Color Module Level 5.

1.3.1. CSS Tutorials

Our CSS learning area contains a wealth of tutorials to take you from beginner level to proficiency, covering all the fundamentals.

1.3.2. CSS Reference

Our exhaustive CSS reference for seasoned Web developers describes every property and concept of CSS.

1.3.3. CSS styling text

With the basics of the CSS language covered, the next CSS topic for you to concentrate on is styling text — one of the most common things you'll do with CSS. Here we look at text styling fundamentals, including setting font, boldness, italics, line and letter spacing, drop shadows, and other text features. We round off the module by looking at applying custom fonts to your page, and styling lists and links.

1.3.4. CSS layout

At this point we've already looked at CSS fundamentals, how to style text, and how to style and manipulate the boxes that your content sits inside. Now it's time to look at how to place your boxes in the right place in relation to the viewport, and to each other. We have covered the necessary prerequisites so we can now dive deep into CSS layout, looking at different display settings, modern layout tools like flexbox, CSS grid, and positioning, and some of the legacy techniques you might still want to know about.

1.4. What is JAVASCRIPT?

JavaScript (JS) is a lightweight interpreted (or just-in-time compiled) programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. JavaScript's dynamic capabilities include runtime object construction, variable parameter lists, function variables, dynamic script creation (via `eval`), object introspection (via `for...in` and Object utilities), and source-code recovery (JavaScript functions store their source text and can be retrieved through `toString()`).

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

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

1.4.1. Functions

Generally speaking, a function is a "subprogram" that can be called by code external (or internal, in the case of recursion) to the function. Like the program itself, a function is composed of a sequence of statements called the function body. Values can be passed to a function as parameters, and the function will return a value. In JavaScript, functions are first-class objects, because they can be passed to other functions, returned from functions, and assigned to variables and properties. They can also have properties and methods just like any other object. What distinguishes them from other objects is that functions can be called. For more examples and explanations, see the JavaScript guide about functions.

1.4.2. Return value

By default, if a function's execution doesn't end at a return statement, or if the return keyword doesn't have an expression after it, then the return value is undefined. The return statement allows you to return an arbitrary value from the function. One function call can only return one value, but you can simulate the effect of returning multiple values by returning an object or array and destructuring the result.

1.5. What is JAVA?

Java is a popular programming language, created in 1995. It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications (especially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

1.5.1. Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming languages in the world
- It has a large demand in the current job market
- It is easy to learn and simple to use
- It is open-source and free.

- It is secure, fast and powerful.
- It has huge community support (tens of millions of developers)
- Java is an object-oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

1.5.2. Features of Java

- Simple
- Object-Oriented
- Portable
- Platform independent
- Secured
- Interpreted
- High Performance
- Multithreaded
- Distributed
- Dynamic

1.5.3. Java - What is OOP?

OOP stands for Object-Oriented Programming.

Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

Object-oriented programming has several advantages over procedural programming:

- OOP is faster and easier to execute
- OOP provides a clear structure for the programs
- OOP helps to keep the Java code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
- OOP makes it possible to create full reusable applications with less code and shorter development time

1.5.4. Java Classes/Objects

Java is an object-oriented programming language.

Everything in Java is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.

A Class is like an object constructor, or a "blueprint" for creating objects.

2. Birajee Travel

Birajee Travel is a dedicated platform designed to streamline travel planning, enabling users to explore destinations, book accommodations, and manage itineraries with ease. This service is ideal for international travelers, business professionals, and adventure seekers looking for comprehensive travel solutions. With access to reliable travel information, up-to-date destination insights, and personalized services, Birajee Travel is a must-have tool for anyone looking to make the most of their travel experiences.

- **Purpose:** To assist travelers in planning and organizing trips efficiently with accurate information and seamless booking options.
- **Features:** Includes real-time destination updates, personalized recommendations, and a booking system to cover all travel needs.
- **Types of Services:** Available as a website, mobile app, and integrated travel management tool for both leisure and business travelers.
- **Uses:** Useful for vacation planning, business travel, group tours, and educational trips.
- **Factors Influencing Services Efficiency:** Dependable travel information, customer feedback, global travel trends, and industry partnerships.

2.1. Advantages of Birajee Travel

- i. Real-Time Destination Insights
- ii. Convenience
- iii. Informed Decision-Making
- iv. Travel Planning
- v. Support for Group Travel and Events
- vi. Ease of International Travel
- vii. Transparent Costs

2.2. Disadvantages of Birajee Travel

- i. Service Fees and Additional Costs
- ii. Constantly Changing Conditions
- iii. No Offline Functionality
- iv. Limited Destination Coverage
- v. Over-reliance on Automation
- vi. Potential for Overbooking

3. WEB DEVELOPMENT

Web development refers to the creating, building, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

Web development, also known as website development, refers to the tasks associated with creating, building, and maintaining websites and web applications that run online on a browser. It may, however, also include web design, web programming, and database management.

Web development is closely related to the job of designing the features and functionality of apps (web design). The term development is usually reserved for the actual construction of these things (that is to say, the programming of sites).

The basic tools involved in web development are programming languages called HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript. There are, however, a number of other programs used to “manage” or facilitate the construction of sites that would otherwise have to be done “from scratch” by writing code. A number of content management systems (CMS) fall into this category, including WordPress, Joomla! Drupal, TYPO3, and Adobe Experience Manager, among others.

The web is just the common name many people use for the World Wide Web (WWW), a division of the Internet consisting of pages that can be accessed by a web browser. Commonly people often regard the web and internet as the same whereas it's not. However, the Internet is the global network of various servers that makes data(information) sharing that happens over the internet possible.

The Internet is used for many things, such as electronic mail, online chat, file transfer, and other documents of the World Wide Web. It is the biggest worldwide communication network of Computers.

