Index

CHAPTER - 1

Introduction

The Internet

The Internet is the global system of interconnected computer networks that use the Internet protocol suite (TCP/IP) to link devices worldwide. It is a network of networks that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies.

Internet was started as a project of US Military project named ARPANET directed by Robert Taylor and managed by Lawrence Roberts.

WWW

The World Wide Web (abbreviated WWW or the Web) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and accessible via the Internet.

Now, to use these services, we need some softwares which can build things and hence helping our work done on the web. Web technologies which we will be covering to build those things are HTML, CSS, Javascript, Node.js, Express.js etc. The primary focus for back-end will be on the Node.js which powers a huge amount of devices websites. Hence, this plays a vital role in the field of web development especially because of its non-blocking IO feature which is discussed whenever people talk about node.js.

Web programming, also known as web development, is the creation of dynamic web applications. Examples of web applications are social networking sites like Facebook or e-commerce sites like Amazon.

The good news is that learning web development is not that hard!

In fact, many argue it's the best form of coding for beginners to learn. It's easy to set up, you get instant results and there's plenty of online training available.

A lot of people learn web coding because they want to create the next Facebook or find a job in the industry. But it's also a good choice if you just want a general introduction to coding, since it's super easy to get started. No matter whether you're looking for a career or just want to learn coding, learning how to develop for the web is for you. It's one of the smartest decisions you will ever make!

"Full Stack" Developers marry both sides into one. A Full Stack developer can comfortably work with both the front and back ends. This is what we focus on building you up to here.

things scale. Back End Developers make use of programming languages like Java, Python, and Ruby (among many others) to work with data.

"Full Stack" Developers marry both sides into one. A Full Stack developer can comfortably work with both the front and back ends. This is what we focus on building you up to here.

Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardised code and proprietary software; user experience design; and search engine optimization.

Types of Web Developers

Earlier, we mentioned that the work could be in the front end, the back end, or full stack. What exactly are these?

The “Front End” is the stuff you see on the website in your browser, including the presentation of content and the user interface elements like the navigation bar. Front End Developers make use of HTML, CSS, Javascript, and their relevant frameworks to ensure that content is presented effectively and that users have an excellent experience.

The “Back End” refers to the guts of the application, which live on the server. They manipulate data appropriately to make sure the Front End has what it needs. This can become very complicated as things scale. Back End Developers make use of programming languages like Java, Python, and Ruby (among many others) to work with data.

“Full Stack” Developers marry both sides into one. A Full Stack developer can comfortably work with both the front and back ends. This is what we focus on building you up to here.

CHAPTER - 2

Any digital project, for example, a website, an android application etc. at the root level is divided Into two blocks:

1. Front-end
2. Back-end

Front-end

These are the two divisions of the project to help the creator develop the project smoothly. This division help working different people work upon the things they are master in. Thus the whole load of the project is balanced.

