Mini Project Report

on

INTERVIEW QUESTIONS WEBSITE

Submitted to

Ajay Kumar Garg Engineering College, Ghaziabad

(BTech Information Technology Sem 5th, 2020-21)

(KSC-554 Project Mini Project or Internship Assessment Report)

Submitted To: -

Mr Sarvachan Verma

Submitted By:-

Kritika Jain(1900270130087)

Kajal Srivastava(1900270130080)

Dr. A.P.J. Abdul Kalam Technical University,
Uttar Pradesh, Lucknow

(I) ACKNOWLEDGEMENT

I want to express my sincere gratitude and thanks to Prof. Anu Chaudhary (H.O.D. IT), Ajay Kumar Garg Engineering College, Ghaziabad for granting me permission for my internship in the field of “Web Technology”.

I would like to thank Dr Parneet Kaur (Faculty of IT Department) for her cooperative attitude. Under her guidance, I was able to complete my internship in Web Technology.

Finally, I pay my thankful regard and gratitude to the team members of Ajay Kumar Garg Engineering College, Ghaziabad for their valuable help, support and guidance.

Kritika Jain

1900270130087

(III year IT)

Kajal Srivastava

1900270130080

(III year IT)

(II) ABSTRACT

The project is an interview questions website. It consists of all the important technical questions which are generally asked in the interviews. There will be three levels- basic, medium and advanced for the questions. The user can see and solve the questions accordingly. Also, the solutions along with pictorial demonstration and graphs are provided for every questions. It is a medium for aspiring students to be ready to face the challenge of the competitive world. It will act as a good practice platform for daily exercises and last-minute practice.

The objective of this project is to provide user a simple and precise platform for all the important technical questions with the best solution. The student can prepare for the interview from the single website only. The picture and graphs for algorithm will help the user to understand the question easily.

(III) TABLE OF CONTENTS

CHAPTER 1. Overview of Web Technology **1-3**

1.1 Internship 1

1.2 About the Course **2**

CHAPTER 2. PROJECT: Interview Questions Website **3-30**

2.1 Introduction **4**

2.2 Problem Statement and Description **5**

2.3 Requirements **8**

2.4 Technologies Used **9-21**

2.5.1 HTML **9**

2.5.2 CSS **11**

2.5.3 JavaScript **13**

2.5.4 JQuery **15**

2.5.5 React **17**

2.5.6 BootStrap **20**

2.5 Data Flow Diagram **22**

2.6 Coding **23**

2.7 Snapshot of project **28**

CHAPTER 3. CONCLUSION **31**

3.1 Summary **31**

(IV) LIST OF FIGURES

1. **Home Page Snapshot………………………………………. 28**
2. **DSA Page Snapshot……………………………………..... 29**
3. **Array Page Snapshot………………………………………. 29**
4. **DBMS Page Snapshot………………………………………. 30**
5. **Home Page Code……………………………………………….. 23**
6. **String Question Snapshot…………………………………… 28**
7. **Footer Snapshot……………………………………………… 30**

Chapter 1: OVERVIEW OF WEB TECHNOLOGY

1.1 Internship

Internships are supervised, structured learning experiences in a professional setting that allow you to gain valuable work experience in a student's chosen field of study. Internships require a minimum of 120 hours (typically, at least 10 hours per week during the fall and spring and either part-time or full-time during the summer).

An internship gives a student the opportunity for career exploration and development, and to learn new skills. It offers the employer the opportunity to bring new ideas and energy into the workplace, develop talent and potentially build a pipeline for future full-time employees.

As an intern, you get a chance to work side by side with accomplished industry professionals and get a pretty good idea of what an entry-level role might entail. You'll not only gain real work experience, but also meet and learn from the pros. And you'll start to build your own network, from your fellow interns to seasoned leaders.

One other less obvious but equally important benefit of an internship is the chance to figure out what you don't want to do. It's often difficult to know where to even start when it comes to job searching. Internships give you the chance to try a few things out without committing. If you're lucky, you'll find something you love. And if not, you'll at least know what doesn't work for you. When it comes to something as tricky as finding the right career, the more information you have to work with, the better.

As internships have gotten more and more common, employers expect to see them on resumes. Applicants with previous work experience are much more competitive than those who only have relevant coursework. Internships offer you the chance to not just build relevant skills and learn about the field, but to demonstrate those skills and industry acumen on the job. For most employers, even ones who are extremely adept at hiring new graduates, nothing quite makes up for real-life experience.

Companies also use internships as talent pipelines to fill their own full-time positions. For employers, internships are a lot of things: a super-extended interview, a training program, and (frequently) a smart way to hire for open roles. This means some college students can walk into their senior years with job offers in hand (and therefore have a much less stressful last year at school).

In short, internships can help you figure out what you want to do with your career and then make it easier to land your first full-time job in that industry.

1.1 About the Course

Web Technology refers to the various tools and techniques that are utilized in the process of communication between different types of devices over the internet. A web browser is used to access web pages. Web browsers can be defined as programs that display text, data, pictures, animation, and video on the Internet. Hyperlinked resources on the World Wide Web can be accessed using software interfaces provided by Web browsers.

Web Technology can be classified into the following sections:

1) **World Wide Web (WWW):** The World Wide Web is based on several different technologies: Web browsers, Hypertext Markup Language (HTML) and Hypertext Transfer Protocol (HTTP).

2) **Web Browser:** The web browser is an application software to explore www (World Wide Web). It provides an interface between the server and the client and requests to the server for web documents and services.

3) **Web Server:** Web server is a program which processes the network requests of the users and serves them with files that create web pages. This exchange takes place using Hypertext Transfer Protocol (HTTP).

4) **Web Pages:** A webpage is a digital document that is linked to the World Wide Web and viewable by anyone connected to the internet has a web browser.

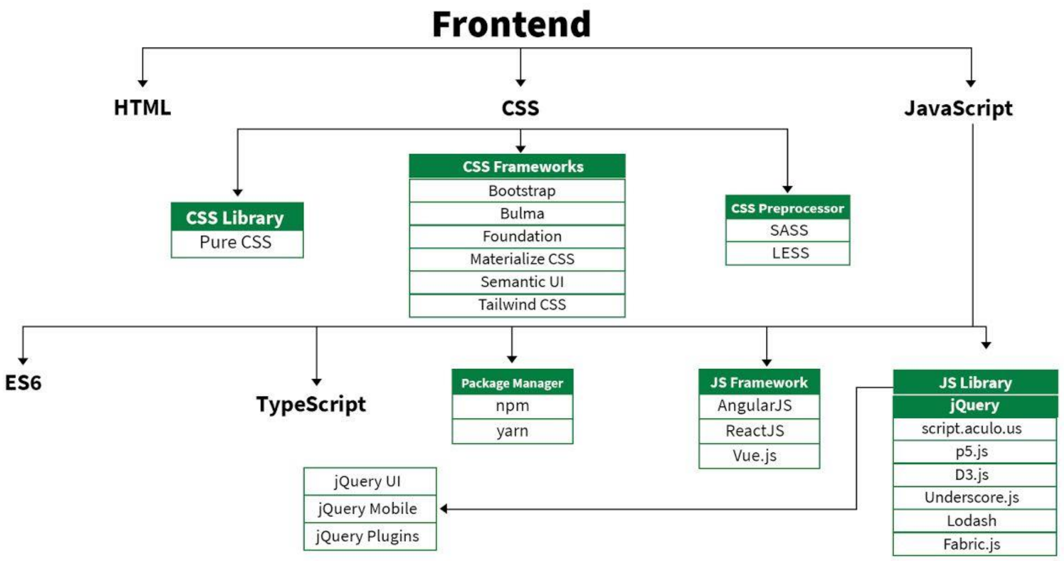
Web Development: Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

Web development can range from developing a simple single static page of plain text to complex web applications, electronic businesses, and social network services. A more comprehensive list of tasks to which Web development commonly refers, may include Web engineering, Web design, Web content development, client liaison, client-side/server-side scripting, Web server and network security configuration, and e-commerce development.