Web Development simply means writing mark-up and coding. It refers to the non-design aspects of building websites. It is the work involved in creating a website for the internet(www) or an intranet(a private network).

The knowledge of web tools that are required in basic web development includes HTML, CSS, JavaScript, and many more. These tools are required by beginners in career building. This knowledge is the criterion used in categorizing web developers into three main categories which are:-

- Front-end Developers: responsible for behaviour and visuals.
- Back-end Developers: deal with servers.
- Full-stack Developers: a combination of both categories above.

These categories are based on certain knowledge of web languages.

3.1. Brief History on Web Development

The web was developed by Tim Berners-Lee, a British scientist in 1989 while working at CERN as just a proposal. The Web was originally conceived and developed to meet the demand for automated information-sharing between scientists in universities and institutes around the world. His second proposal was in 1990, this outlined the principal concepts and it defined important terms behind the Web. The document described a “hypertext

project” called “World Wide Web” in which a “web” of “hypertext documents” could be viewed by “browsers”. The web development hierarchy is as follows:

- Client-side coding.
- Server-side coding.
- Database technology.

Most web developers use Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript (to develop the interface) and PHP, SQL, Node.js, and RUBY(for the server-side of a website).

3.2. Web Designing

Web designing is the process of planning, conceptualizing, and implementing the plan for designing a website in a way that is functional and offers a good user experience. User experience is central to the web designing process. Websites have an array of elements presented in ways that make them easy to navigate. Web designing essentially involves working on every attribute of the website that people interact with, so that the website is simple and efficient, allows users to quickly find the information they need, and looks visually pleasing. All these factors, when combined, decide how well the website is designed.

A beginner should understand that there are many skills required to make his/her career and learning of languages easy. However, the skills needed to become a web designer differs from those needed to become a web developer; though similar in some ways.

The following are the skills needed to become a web designer:

- Imagination
- Creativity
- Patience
- Attention to detail
- Analytical skill
- Communications skills
- Technical ability
- Excellent IT skills
- SEO knowledge

4. FRONT END DEVELOPERS

The front end of a website is everything the user either sees or interacts with when they visit the website. It is responsible for the total look and feel of an online experience.

While the term may sound a little technical, website front ends are really an everyday encounter for almost all of us. If you've opened an app to check the weather, or scrolled through today's headlines on a website, you've interacted with a front end.

Put simply, the front end is a combination of two different elements: the graphic design (the look) and the user interface (the feel). Each of these is created independently, with most of the technical work going into the user interface using web languages like HTML, CSS, and JavaScript.

4.1. Defining the Front-end Developer Role

Fundamentally a front-end developer, historically sometimes referred to as a client-side developer, produces the code sent from a web server to a client/front-end to establish a user interface. For example, when you load a webpage into a web browser, code is sent from a web server to the device/client requesting the page. The code sent will produce the user interface one sees in the web browser. This code is considered front-end code and is a mixture of the web technologies HTML, CSS, and JavaScript.

An individual typically enters into the field of front-end development by first learning to build a simple website using HTML and CSS. HTML or HyperText Markup Language is likened to the structural framing of a house, while CSS or Cascading Style Sheets can be considered the finishing touches one sees once a home is complete.

As a front-end developer progresses in their career, they will move from a simple understanding of web pages and websites to potentially a robust grasp of software development practices as it is done on the web platform using the JavaScript programming language.

The web platform runtime is not just for web browsers. This means that a front-end developer can take their knowledge of web technologies and the web platform run time associated with browsers and web development and use these skills to create native applications for popular operating systems like Windows, Mac OS X, iOS, Android, and Linux.

The most common solutions for WebView development are:

- Electron
- NW.js
- Neutralino.js
- Desk Gap

Front-end developers can also find themselves developing source code using web platform technologies (e.g., CSS and JavaScript), but then the source code created is not used by a web platform run time scenario but instead as input which gets translated to native run time code. Some examples of these environments are:

- Flutter
- Native Script
- React Native (The Instagram application for iOS and Android was developed using React Native)

4.2. Required Skills

A professional front-end developer will minimally have a working knowledge of browsers, the internet and be skilled at using the following web technologies:

- Hypertext Markup Language (aka HTML)
- Cascading Style Sheets (aka CSS)
- Uniform Resource Locators (aka URLs)
- Hypertext Transfer Protocol (aka HTTP)
- JavaScript Programming Language (aka ECMAScript 262)
- JavaScript Object Notation (aka JSON)
- Document Object Model (aka DOM)
- Web APIs (aka Browser APIs)
- Web Content Accessibility Guidelines (aka WCAG) & Accessible Rich Internet Applications (aka ARIA)
- Beyond being skilled at the technologies just mentioned, a front-end developer might also be skilled in one or more of the following areas:
 - Content Management Systems (aka CMS)
 - Node.js
 - Cross-Browser Testing
 - Cross-Platform Testing
 - Unit Testing
 - Cross-Device Testing
 - Accessibility / WAI-ARIA
 - Search Engine Optimization (aka SEO)
 - Interaction or User Interface Design
 - User Experience
 - Usability
 - E-commerce Systems
 - Portal Systems
 - Wireframing
 - CSS Layout / Grids
 - DOM Manipulation (e.g., jQuery)
 - Mobile Web Performance
 - Load Testing
 - Performance Testing
 - Progressive Enhancement / Graceful Degradation
 - Version Control (e.g., GIT)
 - MVC / MVVM / MV*
 - Functional Programming
 - Data Formats (e.g., JSON, XML)
 - Data APIs (e.g. Restful API)
 - Web Font Embedding
 - Scalable Vector Graphics (aka SVG)

5. BACKEND DEVELOPERS

In the computer world, the "backend" refers to any part of a website or software program that users do not see. It contrasts with the frontend, which refers to a program's or website's user interface. In programming terminology, the backend is the "data access layer," while the frontend is the "presentation layer."