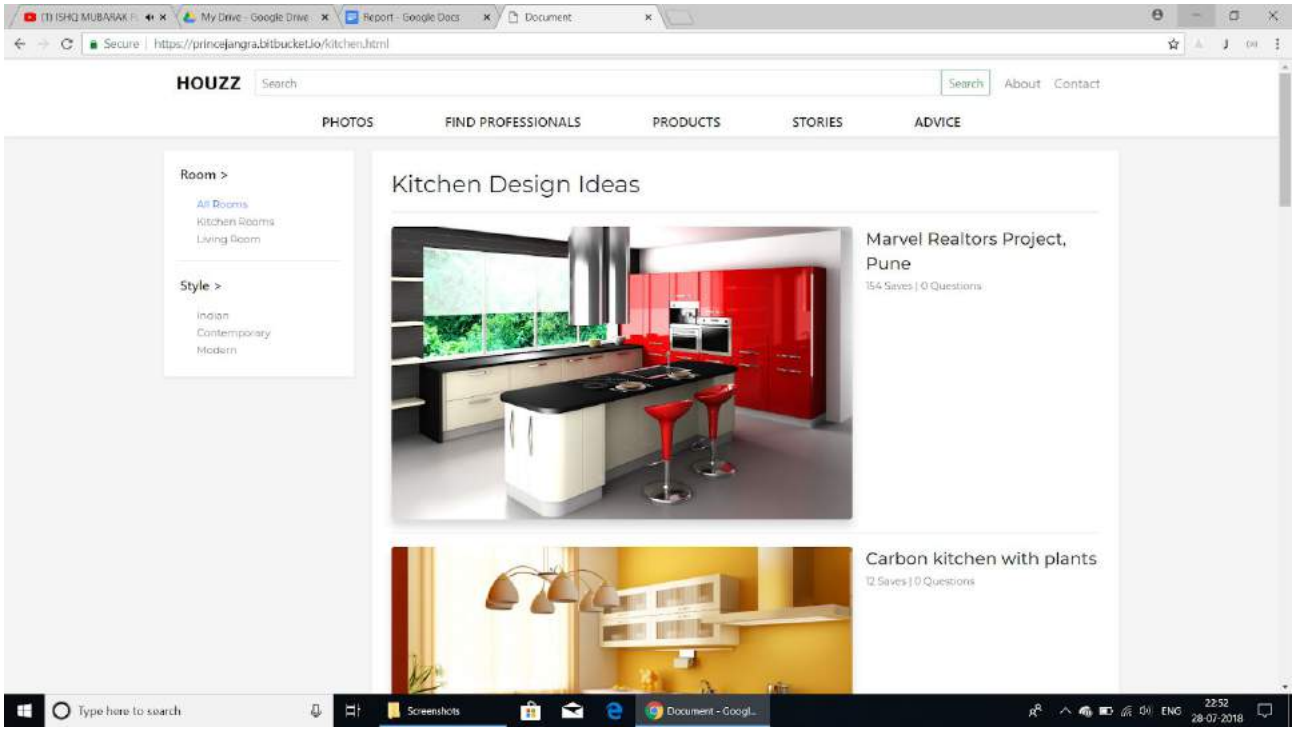
Front-end covers the part of the project which is visible to the user, i.e., it deals with the client side. Anything happening on the user side of the connection can be received or manipulated by the user. It concerns mostly with the user interface and user experience of the website. How the website is presented to the user is the primary goal of the front-end. Simplicity, accessibility, proper user experience, clarity of the actions and feedback are some of the basic features which play a vital role in the best possible front-end.

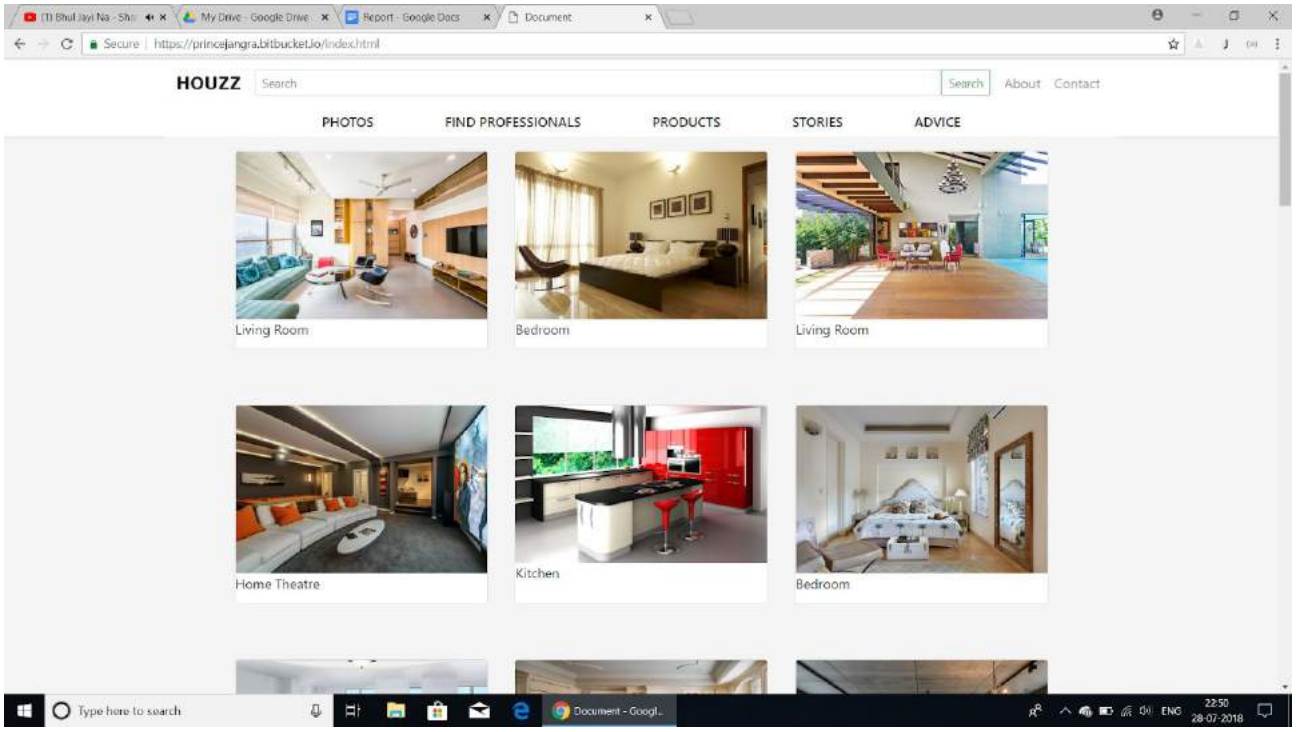
HTML is a markup language which is used for defining the structure of the website.