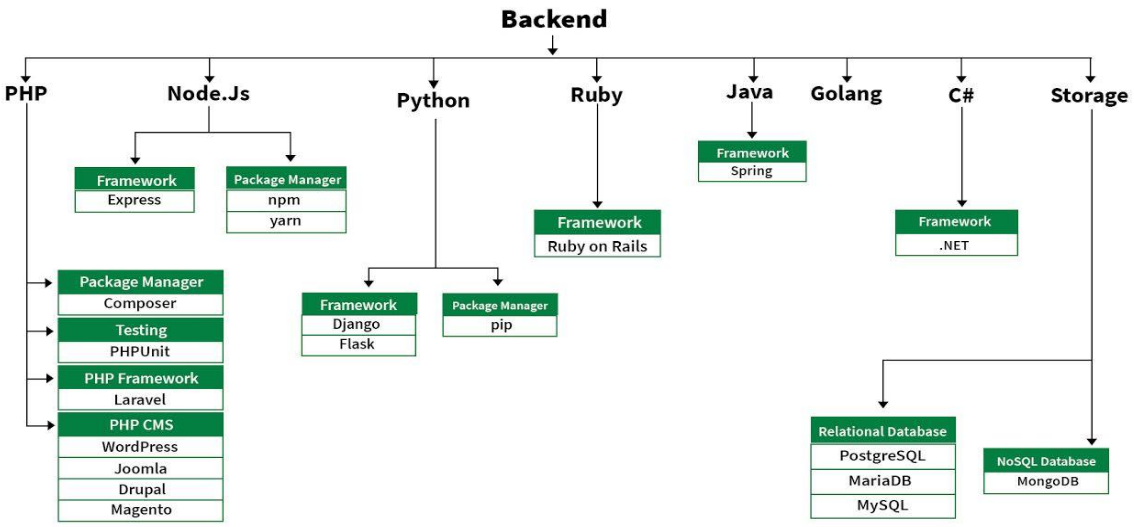
Among Web professionals, Web development usually refers to the main non-design aspects of building Web sites: writing markup and coding. Web development may use content management systems (CMS) to make content changes easier and available with basic technical skills.

Web Development can be classified into two ways:

- **Frontend Development**: The part of a website that the user interacts directly is termed as front end. It is also referred to as the ‘client side’ of the application.



- **Backend Development:** Backend is the server side of a website. It is the part of the website that users cannot see and interact. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.



Chapter 2: PROJECT (Interview questions website)

2.1 Introduction

The project is an interview questions website. It consists of all the important technical questions which are generally asked in the interviews. There will be three levels- basic, medium and advanced for the questions. The user can see and solve the questions accordingly. Also, the solutions along with pictorial demonstration and graphs are provided for every questions. It is a medium for aspiring students to be ready to face the challenge of the competitive world. It will act as a good practice platform for daily exercises and last-minute practice.

The main objective of the project is to learn and practice on almost all coding interview questions asked historically and get referred to the best tech companies.

The objective of this project is to provide user a simple and precise platform for all the important technical questions with the best solution. The student can prepare for the interview from the single website only. The picture and graphs for algorithm will help the user to understand the question easily.

The user will get to run through exactly the kinds of questions which are seen in a real interview. Mock technical interviews & mentorship sessions.

2.2 Problem Statement and Description

An interview is essentially a structured conversation where one participant asks questions, and the other provides answers. In common parlance, the word "interview" refers to a one-on-one conversation between an interviewer and an interviewee. The interviewer asks questions to which the interviewee responds, usually providing information. That information may be used or provided to other audiences immediately or later. This feature is common to many types of interviews – a job interview or interview with a witness to an event may have no other audience present at the time, but the answers will be later provided to others in the employment or investigative process. An interview may also transfer information in both directions.

Technical interviews are typically conducted by employers who are hiring computer science, engineering, IT or other technical professionals. Technical interviews allow interviewers to assess candidates' technical skills, problem-solving skills and critical thinking skills.

In a technical interview, it is not uncommon for interviewers to present brain teasers, numerical reasoning problems or technical assessment problems that test a candidate's ability to create solutions and solve problems. Hiring managers may conduct technical interviews on the phone or through online face-to-face communication platforms as well as in-person.A technical interview evaluates a candidate's abilities to think creatively, adapt solutions and find ways to solve technical problems. Success in a technical interview signifies that a candidate's qualifications meet the job requirements. Because a technical interview assesses a candidate's approaches to solving problems, the interviewer may use a set of technical problems that the candidate is expected to solve during an interview.

The interviewer can assess the way the candidate approaches the problem, breaks it down and solves it. While arriving at a correct answer can be beneficial in this type of interviewing scenario, interviewers are typically looking for your methodology when solving problems and may place more emphasis on that instead of a correct answer.

This webpage provides user a complete platform to prepare for a technical interview. Basically, the first round consists of the online test which generally has three different rounds-

- Aptitude
- Verbal Analogies
- Coding

Aptitude: There are several categories of aptitude test that are designed to measure technical skills. Mechanical reasoning, spatial reasoning and diagrammatic reasoning are just some of the more specialised tests. Fault diagnosis is another test commonly found in an engineering battery of tests. Numerical reasoning is seen as an important test for many engineering employers as they look for evidence that you can identify the important elements of numerical data and pick it out.

Engineering aptitude questions tend to be based on engineering content. You generally do not need to have outside knowledge on the numerical and verbal reasoning tests, but the tests examining your technical reasoning skills will assume a level of knowledge.

Some of the skills that employers are looking for in engineers and will therefore be tested through the engineering aptitude tests are:

- Critical thinking
- Ability to detect and analyse systematic themes in data
- Attention to detail

Types of Aptitude tests:

- **Abstract Reasoning Test** :- Tests that measure abstract reasoning assess a candidate's ability to think laterally and from a unique perspective. Abstract reasoning tests also assess an individual's ability to make logical connections between the available sets of information. A high level of abstract reasoning is related to creativity and helps solve novel problems. Therefore, learning new skills is easy for people with strong abstract reasoning abilities as they can process ambiguous information effectively.
- **Spatial Reasoning Test**:- Spatial reasoning defines a candidate's ability to understand and visualize two-dimensional and three-dimensional patterns and shapes. Spatial reasoning tests help identify people who can easily understand an object's spatial distribution and movement.
- **Logical and Critical Reasoning Test**:-These tests help evaluate the candidates on their logic inference irrespective of the job roles. It's about drawing sensible conclusions based on given information and statements. Logical thinking is critical in daily life and various business scenarios. Likewise, critical thinking is a creative way to approach a problem, as it involves validations using comprehensive judgment and assumptions.