Most modern websites are dynamic, meaning webpage content is generated on-the-fly. A dynamic page contains one or more scripts that run on the web server each time the page is accessed. These scripts generate the content of the page, which is sent to the user's web browser. Everything that happens before the page is displayed in a web browser is part of the backend.

Examples of backend processes include:

- processing an incoming webpage request
- running a script (PHP, ASP, JSP, etc.) to generate HTML
- accessing data, such as an article, from a database using an SQL queries
- storing or updating records in a database
- encrypting and decrypting data
- handling file uploads and downloads
- processing user input via JavaScript

All of the examples above, besides the last one, are server-side processes that run on the web server. JavaScript is a client-side process, meaning it runs in the web browser. JavaScript may be considered a backend or a frontend process, depending on if the code affects the user interface or not.

The backend and frontend work together to create the full user experience. Data generated in the backend is passed to the frontend and presented to the user. While some organizations have separate backend and frontend development teams, the line between the two layers is rarely black and white. Therefore, many developers write code for both the backend and frontend. This is known as full-stack development.

A back-end web developer is responsible for server-side web application logic and integration of the work front- end web developers do. Back-end developers usually write web services and APIs used by front-end developers and mobile application developers.

Writing a good job description and a job ad for a back-end developer requires proper separation of concerns. Posting a generic web developer description in your job ad, when you are looking for an advanced back-end web developer, will bring numerous applications from people who are specialized in building front-end user interface, or web designers who have absolutely no knowledge about programming.

There are technologies and knowledge that are common to all web developers, and some that are specific to back-end developers. This article will provide you with a sample back-end web developer job description that will help you write a perfect job ad and assure that you easily find and hire the person that matches your specific criteria.

Back-end development means working on server-side software, which focuses on everything you can't see on a website. Back-end developers ensure the website performs correctly, focusing on

databases, back-end logic, application programming interface (APIs), architecture, and servers. They use code that helps browsers communicate with databases, store, understand, and delete data.

On a team, back-end developers collaborate with front-end developers, product managers, principal architects, and website testers to build the structure of a website or mobile app. Back-end developers must be familiar with many kinds of tools and frameworks, including languages such as Python, Java, and Ruby. They make sure the back-end performs quickly and responsively to front-end user requests.

Back-end developer vs. back-end engineer

Back-end developers and engineers have similar technical skills, but different responsibilities. A back-end developer is like a teacher, whereas a back-end engineer is like a school principal who creates the big-picture strategy. Back-end developers tend to focus on smaller parts of the whole, such as applications and programs, working with teams to prove their design methods are viable.

Back-end developer tasks and responsibilities

Back-end developers are required to have technical expertise, analytical thinking, and excellent collaboration skills. As a back-end web developer, you should be able to work independently to design the web infrastructure.

5.1. Programming Languages

5.1.1. PHP

PHP, an acronym for Hypertext Preprocessor, is a server-side scripting language and is one of the most widely used programming languages for backend web development. We can validate this statement by letting you know that platforms like Wikipedia, WordPress, Facebook, and many others are relying on PHP. This particular language is preferred for web development because of various prominent reasons such as cross-platform compatibility, OOPs features, easy integration with HTML, CSS, JavaScript, etc, huge community support, better flexibility & security, and many more. In addition, the language is quite easy to learn and use. Furthermore, there are various renowned PHP frameworks out there such as Laravel, Symfony, CodeIgniter, etc. that you can consider.

5.1.2. Python

Another language that is being preferred for backend development, specifically in current times, is Python. It is a high-level, general-purpose programming language that supports multiple programming paradigms such as Object-Oriented, Procedural, and Functional. Instagram, Spotify, Google, etc. are some of the popular platforms that are using Python in their tech stacks. The language provides you with some remarkable features such as rich library support, easy integration with other languages, GUI Programming support, compatibility with trending technologies, etc. Also, Python has a very simple syntax and comes up with better code readability aspects that subsequently make it easy to learn and use. Several popular Python web frameworks to you can take into consideration are Django, Flask, etc.

5.1.3. JavaScript

Honestly, JavaScript needs no introduction for its extensive role in web development. It is actually one of the fundamental units of web development alongside HTML and CSS. The language is being preferred by a huge number of developers for web development, and it is ranking at the top position as well at various renowned indices for top programming languages. Apart from web development, the language can also be used for game development, mobile application development, etc. Some of

the considerable features of this particular language are – Imperative & Structured, Light-Weight, Object-oriented Programming Support, Platform Independent, and many others. Also, the best part is that JavaScript can be used for both – Frontend Development and Backend Development.

Other languages that you can consider: Java, Go, Ruby, etc.

5.2. Frameworks

5.2.1. Laravel

Laravel is one of the most popular PHP frameworks available in the tech world. This open-source web framework follows the Model-View-Controller (MVC) architectural pattern and provides you with numerous enriching features like a built-in command-line tool called Artisan, pre-installed Object-Oriented and Modular libraries, Eloquent ORM (Object Relational Mapping), Template Engine, and many others. Moreover, the framework leverages you with various tools for particular tasks like dependency injection, unit testing, etc. The ready-to-use packages provided by Laravel include Cashier for managing subscription billing services, Envoy that provides a clean & minimal syntax for defining common tasks you run on your remote servers, Socialite for the simplified mechanism for OAuth authentication with providers like Facebook, GitHub, Google, etc. and various others.