These are the basic things to create front-end of any website. While there are many things to learn afterwards and to use them for a much highly sophisticated front-end of a website.

What is Front-End Web Development?

Front-end web development is everything involved in programming the user interface of a web application. Typically it refers to the Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript portion of web site production as opposed to the database or server-side programming. It encompasses everything from building a simple page of HTML text to creating complex, responsive HTML5 websites designed to be accessed via various different browsers, devices and screen sizes.


HOUZZ
Search
About
Contact
PHOTOS
FIND PROFESSIONALS
PRODUCTS
STORIES
ADVICE
Room >
All Rooms
Kitchen Rooms
Living Room
Style >
Indian
Contemporary
Modern
Kitchen Design Ideas
Marvel Realtors Project, Pune
154 Saves | 0 Questions
Carbon kitchen with plants
12 Saves | 0 Questions


HOUZZ
Search
About
Contact
PHOTOS
FIND PROFESSIONALS
PRODUCTS
STORIES
ADVICE
Living Room
Bedroom
Living Room
Home Theatre
Kitchen
Bedroom

CHAPTER - 3

Back-end

Back-end is the part of the website which deals with the core functioning of the website and is hidden to the user for user's safety. User shouldn't know what is happening on the website, this is the concern of the back-end developers. Having back-end makes the website more dynamic.

When users interact with the website which involves back-end, it makes the creators easy to involve with users for the main purpose of the website. Back-end involves maintaining the database of various users, helping them to get things done through the various tools and services developed by the programmers of the back-end. Common objectives of the back-end are to involve users with the website, maintaining the proper database for various users.

BACKEND DEVELOPMENT

The backend of a web application is an enabler for a frontend experience. An application's frontend may be the most beautifully crafted web page, but if the application itself doesn't work, the application will be a failure. The backend of an application is responsible for things like calculations, business logic, database interactions, and performance. Most of the code that is required to make an application work will be done on the backend. Backend code is run on the server, as opposed to the client. This means that backend developers not only need to understand programming languages and databases, but they must have an understanding of server architecture as well. If an application is slow, crashes often, or constantly throws errors at users, it's likely because of backend problems.

Backend development is not all ones and zeros though. Much like frontend development, backend development has a human aspect to it as well. Since most of the code for an application is written on the backend, it should be easy to understand and work with. Most

backend languages – like Ruby and Python – have standardized styles and idioms that make reading and writing code more efficient and enjoyable.

What Do Back-End Developers Do?

What back-end developers do can vary greatly depending on the size and the scope of the application they are working on. I've held many jobs where I was a back-end developer, working on the business logic in an application, and feeding and retrieving data from the front-end.

In the web development world, most back-end developers concern themselves with building the actual logic behind the application they are working on.