Verbal Analogies:

Engineering communication skills are essential for presenting concepts and designs, and critical to the engineering design process. Engineers with the ability to clearly communicate can confidently present their ideas to decision-makers in presentations, meetings, and reports. Company leaders expect to read and hear professional-level presentations. Spelling mistakes in a report or a mumbled presentation may make an executive think twice about an engineering design proposal's reliability. This can make engineering communication skills as important as technical skills.This type of aptitude test evaluates how well an individual can use words. Excellent command of language is one of the critical aspects for ensuring good communication. Recruiters can ensure the candidates' job readiness and measure their working skills using language and comprehension evaluation.

Coding:

Coding, in the simplest of terms, is telling a computer what you want it to do, which involves typing in step-by-step commands for the computer to follow.

Computers are not clever things, however they are very obedient. They will do exactly what you want them to do, so long as you tell them how to do it correctly.

Learning to code has been likened to learning a foreign language, or perhaps more specifically a family of foreign languages.

There are many different coding languages, each one designed with certain things in mind. Examples include C, a 'low level' but fast programming language that is good for anything graphically intensive like games; Javascript, which was specifically designed for dealing with web content; and Perl, a multi-functional language that is often referred to as the 'swiss army knife' of programming.

Data structures and algorithms play a major role in implementing software and in the hiring process as well. A lot of students and professionals have this question that why these companies' interviews are focused on DSA instead of language/frameworks/tools specific questions? Let us explain why it happens…

When you ask someone to make a decision for something the good one will be able to tell you "I choose to do X because it's better than A, B in these ways. I could have gone with C, but I felt this was a better choice because of this". In our daily life, we always go with that person who can complete the task in a short amount of time with efficiency and using fewer resources. The same things happen with these companies. The problem faced by these companies is much harder and on a much larger scale. Software developers also have to make the right decisions when it comes to solving the problems of these companies.

Code powers our digital world. Every website, smartphone app, computer programme, calculator and even microwave relies on code in order to operate. This makes coders the architects and builders of the digital age.

Over the next 10 years it is estimated that there will be 1.4 million jobs in computer sciences and only around 400,000 graduates qualified to do them.

Jobs not directly linked to computer sciences – such as banking, medicine and journalism – will also be affected by the need for at least an understanding of programming and coding.

2.3 Requirements

SYSTEM REQUIREMENT

The requirement definition is concerned with the analysis of the existing system with the aim of determining and structuring the requirement of the proposed system. It is achieved with the aid of user requirement. The system needs to be adept with the currentWindows/ OperatingSystem.

REQUIREMENT SPECIFICATION

Requirement Specification a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains nonfunctional requirements. Non-functional requirements impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints)

FUNCTIONAL REQUIREMENTS

Functional requirements define the specific functions that the system performs, along with the data operated on by the functions. The functional requirements are presented in scenarios that depict an operational system from the perspective of its end users. Included are one or more examples of all system features and an enumeration of all the specific requirements associated with these features.

- The system shall incorporate mechanism to authenticate its users

- The system shall verify and validate all user input and should notify in case of error detection and should help the user in error correction

- The system shall allow sharing of files in the system

- The system shall allow display of reminder/warning messages.

NON-FUNCTIONAL REQUIREMENT

Non-functional requirements address aspects of the system other than the specific functions it performs. These aspects include system performance, costs, and such general system characteristics as reliability, security, and portability. The non-functional requirements also address aspects of the system development process and operational personnel. It includes the following:

- The system shall be user friendly and consistent
- The system shall provide attractive graphical interface for the user
- The system shall allow developer access to installed environment
- The system shall target customer base

2.4 Technologies Used

2.4.1 HTML:

The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document**.**

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991. It describes 18 elements comprising the initial, relatively simple design of HTML. Except for the hyperlink tag, these were strongly influenced by SGMLguid, an in-house Standard Generalized Markup Language (SGML)-based documentation format at CERN. Eleven of these elements still exist in HTML 4.

HTML is a markup language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML markup are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS.

HTML versions timeline

HTML 2

November 24, 1995

HTML 3

January 14, 1997

HTML 4

December 18, 1997

HTML 5

October 28, 2014

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags). The text between <html> and </html> describes the web page, and the text between <body> and </body> is the visible page content. The markup text <title>This is a title</title> defines the browser page title shown on browser tabs and window titles, and the tag <div> defines a division of the page used for easy styling.

The Document Type Declaration <!DOCTYPE html> is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.

ELEMENTS

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus: <p>.In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" <p> and "end tag" </p>. The text content of the element, if any, is placed between these tags.Tags may also enclose further tag markup between the start and end, including a mixture of tags and text.

ATTRIBUTES

Most of the attributes of an element are name-value pairs, separated by = and written within the start tag of an element after the element's name. The value may be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML). Leaving attribute values unquoted is considered unsafe.In contrast with name-value pair attributes, there are some attributes that affect the element simply by their presence in the start tag of the element,like the ismap attribute for the img element.

DATA TYPES

HTML defines several data types for element content, such as script data and stylesheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

DOCUMENT TYPE DECLARATION

HTML documents are required to start with a Document Type Declaration (informally, a "doctype"). In browsers, the doctype helps to define the rendering mode—particularly whether to use quirks mode.

The original purpose of the doctype was to enable parsing and validation of HTML documents by SGML tools based on the Document Type Definition (DTD). The DTD to which the DOCTYPE refers contains a machine-readable grammar specifying the permitted and prohibited content for a document conforming to such a DTD. Browsers, on the other hand, do not implement HTML as an application of SGML and by consequence do not read the DTD.

2.4.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL**.**

Variations

CSS has various levels and profiles. Each level of CSS builds upon the last, typically adding new features and typically denoted as CSS 1, CSS 2, CSS 3, and CSS 4. Profiles are typically a subset of one or more levels of CSS built for a particular device or user interface. Currently there are profiles for mobile devices, printers, and television sets. Profiles should not be confused with media types, which were added in CSS 2.

CSS 1

The first CSS specification to become an official W3C Recommendation is CSS level 1, published on December 17, 1996. Håkon Wium Lie and Bert Bos are credited as the original developers. Among its capabilities are support for

- Font properties such as typeface and emphasis
- Color of text, backgrounds, and other elements
- Text attributes such as spacing between words, letters, and lines of text
- Alignment of text, images, tables and other elements
- Margin, border, padding, and positioning for most elements
- Unique identification and generic classification of groups of attributes
- The W3C no longer maintains the CSS 1 Recommendation.

CSS 2

CSS level 2 specification was developed by the W3C and published as a recommendation in May 1998. A superset of CSS 1, CSS 2 includes a number of new capabilities like absolute, relative, and fixed positioning of elements and z-index, the concept of media types, support for aural style sheets (which were later replaced by the CSS 3 speech modules) and bidirectional text, and new font properties such as shadows.The W3C no longer maintains the CSS 2 recommendation.