5.2.2. Django

Django is a Python-based open-source web framework that allows you to do web development more efficiently and without any hassle. Django follows the model-template-views (MTV) architectural pattern. The reason behind the immense popularity and demand for this particular framework is some of its noticeable features such as extensibility, rapid development, scalability, security, vast community, and many more. Businesses are using Django for various distinct web development areas such as social networking platforms, scientific computing platforms, content management systems, and various others. Some of the popular websites that are using Django are – Instagram, Mozilla, Pinterest, etc. If you want to get into web development using Python – you can surely give it a try to Django.

Other frameworks that you can consider: Angular, Meteor, Spring, Ruby on Rails, etc.

5.3. Databases

5.3.1. MongoDB

MongoDB is a free and open-source document-oriented database that is very much popular among web developers. This NoSQL database uses JSON-like documents with optional schemas for storage & retrieval of data and comes up with much-needed scalability and flexibility. MongoDB provides you with numerous prominent features such as Ad-hoc queries, Indexing, Replication, Load Balancing, File Storage, Aggregation, Transactions, etc. In addition, the security and data recovery aspects of this particular database are quite better compared to various other database management systems. Other than that, MongoDB is compatible with various standard programming languages like C/C++, Java, PHP, Python, Ruby, and several others.

5.3.2. MySQL

MySQL is another open-source relational database management system that is widely used for web-based applications. It is a fast and high-performance database that provides better scalability, usability, and reliability. Also, MySQL provides cross-platform compatibility, strong indexing support, SSL support for secured connections, powerful data encryption and accuracy, built-in replication support, and various other features.

Meanwhile, MySQL can work on various distinct operating systems and is compatible with many popular languages like PHP, Java, etc. Let us tell you this as well that this particular framework, MySQL, is used by various renowned websites like Flickr, Twitter, Facebook, Drupal, Joomla, and many others.

Other databases that you can consider: Oracle, PostgreSQL, etc.

5.4. Web Server

5.4.1. Apache

The Apache, or you can say Apache HTTP Server, is an open-source cross-platform web server that was developed by the Apache Software Foundation. It is one of the leading web servers, and you can understand it with the fact that approximately half of all the websites across the world are powered by Apache. There are numerous worthwhile features that come up with Apache like Loadable Dynamic Modules, Multiple Request Processing modes, CGI support, User and Session tracking, Real-time status views, XML & FTP support, and many others. Also, the Apache Web Server is very much compatible with almost all operating systems such as Linux, macOS, Windows, etc.

5.4.2. NGINX

NGINX is another open-source web server that is also being used for reverse proxying, load balancing, caching, mail proxying, and more. It uses an asynchronous event-driven approach to handle requests and subsequently its modular event-driven architecture provides more expected performance under high loads. Some of the enriching features of the NGINX web server are – handling static files & auto-indexing, IPv6-compatible, URL rewriting & redirection, etc. Other than that, various additional features such as advanced load balancing, session persistence based on cookies, Cache Purging API, etc. are included in NGINX Plus. Dropbox, Netflix, WordPress.com, etc. are some of those popular platforms that are using NGINX.

Other web servers that you can consider: Lighttpd, Microsoft IIS, etc.

Other Tools

Docker

Postman

Jira

6. FULL STACK DEVELOPERS

A full-stack developer is a developer or engineer who can build both the front end and the back end of a website. The front end (the parts of a website a user sees and interacts with) and the back end (the behind-the-scenes data storage and processing) require different skill sets. Since full-stack developers are involved with all aspects of the development process, they must have expertise in both.

A full-stack developer can work in-house or at a computer development company that engineers websites, software, and other components for other businesses.

What does a full-stack developer do?

Full-stack developers design and create websites and applications for various platforms. A full-stack developer's job description might include the following:

- Develop and maintain web services and interfaces
- Contribute to front-end and back-end development processes
- Build new product features or APIs
- Perform tests, troubleshoot software, and fix bugs
- Collaborate with other departments on projects and sprints

The world of full-stack development is large, and many new and evolving technologies continually push the limits of what a full-stack developer can create. Staying on top of cutting-edge technology and techniques in the full-stack development field is one of the many exciting aspects of working in this role.

How to become a full-stack developer

Consider earning a degree.

72% of software developers hold a bachelor's degree, although it's not a strict requirement. Full-stack developers typically study computer science, computer engineering, or a related field. A degree can hone the technical and workplace skills necessary to be an effective full-stack engineer.

Research entry-level roles.

There are various levels of full-stack development jobs. Search for junior or entry-level positions on job sites to understand what skills employers are looking for in applicants.

Develop your coding skills.

To be an effective full-stack developer, you should acquire and continue upgrading your knowledge of applicable programming language skills. Consider taking courses in relevant languages like Python, HTML, CSS, and JavaScript.

7. WEB DEVELOPMENT TOOLS

Web development tools have come a long way in just a few short years. Thanks to this, we can harness the power of highly tested libraries to improve our workflow and benefit from greater possibilities when it comes to responsive design

Not only that, we can build things together thanks to ever-improving version control systems.

From browser add-ons and plugins, to processors that streamline your code, there have never been more possibilities for creating awesome web applications.

But with the number of web dev tools increasing almost daily, finding the best software to get the job done can sometimes feel daunting.

To help you out, we've created a list of essential tools for frontend development to get you started. If you're interested in finding out about one in particular, simply select it from the list below.

- Sublime Text
- Chrome Developer Tools
- jQuery
- GitHub
- CodePen
- Angular
- Sass

7.1. Sublime Text

Let's start with the basics: a first-rate code editor—one that features a well-designed, super-efficient, and ultra speedy user interface. There are several that do this well, but arguably the best (and most popular) is Sublime Text.

Artfully run by a one-man development team, the secret to Sublime's success lies in the program's vast array of keyboard shortcuts—such as the ability to perform simultaneous editing (making the same interactive changes to multiple selected areas) as well as quick navigation to files, symbols, and lines. And when you're spending 8+ hours with your editor each day, those precious few seconds saved for each process really do add up.