Often, front-end developers will build out a user interface and back-end developers will write code that makes it all work.

For example, a front-end developer might create a screen in an application with a button to press to get the customer's data.

A back-end developer might write the code that makes that button work by figuring out what data to fetch from the database for the appropriate customer and delivering it back to the front-end, where it is eventually displayed.

A back-end developer might also be heavily involved in the architecture of a system, deciding how to organize the logic of the system so that it can be maintained and run properly.

He might be involved in building frameworks or the architecture of a system to make it easier to program against. Back-end developers tend to spend much more time implementing algorithms and solving problems than front-end developers do.

I've always liked back-end development work because it feels like more of a challenge.

That's not to say that front-end developers don't ever solve difficult problems, but often front-end development work is more about creating user interfaces and hooking them up rather than implementing the actual business logic that makes the app work.

CHAPTER - 4

Node.js

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside the browser. According to official node.js website,

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

I/O refers to input/output. It can be anything ranging from reading/writing local files to making an HTTP request to an API. I/O takes time and hence blocks other functions.

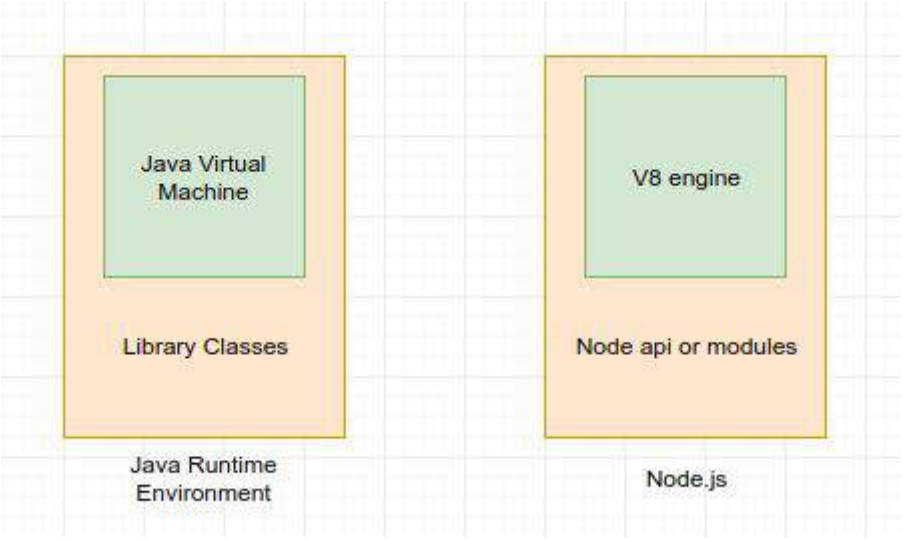


Fig. Node.js basic structure

History

Node.js was originally written by Ryan Dahl in 2009. He was inspired to create Node.js after seeing a file upload progress bar on Flickr. The browser did not know how much of the file had been uploaded and had to query the Web server. He desired an easier way. He demonstrated the project at the inaugural European JSConf on November 8, 2009. Node.js combined Google's V8 JavaScript engine, an event loop, and a low-level I/O API.

Why Node.js?

Node.js lets us use JavaScript language on the server, so it allows us to write JavaScript outside the browser which, till now, was used only for front-end things only.

Node.js operates on a single thread, using non-blocking I/O calls, allowing it to support tens of thousands of concurrent connections without incurring the cost of thread context switching. It uses asynchronous programming. A common task for a web server can be to open a file on the server and return the content to the client but servers should not be used for simple tasks when you can get them done without the help of servers.

Here is how PHP or ASP handles a file request:

1. Sends the task to the computer's file system.
2. Waits while the file system opens and reads the file.
3. Returns the content to the client.
4. Ready to handle the next request.

Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

About Node.js

As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection the callback is fired, but if there is no work to be done, Node will sleep.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Node.js eliminates the waiting, and simply continues with the next request. Node.js runs single-threaded, non-blocking, asynchronously, which is very memory efficient.