CSS 2.1

CSS level 2 revision 1, often referred to as "CSS 2.1", fixes errors in CSS 2, removes poorly supported or not fully interoperable features and adds already implemented browser extensions to the specification. To comply with the W3C Process for standardizing technical specifications, CSS 2.1 went back and forth between Working Draft status and Candidate Recommendation status for many years. CSS 2.1 first became a Candidate Recommendation on February 25, 2004, but it was reverted to a Working Draft on June 13, 2005 for further review. It returned to Candidate Recommendation on 19 July 2007 and then updated twice in 2009. However, because changes and clarifications were made, it again went back to Last Call Working Draft on 7 December 2010.

CSS 2.1 went to Proposed Recommendation on 12 April 2011.After being reviewed by the W3C Advisory Committee, it was finally published as a W3C Recommendation on 7 June 2011.

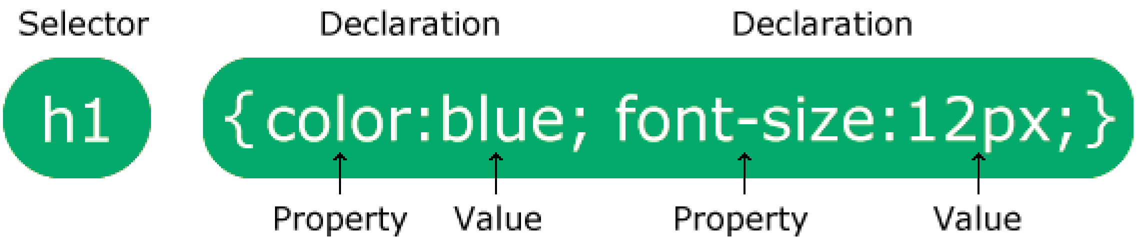
CSS 2.1 was planned as the first and final revision of level 2—but low priority work on CSS 2.2 began in 2015.

CSS 3

"CSS3" redirects here. For other uses, see CSS3 (disambiguation).

CSS Syntax

A CSS rule consists of a selector and a declaration block.



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.Each declaration includes a CSS property name and a value, separated by a colon.

2.4.3 JAVASCRIPT

JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification.JavaScript is high-level, often just-in-time compiled and multi-paradigm. It has dynamic typing, prototype-based object-orientation and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.Over 97% of websites use it client-side for web page behavior,often incorporating third-party libraries.All major web browsers have a dedicated JavaScript engine to execute the code on the user's device.As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.JavaScript engines were originally used only in web browsers, but they are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

JAVASCRIPT WHERE TO USE

1.JavaScript in <head> or <body>

You can place any number of scripts in an HTML document .Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

2.JavaScript in <head>

a JavaScript function is placed in the <head> section of an HTML page.

3.External JavaScript

JavaScript was initially created to “make web pages alive”.

The programs in this language are called scripts. They can be written right in a web page’s HTML and run automatically as the page loads.

Scripts are provided and executed as plain text. They don’t need special preparation or compilation to run.

In this aspect, JavaScript is very different from another language called Java.

Modern JavaScript is a “safe” programming language. It does not provide low-level access to memory or CPU, because it was initially created for browsers which do not require it.

JavaScript’s capabilities greatly depend on the environment it’s running in. For instance, Node.js supports functions that allow JavaScript to read/write arbitrary files, perform network requests, etc.

In-browser JavaScript can do everything related to webpage manipulation, interaction with the user, and the webserver.

For instance, in-browser JavaScript is able to:

- Add new HTML to the page, change the existing content, modify styles.
- React to user actions, run on mouse clicks, pointer movements, key presses.
- Send requests over the network to remote servers, download and upload files (so-called AJAX and COMET technologies).
- Get and set cookies, ask questions to the visitor, show messages.
- Remember the data on the client-side (“local storage”).

There are at least three great things about JavaScript:

- Full integration with HTML/CSS.
- Simple things are done simply.
- Supported by all major browsers and enabled by default.
- JavaScript is the only browser technology that combines these three things.

That’s what makes JavaScript unique. That’s why it’s the most widespread tool for creating browser interfaces.

That said, JavaScript also allows to create servers, mobile applications, etc.

2.4.4 JQUERY

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

DOM Traversal and Manipulation

Get the <button> element with the class 'continue' and change its HTML to 'Next Step...'

```
$( "button.continue" ).html( "Next Step..." )
```

Event Handling

Show the #banner-message element that is hidden with display:none in its CSS when any button in #button-container is clicked.

```
var hiddenBox = $( "#banner-message" );
$( "#button-container button" ).on( "click", function( event ) {
  hiddenBox.show();
});
```

Ajax

Call a local script on the server /api/getWeather with the query parameter zipcode=97201 and replace the element #weather-temp's html with the returned text.

```
$.ajax({
  url: "/api/getWeather",
  data: {
    zipcode: 97201
  },
  success: function( result ) {
    $( "#weather-temp" ).html( "<strong>" + result + "</strong> degrees" );
  }
});
```

jQuery is a lightweight, "write less, do more", JavaScript library.The purpose of jQuery is to make it much easier to use JavaScript on your website.jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

Tip: In addition, jQuery has plugins for almost any task out there.

Why jQuery?

There are lots of other JavaScript libraries out there, but jQuery is probably the most popular, and also the most extendable.

Many of the biggest companies on the Web use jQuery, such as:

- Google
- Microsoft
- IBM
- Netflix

2.4.5 REACT

React (also known as React.js or ReactJS) is a free and open-source front-end JavaScript library[3] for building user interfaces based on UI components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies.[4][5][6] React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Notable features

Declarative
React adheres to the declarative programming paradigm. Developers design views for each state of an application, and React updates and renders components when data changes.

Components
React code is made of entities called components. Components can be rendered to a particular element in the DOM using the React DOM library. When rendering a component, one can pass in values that are known as "props":

```
ReactDOM.render(<Greeter greeting="Hello World!" />,
document.getElementById('myReactApp'));
```

The two primary ways of declaring components in React is via function components and class-based components.

Function components
Function components are declared with a function that then returns some JSX.

```
const Greeter = (props) => <div>Hello, {props.name}!</div>;
```

Class-based components
Class-based components are declared using ES6 classes**.**

```
class ParentComponent extends React.Component {
  state = { color: 'green' };
  render() {
    return (
      <ChildComponent color={this.state.color} />
    );
  }}
```

Virtual DOM

Another notable feature is the use of a virtual Document Object Model, or virtual DOM. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently.[9] This process is called reconciliation. This allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change. This selective rendering provides a major performance boost.[10] It saves the effort of recalculating the CSS style, layout for the page and rendering for the entire page.

Lifecycle methods

Lifecycle methods use a form of hooking that allows the execution of code at set points during a component's lifetime.

shouldComponentUpdate allows the developer to prevent unnecessary re-rendering of a component by returning false if a render is not required.

componentDidMount is called once the component has "mounted" (the component has been created in the user interface, often by associating it with a DOM node). This is commonly used to trigger data loading from a remote source via an API.

componentWillUnmount is called immediately before the component is torn down or "unmounted". This is commonly used to clear resource-demanding dependencies to the component that will not simply be removed with the unmounting of the component (e.g., removing any setInterval() instances that are related to the component, or an "eventListener" set on the "document" because of the presence of the component)

render is the most important lifecycle method and the only required one in any component. It is usually called every time the component's state is updated, which should be reflected in the user interface.