7.2. Chrome Developer Tools

Wouldn't it be great if you could edit your HTML and CSS in real-time, or debug your JavaScript, all while viewing a thorough performance analysis of your website?

Google's built-in Chrome Developer Tools let you do just that. Bundled and available in both Chrome and Safari, they allow developers access into the internals of their web application. On top of this, a palette of network tools can help optimize your loading flows, while a timeline gives you a deeper understanding of what the browser is doing at any given moment.

7.3. jQuery

JavaScript has long been considered an essential frontend language by developers, although it's not without its problems: riddled with browser inconsistencies, its somewhat complicated and unapproachable syntax meant that functionality often suffered.

That was until 2006, when jQuery—a fast, small, cross-platform JavaScript library aimed at simplifying the frontend process—appeared on the scene. By abstracting a lot of the functionality usually left for developers to solve on their own, jQuery allowed greater scope for creating animations, adding plug-ins, or even just navigating documents.

7.4. GitHub

It's every developer's worst nightmare—you're working on a new project feature and you screw up. Enter version control systems (VCS)—and more specifically, GitHub.

By rolling out your project with the service, you can view any changes you've made or even go back to your previous state (making pesky mistakes a thing of the past). There are so many reasons why GitHub is vital to developers. The repository hosting service also boasts a rich open-source development community (making collaboration between teams as easy as pie), as well as providing several other components such as bug tracking, feature requests, task management, and wikis for every project.

7.5. CodePen

Despite being around since 2012, the ever-increasing of people learning programming means that 2023 is going to be another bumper year for this tool beloved by the frontend community. There is almost no better way of showcasing your HTML, CSS, and JavaScript snippets, and as a result their embeds are an increasingly common sight across coding resources online.

7.6. Angular

HTML is usually the cornerstone of any frontend developer's toolbox, but it has what many perceive to be a serious flaw: it wasn't designed to manage dynamic views.

This is where AngularJS, an open-source web framework, came in. Developed by Google, AngularJS lets you extend your application's HTML syntax, resulting in a more expressive, readable, and quick to develop environment that could otherwise not have been built with HTML alone.

7.7. Sass

Web dev tools that save time are your best friend, and one of the first things you'll learn about code is that it needs to be DRY ("Don't Repeat Yourself"). The second thing you'll probably learn is that CSS is usually not very DRY.

Enter the world of the CSS preprocessor, a tool that will help you write maintainable, future-proof code, all while reducing the amount of CSS you have to write (keeping it DRY).

8. WEB SECURITY

Web security in development refers to the practices and measures taken during the creation and maintenance of websites and web applications to protect them from various threats and vulnerabilities. It is a critical aspect of the software development lifecycle, as the internet is rife with potential risks such as data breaches, cyberattacks, and unauthorized access. Here are some key aspects of web security in development:

- 8.1. Authentication and Authorization:** Implement strong user authentication and authorization mechanisms to ensure that only authorized users have access to specific parts of a website or application. This includes enforcing password policies, using multi-factor authentication, and defining role-based access control.
- 8.2. Secure Coding Practices:** Developers should follow secure coding practices to write code that is resilient to common security vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Regular code reviews and security testing can help identify and mitigate these issues.
- 8.3. Data Encryption:** Sensitive data, both in transit and at rest, should be encrypted to prevent eavesdropping and unauthorized access. Transport Layer Security (TLS) should be used to secure data in transit, while data at rest should be protected using encryption algorithms.
- 8.4. Input Validation:** Validate and sanitize user inputs to prevent malicious input from being executed as code. Input validation helps protect against SQL injection, XSS, and other injection attacks.
- 8.5. Patch Management:** Keep all software components, including web servers, frameworks, and third party libraries, up to date with the latest security patches. Vulnerabilities in outdated software can be exploited by attackers.
- 8.6. Security Headers:** Utilize security headers like Content Security Policy (CSP), Cross-Origin Resource Sharing (CORS), and HTTP Strict Transport Security (HSTS) to control and enhance the security of web applications.
- 8.7. Session Management:** Implement secure session management techniques, such as using unique session tokens, setting session timeouts, and properly handling session data to prevent session fixation and session hijacking.
- 8.8. Error Handling:** Customize error messages to avoid revealing sensitive information about the application's structure or underlying technologies to potential attackers.
- 8.9. Security Testing:** Regularly conduct security testing, including vulnerability scanning, penetration testing, and code reviews, to identify and remediate vulnerabilities in the application.

9. RESPONSIVE WEB DESIGN ABILITY

9.1. Responsive Web Design:

Responsive web design is an approach to web development that aims to create websites and web applications that adapt and display optimally on various devices and screen sizes. This design philosophy

recognizes that users access the internet from a wide range of devices, including desktop computers, laptops, smartphones, and tablets, each with different screen dimensions and orientations.

9.2. Key aspects of responsive web design include:

9.2.1. Flexible Layouts: Using fluid grids and CSS media queries to ensure that content and design elements adjust proportionally to the screen size.

9.2.2. Flexible Images: Employing techniques like CSS max-width to ensure that images scale appropriately without distortion.

9.2.3. Media Queries: Using CSS media queries to apply different styles or layouts based on the device's screen size or other characteristics.

9.2.4. Mobile-First Approach: Starting the design process with mobile devices in mind and progressively enhancing the user experience for larger screens. Responsive web design enhances user experience by ensuring that websites are visually appealing and functional regardless of the device being used, ultimately leading to higher user engagement and satisfaction.

9.3. Accessibility in Development:

Accessibility in web development refers to the practice of creating digital content and interfaces that can be used and understood by people with a wide range of abilities and disabilities. It aims to make the web inclusive for everyone, including individuals with visual, auditory, motor, cognitive, or other impairments.