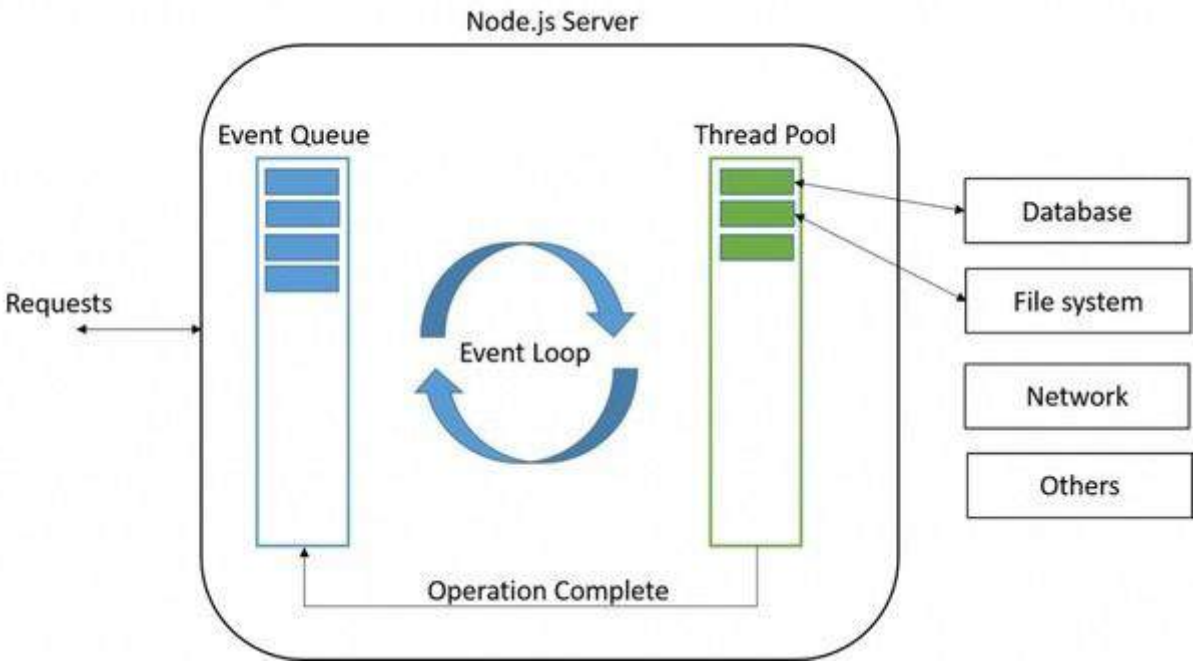


Fig. Working of node.js

CHAPTER - 5

EXPRESS js

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Performance

Express provides a thin layer of fundamental web application features, without obscuring Node.js features that you know and love.

Hello world example

```
const express = require('express' 4.16.3 )
const app = express()

app.get('/', (req, res) => res.send('Hello World!'))

app.listen(3000, () => console.log('Example app listening on port 3000!'))
```

Basic routing

Routing refers to determining how an application responds to a client request to a particular endpoint, which is a URI (or path) and a specific HTTP request method (GET, POST, and 4so on)

Each route can have one or more handler functions, which are executed when the route is matched.

Route definition takes the following structure:

```
app.METHOD(PATH, HANDLER)
```

Where:

app is an instance of express.

METHOD is an HTTP request method, in lowercase.

PATH is a path on the server.

HANDLER is the function executed when the route is matched.

Routing

Routing refers to how an application's endpoints (URIs) respond to client requests. For an introduction to routing, see Basic routing.

You define routing using methods of the Express app object that correspond to HTTP methods; for example, app.get() to handle GET requests and app.post to handle POST requests. For a full list, see app.METHOD. You can also use app.all() to handle all HTTP methods and app.use() to specify middleware as the callback function (See Using middleware for details).

These routing methods specify a callback function (sometimes called "handler functions") called when the application receives a request to the specified route (endpoint) and HTTP method. In other words, the application "listens" for requests that match the specified route(s) and method(s), and when it detects a match, it calls the specified callback function.

In fact, the routing methods can have more than one callback function as arguments. With multiple callback functions, it is important to provide next as an argument to the callback function and then call next() within the body of the function to hand off control to the next callback.