JSX

JSX, or JavaScript XML, is an extension to the JavaScript language syntax.[11] Similar in appearance to HTML, JSX provides a way to structure component rendering using syntax familiar to many developers. React components are typically written using JSX, although they do not have to be (components may also be written in pure JavaScript). JSX is similar to another extension syntax created by Facebook for PHP called XHP.

An example of JSX code:

```
class App extends React.Component {
  render() {
    return (
      <div>
        <p>Header</p>
        <p>Content</p>
        <p>Footer</p>
      </div>
    );}}
```

React hooks

Hooks are functions that let developers "hook into" React state and lifecycle features from function components. Hooks do not work inside classes — they let you use React without classes.

React provides a few built-in hooks like useState,useContext, useReducer , useMemo and useEffect.Others are documented in the Hooks API Reference.useState , useReducer and useEffect, which are the most used, are for controlling state and side effects respectively.

Rules of hooks

There are rules of hooks which describe the characteristic code pattern that hooks rely on. It is the modern way to handle state with React.

Hooks should only be called at the top level (not inside loops or if statements).

Hooks should only be called from React function components, not normal functions or class components.

Although these rules can't be enforced at runtime, code analysis tools such as linters can be configured to detect many mistakes during development. The rules apply to both usage of hooks and the implementation of custom hooks,which may call other hooks.

2.4.6 BOOTSTRAP

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.Bootstrap is completely free to download and use.
Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.
As of August 2021, Bootstrap is the tenth most starred project on GitHub, with over 152,000 stars, behind freeCodeCamp (over 328,000 stars), Vue.js framework, React library, TensorFlow and others.
Bootstrap, originally named Twitter Blueprint, was developed by Mark Otto and Jacob Thornton at Twitter as a framework to encourage consistency across internal tools. Before Bootstrap, various libraries were used for interface development, which led to inconsistencies and a high maintenance burden. According to Twitter developer Mark Otto:

A super small group of developers and I got together to design and build a new internal tool and saw an opportunity to do something more. Through that process, we saw ourselves build something much more substantial than another internal tool. Months later, we ended up with an early version of Bootstrap as a way to document and share common design patterns and assets within the company.

After a few months of development by a small group, many developers at Twitter began to contribute to the project as a part of Hack Week, a hackathon-style week for the Twitter development team. It was renamed from Twitter Blueprint to Bootstrap, and released as an open source project on August 19, 2011.It has continued to be maintained by Mark Otto, Jacob Thornton, and a small group of core developers, as well as a large community of contributors.

Bootstrap 2
On January 31, 2012, Bootstrap 2 was released, which added built-in support for Glyphicons, several new components, as well as changes to many of the existing components. This version supports responsive web design, meaning the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (whether desktop, tablet, or mobile phone).

Bootstrap 3
On August 19, 2013, Bootstrap 3, was released. It redesigned components to use flat design and a mobile first approach. Bootstrap 3 features new plugin system with namespaced events. Bootstrap 3 dropped Internet Explorer 7 and Firefox 3.6 support, but there is an optional polyfil for these browsers.

Bootstrap 4

Mark Otto announced Bootstrap 4 on October 29, 2014.The first alpha version of Bootstrap 4 was released on August 19, 2015.The first beta version was released on August 10, 2017.[10] Mark suspended work on Bootstrap 3 on September 6, 2016, to free up time to work on Bootstrap 4. Bootstrap 4 was finalized on January 18, 2018.

Significant changes include:

- Major rewrite of the code
- Replacing Less with Sass
- Addition of Reboot, a collection of element-specific CSS changes in a single file, based on Normalize
- Dropping support for IE8, IE9, and iOS 6
- CSS Flexible Box support
- Adding navigation customization options
- Adding responsive spacing and sizing utilities
- Switching from the pixels unit in CSS to root ems
- Increasing global font size from 14px to 16px for enhanced readability
- Dropping the panel, thumbnail, pager, and well components
- Dropping the Glyphicons icon font
- Huge number[quantify] of utility classes
- Improved form styling, buttons, drop-down menus, media objects and image classes
- Bootstrap 4 supports the latest versions of Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE10 and the latest Firefox Extended Support Release (ESR).

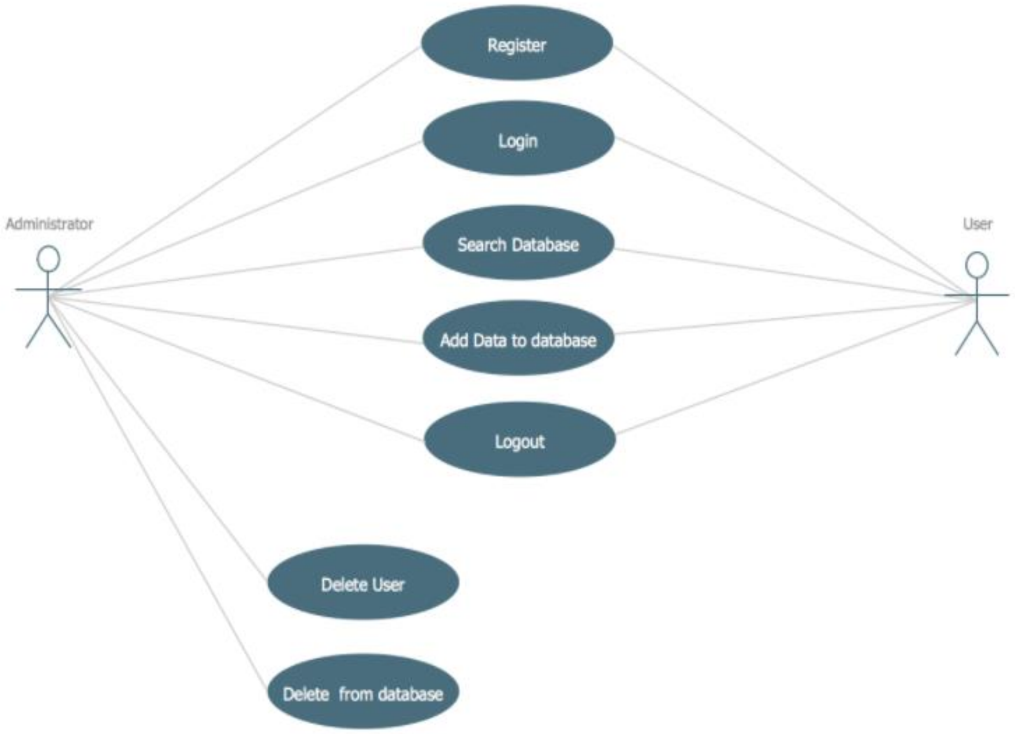
Bootstrap 5

Bootstrap 5 was officially released on May 5, 2021.