Key considerations for accessibility in development include:

9.3.1. Semantic HTML: Using proper HTML elements and markup to ensure the structure and content of a web page are clear and meaningful, aiding screen readers and other assistive technologies.

9.3.2. Keyboard Navigation: Ensuring that all interactive elements and functions on a website can be accessed and operated using a keyboard alone.

9.3.3. Alt Text for Images: Providing descriptive alternative text for images to convey their content to users who cannot see them.

9.3.4. Proper Color Contrast: Ensuring sufficient contrast between text and background colors to make content readable for individuals with visual impairments.

9.3.5. Accessible Forms: Creating forms that are easy to navigate and complete, with clear labels and error messages.

9.3.6. Testing with Assistive Technologies: Conducting usability testing with screen readers and other assistive technologies to identify and address accessibility issues.

10. GALLERY

10.1. Birajee Travel

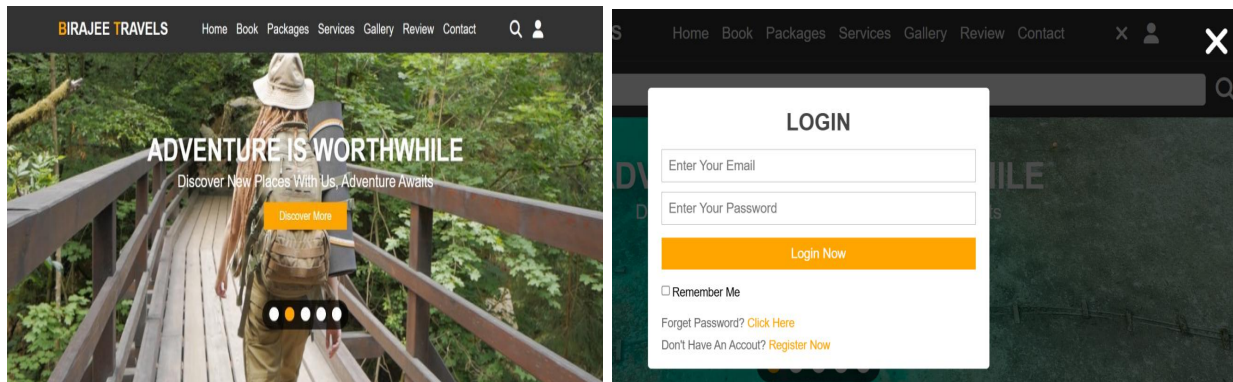


Fig: - Home page of Birajee Travels

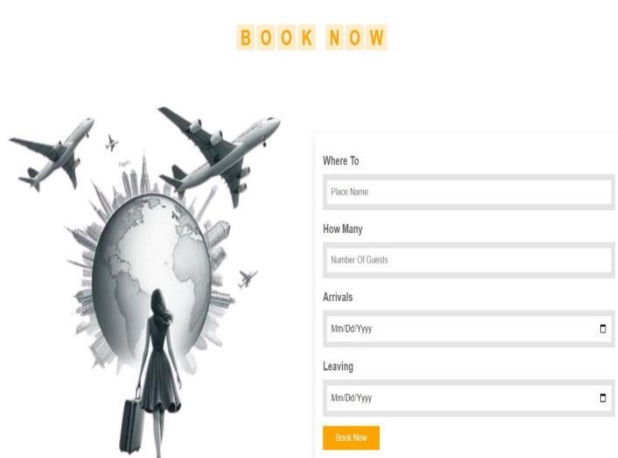


Fig: - Book Now page of Birajee Travel

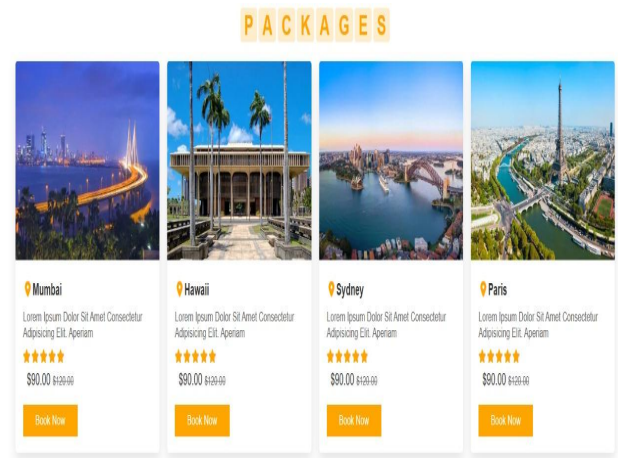


Fig: - Packages of Birajee Travels

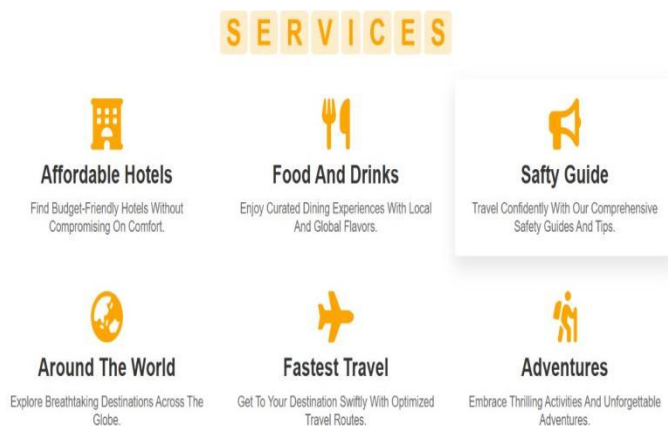


Fig: - Services of Birajee Travels

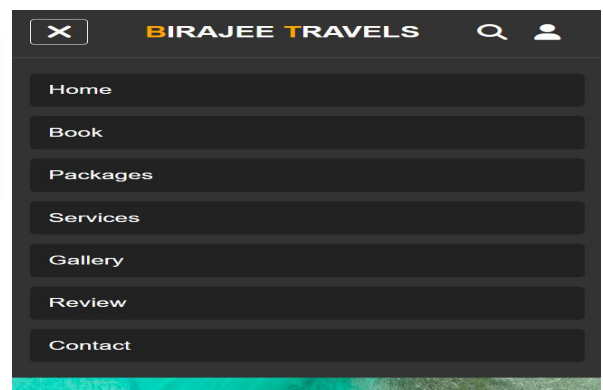


Fig: - Mobile view of Birajee Travels

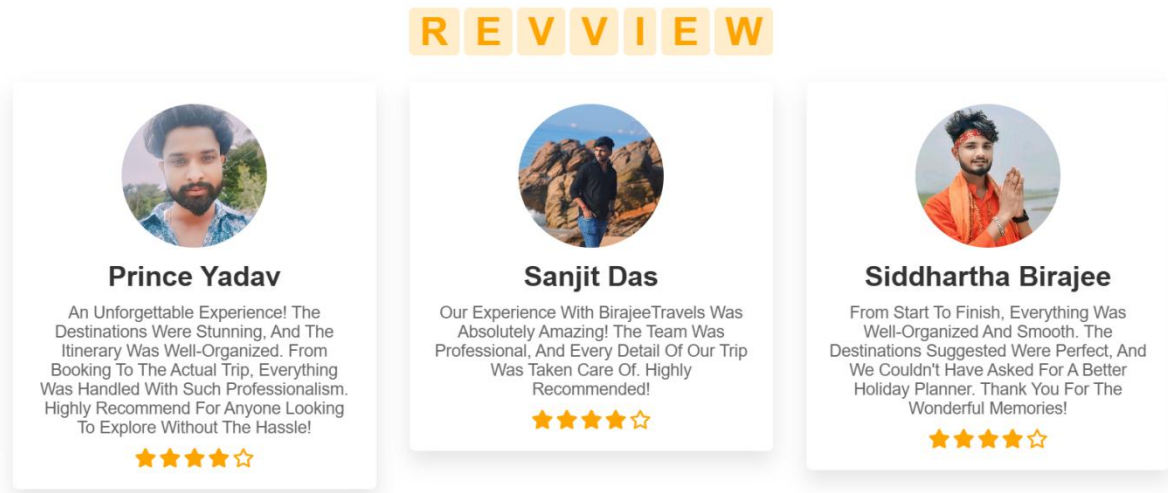


Fig: - Reviews of Birajee Travels

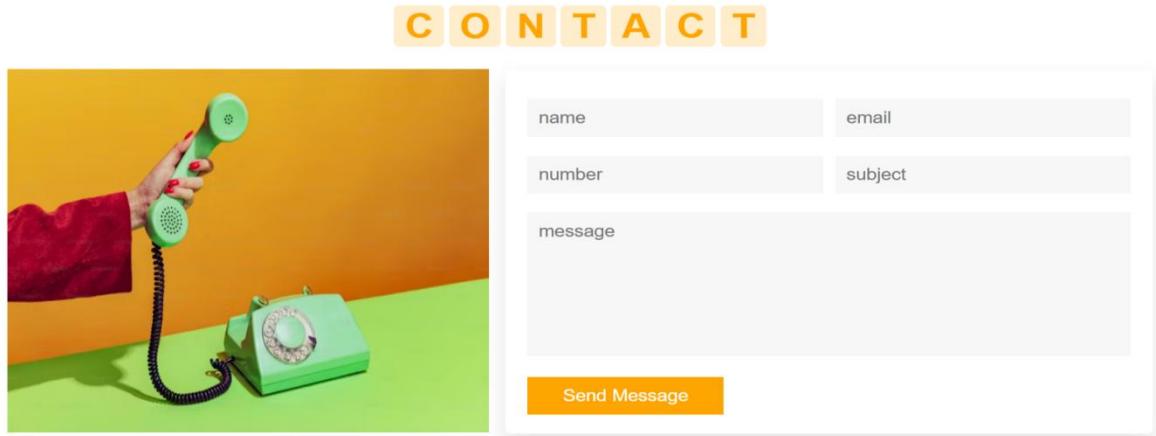


Fig: - Contact of Birajee Travels

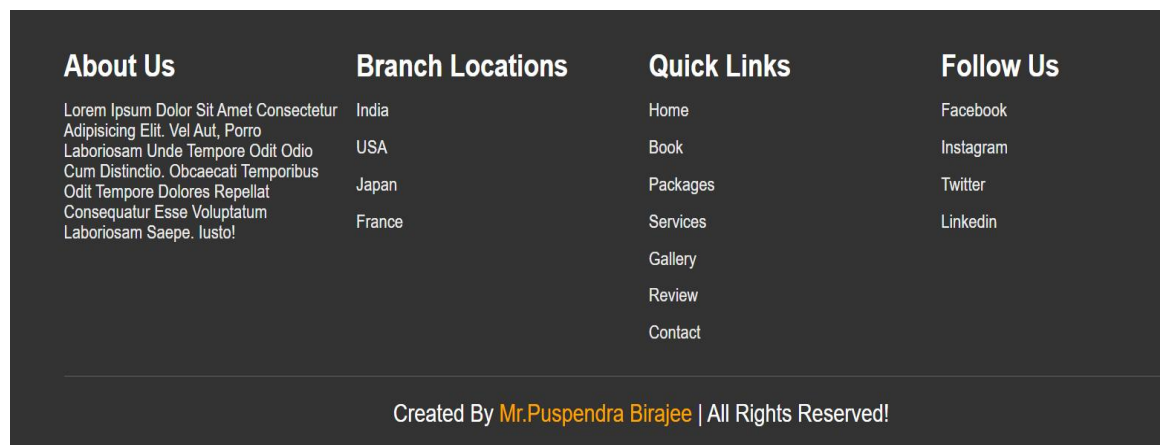


Fig: - Footer page of Birajee Travels

11.SUMMARY

This report is about my 2 Months internship program with Brain o vision. In this comprehensive report, I have discussed every major aspect of the company which I observed and perceived during my internship program.