The following code is an example of a very basic route.

```
var express = require('express')
var app = express()

// respond with "hello world" when a GET request is made to the homepage
app.get('/', function (req, res) {
  res.send('hello world')
})
```

Route methods

A route method is derived from one of the HTTP methods, and is attached to an instance of the express class.

The following code is an example of routes that are defined for the GET and the POST methods to the root of the app.

```
// GET method route
app.get('/', function (req, res) {
  res.send('GET request to the homepage')
})

// POST method route
app.post('/', function (req, res) {
  res.send('POST request to the homepage')
})
```

SQL

SQL is a standard language for accessing and manipulating databases.

What is SQL?

SQL stands for Structured Query Language

SQL lets you access and manipulate databases

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987

What Can SQL do?

SQL can execute queries against a database

SQL can retrieve data from a database

SQL can insert records in a database

SQL can update records in a database

SQL can delete records from a database

SQL can create new databases

SQL can create new tables in a database

SQL can create stored procedures in a database

SQL can create views in a database

SQL can set permissions on tables, procedures, and views

List of SQL Commands

BACKGROUND

SQL, Structured Query Language, is a programming language designed to manage data stored in relational databases. SQL operates through simple, declarative statements. This keeps data accurate and secure, and it helps maintain the integrity of databases, regardless of size

COMMANDS

ALTER TABLE

ALTER TABLE table_name

ADD column_name datatype;

ALTER TABLE lets you add columns to a table in a database.

AND

SELECT column_name(s)

FROM table_name

WHERE column_1 = value_1

AND column_2 = value_2;

AND is an operator that combines two conditions. Both conditions must be true for the row to be included in the result set.

CREATE TABLE

```
CREATE TABLE table_name (
  column_1 datatype,
  column_2 datatype,
  column_3 datatype
);
```

CREATE TABLE creates a new table in the database. It allows you to specify the name of the table and the name of each column in the table.

INSERT

```
INSERT INTO table_name (column_1, column_2, column_3)
VALUES (value_1, 'value_2', value_3);
```

INSERT statements are used to add a new row to a table.

CHAPTER - 6

Project:

After learning many of the things in web development area, it's time to build something upon the learnt things to show our skills and learn them more. So, we made a project using the things we learnt so far. Our project is focused on the development of a common platform which will host the various samples of the beautifully designed and constructed architectures like interiors of houses. Many times, we face a situation when we construct or modify our home, how do we design the interior of this new component of our home or in what way should I instruct the home's bedroom, living room etc. Thus, here we are, with the idea of simplifying this thought and getting down to the solution of all those confusing questions. A user can look into the various samples provided by the other users of the website who have built their houses with the help of professional interior designers and architects. Looking into these ideas, a user gets little help with the idea of his dream design for his home. Hence, this website is primarily focused on the showcasing various designs to get an idea of how we will be building our homes.

There will also be a help section to help people with their confusions about various designs. Thus, it is also a way of connecting variety of skilled and unskilled people.

It is inspired by the popular website Houzz.com and some of the assets like, color theme, have been derived from the mentioned. This has made us practicing the various technologies learned throughout the course and made them better.

We have used following technologies in our project:

- HTML
- CSS
- Bootstrap
- JQuery

- JavaScript
- Node.js
- Sequelize.js
- SQLite
- Passport.js

For front-end, we have used HTML, CSS, JQuery and JavaScript.

These technologies have pushed our limits of thinking the way we couldn't have thought about if we hadn't created this project

For the basic structure of the website, we have used HTML. It has helped us creating the basic markup of the website. As we know, layout of any website is the skeleton of it. Your website can't stand without the skeleton; hence HTML has been very useful.