Major changes include:

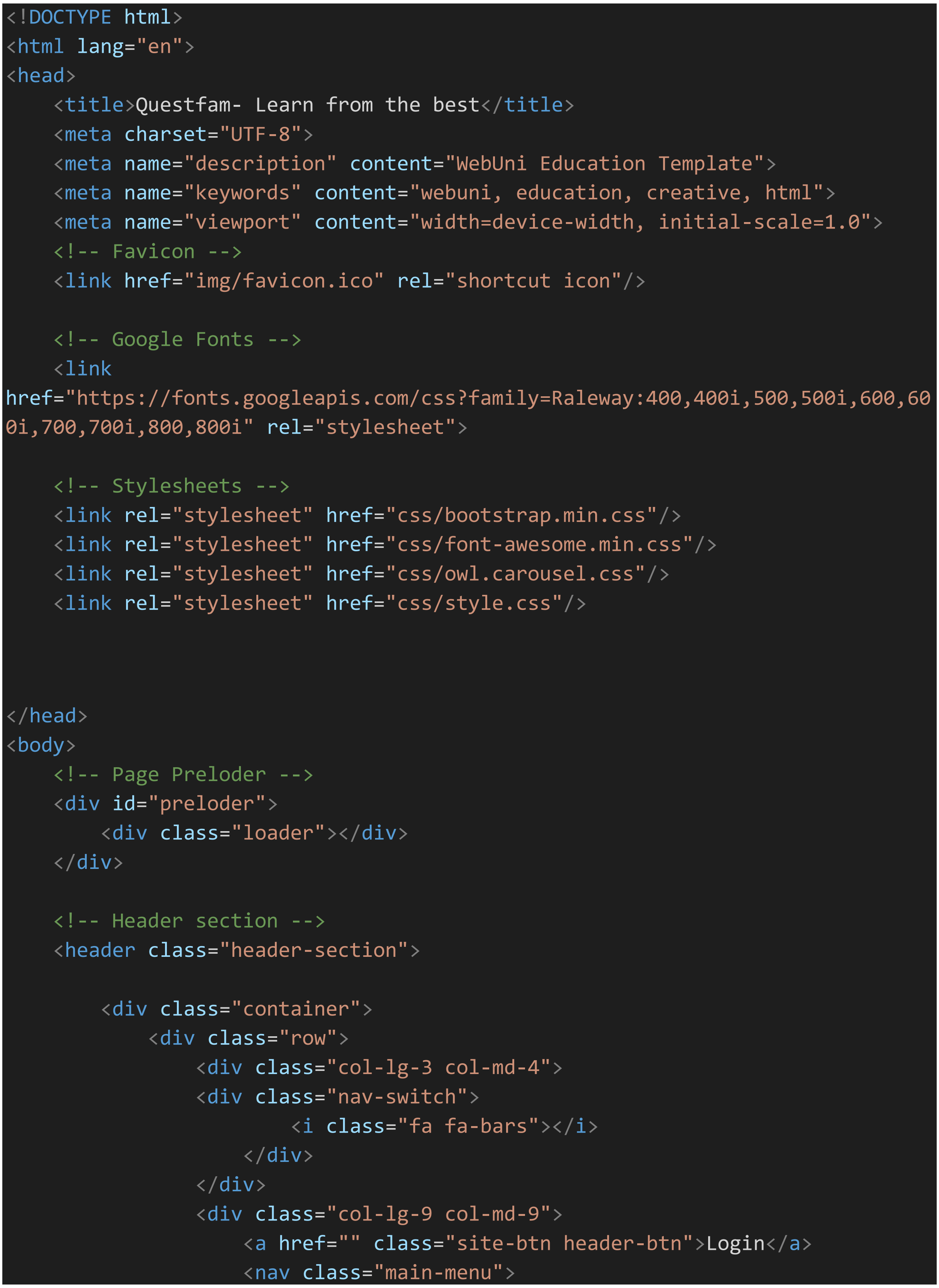
- New offcanvas menu component
- Removing dependence on jQuery in favor of vanilla JavaScript
- Rewriting the grid to support responsive gutters and columns placed outside of rows
- Migrating the documentation from Jekyll to Hugo
- Dropping support for Internet Explorer[17]
- Moving testing infrastructure from QUnit to Jasmine
- Adding custom set of SVG icons
- Adding CSS custom properties
- Improved API
- Enhanced grid system
- Improved customizing docs