During my internship program, I have learned and mainly worked on Web Development. All the details have been discussed in detail. All the policies and procedures of the company have been discussed in detail.

The main purpose of the internship is to learn by working in practical environment and to apply the knowledge acquired during the studies in real world scenario to tackle the problems using the knowledge and skill learned during the academic process.

12.ABOUT THE COMPANY

Brain o vision, one of the leading and fast-growing companies in the area of Software Development, Web Designing and all the Digital Solutions to the corporate world and Nation. We are a global IT consulting company with in-depth expertise in providing end-to-end solutions. We work on diverse projects ranging from simple information systems and websites to complex enterprise type architectures, desktop or web-enabled applications.

Brain o vision Solutions India Private Limited. is an organization started in 2014 which serves web solutions, software development and tech education in the corporate field. This is the place for brands, students, and faculty to find the best solution.

12.1. Vision:

Our vision is to be an accessible platform for Technical Solutions to our clients. Helping them to integrate critical applications and implement projects that achieve business objectives and improve performance. We aim to make our services widely available and useful for our clients

12.2. Mission:

Along with software development and solutions our aim is to fill the gap between academics and corporate. Involve every student in the real-time projects and make them corporate ready. Provide an opportunity to engineering students and make them gain exposure to the diverse field.

13. OPPORTUNITIES

During these two months of the internship, I was given the opportunity to perform the following role:

Intern:

- a. Coordinating with the team members and team leads on a regular basis to keep a track of the activities like the meetings held and about the work to be done.
- b. I learned about developing the applications using different tools.
- c. For that I have referred the GitHub repositories related to gain the complete knowledge on that.
- d. Then I have gathered the requirements.
- e. They also provide us the opportunity to voluntarily interact in other projects as well.
- f. They have given different tasks to develop different parts of the application.
- g. Also, they have finally conducted some tests to certify with the completion of internship.

14. TRAINING

In these two months of the training, they have provided us the training in Full Stack using different tools.

They have provided us with the training of several technologies like:

- HTML
- CSS
- JavaScript
- Java

HTML: When you are browsing the web, the pages you are viewing are just text documents sitting on someone else's computer. The text in a typical web page is wrapped in HTML tags, which tell your browser about the structure of the document. With this information, the browser can decide how to display the information in a way that makes sense.

CSS: Cascading Style Sheets, commonly referred to as CSS, is a fundamental technology used in web development to control the presentation and layout of web pages. It allows developers to define the visual styling, positioning, and formatting of HTML elements within a web page. CSS plays a crucial role in creating visually appealing and user-friendly websites.

JAVASCRIPT: JavaScript is a high-level, versatile, and widely used programming language that is primarily known for its role in web development. It allows developers to add interactivity, manipulate web page content, and create dynamic user experiences in web applications. JavaScript is supported by all modern web browsers, making it an essential language for front-end and, increasingly, back-end web development.

JAVA: Java is a multi-platform, object-oriented, and network-centric language that can be used as a platform in itself. It is a fast, secure, reliable programming language for coding everything from mobile apps and enterprise software to big data applications and server-side technologies.

15. CHALLENGES FACED

- At the beginning of internship, I faced difficulty for understanding the applications and different tools.
- I faced difficulty in installing the software and setup of environment.
- I faced difficulty in managing college and internship timings.
- I faced difficulty in understanding the advanced topics in development.
- I faced difficulty in managing the memory in pc.
- Even with these difficulties, I am able to complete the internship and I hope so that it helps me in securing a new job.

16. CONCLUSION

Concluding an internship can be a significant milestone in one's professional development. Here's a summary or conclusion regarding the experience of completing an internship:

- **Learning and Growth:** The internship provided an invaluable opportunity to learn and grow both personally and professionally. Through hands-on experience, I gained practical skills, expanded my knowledge, and developed a deeper understanding of my chosen field.
- **Real-World Application:** The internship allowed me to apply classroom knowledge to real-world scenarios, bridging the gap between theory and practice. I encountered challenges and problem-solving opportunities that helped me develop a more practical perspective on the industry.
- **Skill Development:** Over the course of the internship, I had the chance to develop and refine a range of skills, including technical skills specific to the field, as well as soft skills like communication, teamwork, and time management.
- **Professional Network:** I had the privilege of working alongside experienced professionals in the industry, which allowed me to build a valuable network of contacts. These connections can be instrumental in future career opportunities and collaborations.
- **Mentorship and Guidance:** I benefited from the mentorship and guidance of seasoned professionals who provided feedback, shared their expertise, and offered valuable insights. This guidance was instrumental in my development.
- **Contributions to Projects:** Throughout the internship, I actively contributed to meaningful projects, making a tangible impact on the organization. These contributions not only added to my portfolio but also gave me a sense of accomplishment.
- **Adaptability:** The internship environment often presented dynamic challenges and changes, forcing me to adapt and think on my feet. This adaptability is a crucial skill that will serve me well in future endeavours.
- **Self-Reflection:** The internship allowed me to reflect on my career goals, interests, and strengths. It helped me gain clarity about my professional path and the direction I want to take in my career.
- **Resume Enhancement:** Completing this internship has added a valuable entry to my resume, demonstrating to future employers my commitment to continuous learning and professional development.
- **Gratitude:** I am grateful for the opportunity to be part of the organization and for the trust placed in me as an intern. I want to express my appreciation to the entire team for their support and mentorship during this period.