HTML/CSS

The building blocks of web pages - HTML and CSS. Learn how to use the latest HTML5 web development technologies along with CSS3 stylesheets to create responsive eye-catching web-sites. We also cover UI design patterns like:

Table Layouts

Flex Boxes

Bootstrap Columns

Media Queries and Mobile Responsive Design

Grids

Javascript

HTML and CSS brings the content and design together, while Javascript is at the heart of all action. Learn to act on events like 'clicks', 'hovers' and 'drag and drop'. Javascript is one of the most powerful and ubiquitous languages in modern software development and our course covers it in depth including:

ECMAScript 6 Syntax and Standards

Function Closures and IIFEs

Classes, Constructors and Prototypes

Lexical Scopes, Arrow Functions and Variable spreading

NODEJS

Javascript is not just only on the frontend, but a potent force on the server too. Built by Ryan Dahl in 2009 as a platform to run JS code on bare metal, NodeJS is currently the fastest growing ecosystem. We will cover:

NodeJS Modules

Filesystem API, Events and Streams

ExpressJS Framework for creating REST APIs

Handlebars for server side web rendering

Socket.IO for realtime communication

Databases

What is a server if it cannot store data on databases. We will cover storing data in:

Flat files on server

MySQL

Using ORMs like Sequelize

MongoDB

Advanced Topics & Deployment

Finally, we will cover adding basic security to websites including user authentication and authorization, SSL transport, checking for SQL injection and other vulnerabilities. We will also cover how to deploy your server to commonly used infrastructure providers like Amazon Web Services, Google Cloud or DigitalOcean

Security

Scaling

Using Frontend Templates

3rd Party Libraries and Frameworks

Deploying a live web project

CHAPTER - 7

Conclusion

The increased popularity of JavaScript has dramatically changed the face of web development today. Several years ago, it was even difficult to imagine things which we can do with JavaScript easily running in the browser as well as on the server. NodeJS is such a JavaScript runtime environment which is built on Chrome's V8 JavaScript engine. It is fast, efficient and is a highly scalable web server for web applications. That's why it has become the most preferred JS framework for all web developers.

The developed applications of Node.js are often buckled with MongoDB and Express.js that also use JavaScript. Express.js helps manage the middleware of a node application, while MongoDB is a non-SQL database based on documents.

Now we will discuss the benefits of using Node JS for web app development.

1) Real-Time Web apps

NodeJS web development allows you to create real-time web apps at a lighting speed same like the amount of time required for making a very simple blog in PHP. Therefore, NodeJS is a clear winner when it comes to creating multi-user real-time web applications for chat apps and gaming apps.

2) Easy Coding

With Node.js web development, it allows web developers to code in JavaScript for both the server and client. Hence, it makes it convenient for transferring data between the client and the server to coordinate the work well simultaneously. Data changes made to the server appear instantly on the client and the web page that displays this data automatically updates. Node.js satisfies all the needs of the development process and offers scalable and fast network applications.

3) Dynamic NPM

Since NodeJS is an open-source platform, it provides an edge with a shared repository of modules and dynamic tools. The number of modules that are more than 60000 in the NPM(Node Package Manager) has increased with the significant growth and is about to surpass the RoR platform (Ruby on Rails). Since NPM is robust and super fast, it helps to make the dependency management perfect. With the great popularity of Node.js, the community of nodes is strengthened day by day.

4) Hosting

Hosting has a gained momentum after NodeJS is highly demanded by more and more number of web developers for their web app development project. PaaS (Platform as a Service) service providers like Modulus and Heroku are allowing node deployments officially without any problem.

5) Community Friendly

NodeJS provides a large open source community who delivers various outstanding modules that makes Node JS applications shine everywhere. One of the popular ones is Socket.io, a module which can handle the constant communication between the client and the server that allows the server to send updates in real time to the clients. It works with the best technology used to form these links away from the developer, especially for specific clients.

Conclusion:

As we know that nothing is perfect in this universe, there are some things that were not to the liking of some users, but the Node.js API is still changing and, as it matures, certain parts are more reliable than others. Nodejs has brought a great revolution in the world of development and the most popular option for several brands such as eBay, Walmart, Yahoo, etc. Node.js is really a blessing for a developer and should be used by all companies.

It now plays a critical role in the technology stack. You can take advantage of the benefits of Node.js and enjoy fast and scalable network applications by hiring top Node JS development company like ValueCoders. It is a leading software development company in India having a

team of expert Node JS developers who have successfully delivered more than 4200 projects to 2500+ satisfied customers in last 13 years.

Thank you