2.5 Data Flow Diagram

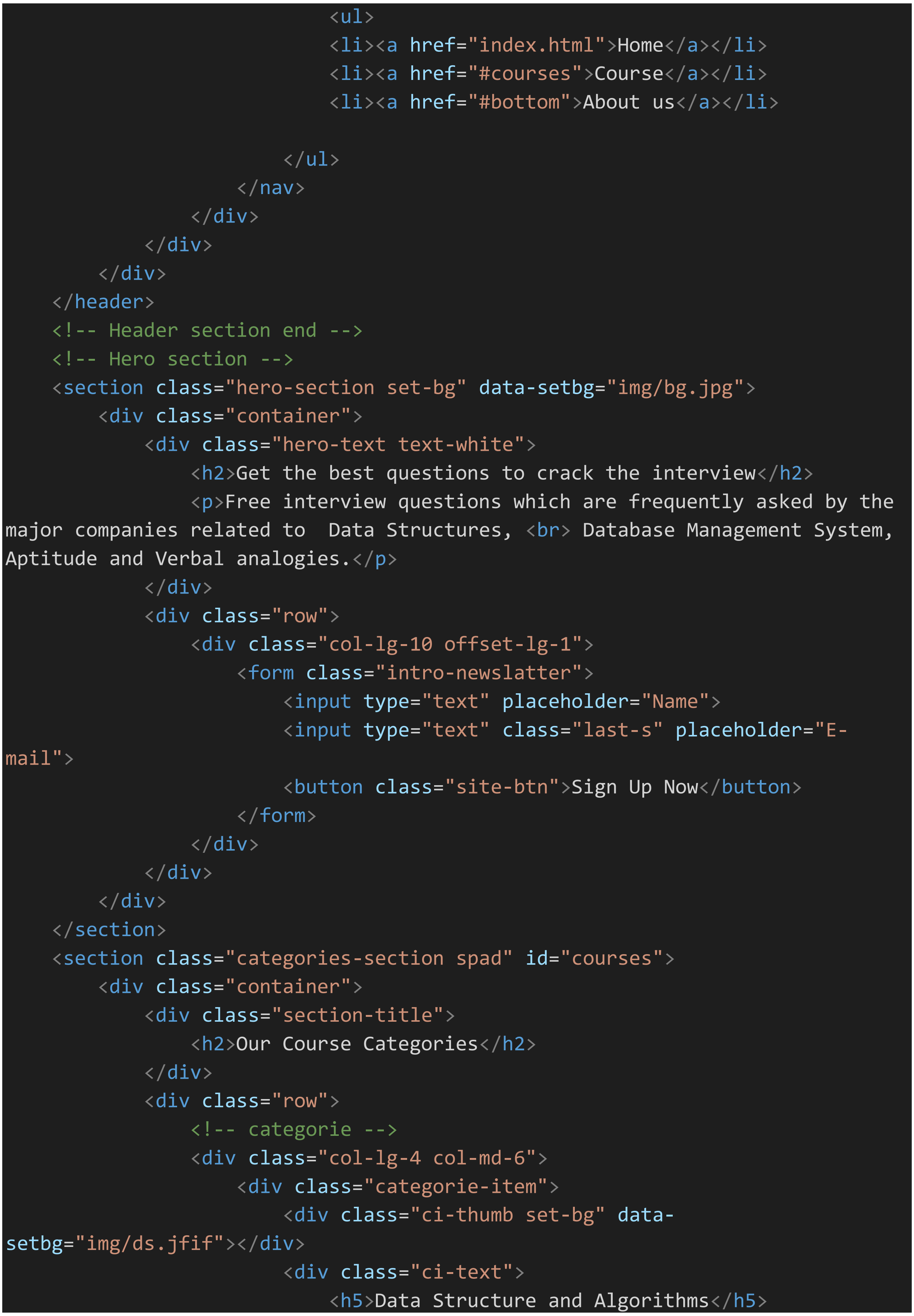


2.6 Coding

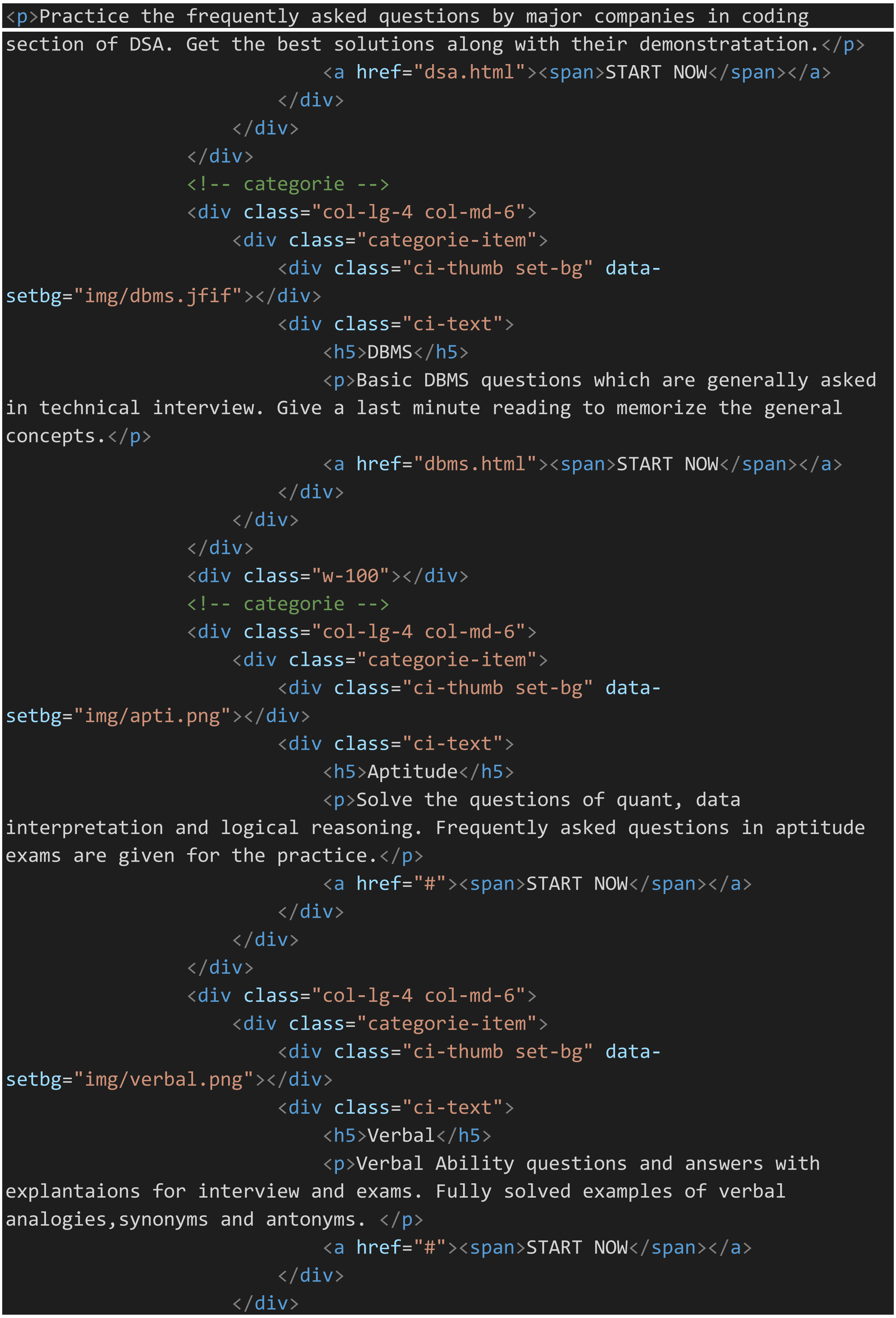
Index.html (Front webpage)



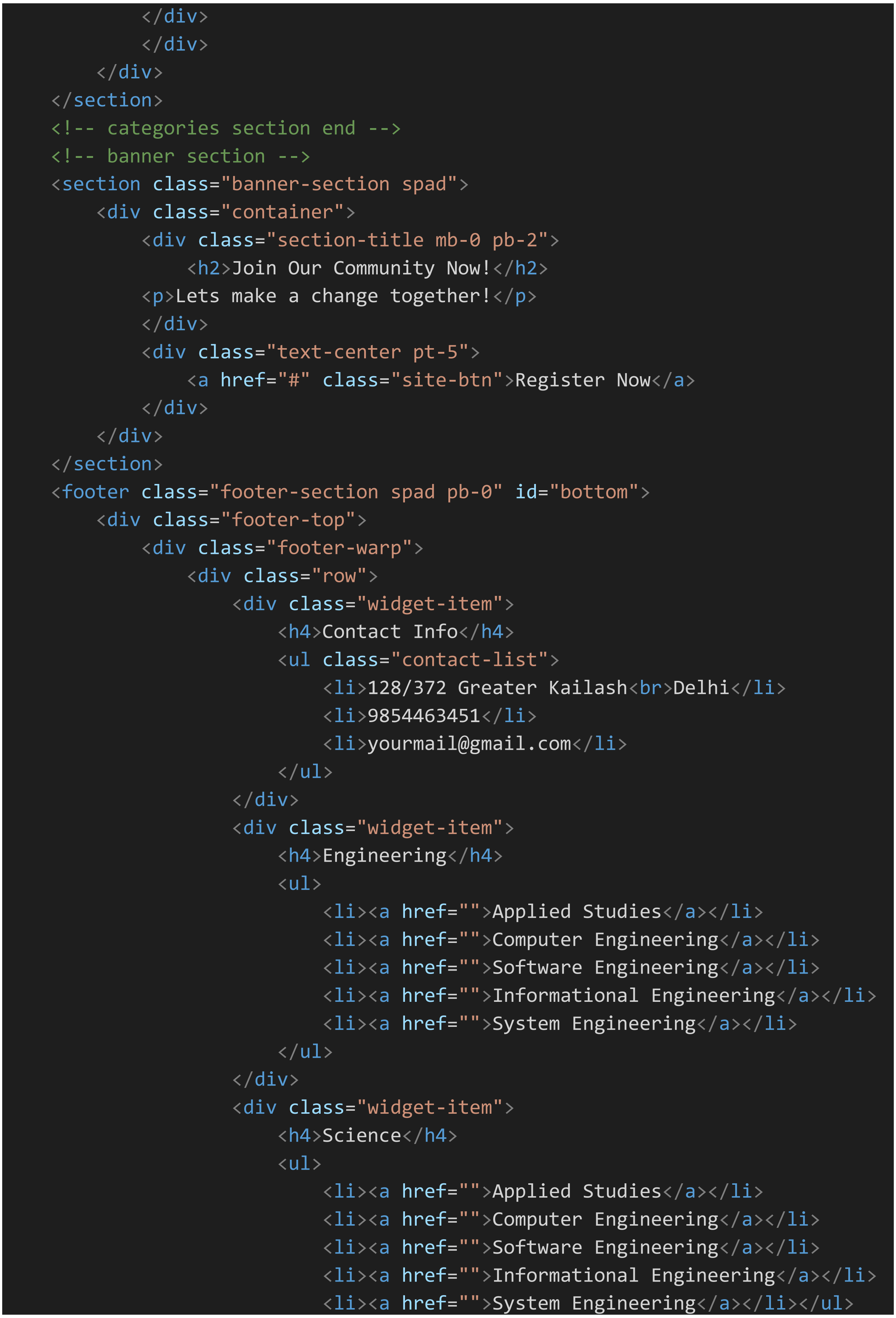
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Questfam- Learn from the best</title>
    <meta charset="UTF-8">
    <meta name="description" content="WebUni Education Template">
    <meta name="keywords" content="webuni, education, creative, html">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Favicon -->
    <link href="img/favicon.ico" rel="shortcut icon"/>

    <!-- Google Fonts -->
    <link
href="https://fonts.googleapis.com/css?family=Raleway:400,400i,500,500i,600,600i,700,700i,800,800i" rel="stylesheet">

    <!-- Stylesheets -->
    <link rel="stylesheet" href="css/bootstrap.min.css"/>
    <link rel="stylesheet" href="css/font-awesome.min.css"/>
    <link rel="stylesheet" href="css/owl.carousel.css"/>
    <link rel="stylesheet" href="css/style.css"/>



</head>
<body>
    <!-- Page Preloder -->
    <div id="preloder">
        <div class="loader"></div>
    </div>

    <!-- Header section -->
    <header class="header-section">

        <div class="container">
            <div class="row">
                <div class="col-lg-3 col-md-4">
                <div class="nav-switch">
                        <i class="fa fa-bars"></i>
                    </div>
                </div>
                <div class="col-lg-9 col-md-9">
                    <a href="" class="site-btn header-btn">Login</a>
                    <nav class="main-menu">
```



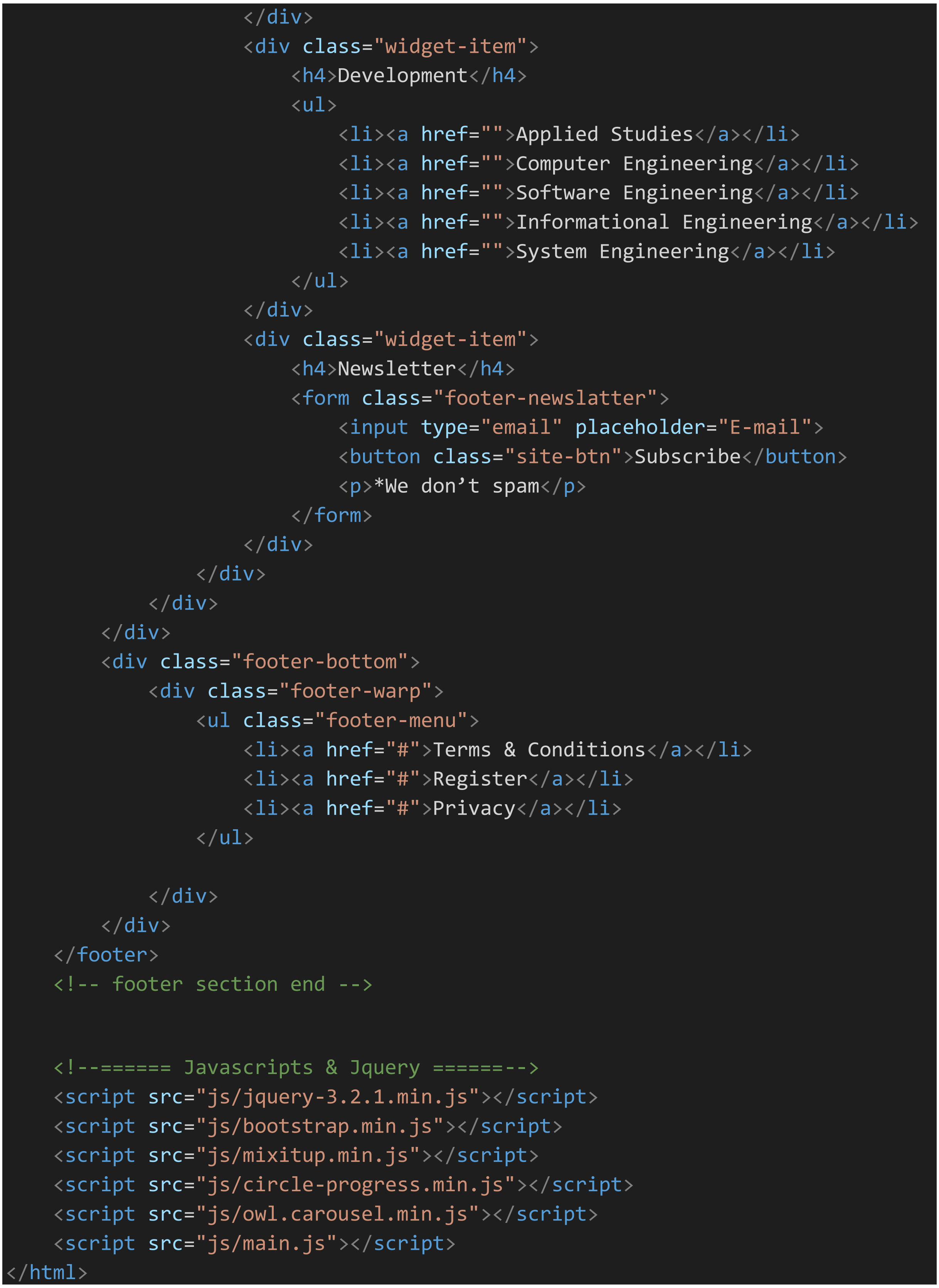
```
                            <ul>
                            <li><a href="index.html">Home</a></li>
                            <li><a href="#courses">Course</a></li>
                            <li><a href="#bottom">About us</a></li>

                        </ul>
                    </nav>
                </div>
            </div>
        </div>
    </header>
    <!-- Header section end -->
    <!-- Hero section -->
    <section class="hero-section set-bg" data-setbg="img/bg.jpg">
        <div class="container">
            <div class="hero-text text-white">
                <h2>Get the best questions to crack the interview</h2>
                <p>Free interview questions which are frequently asked by the
major companies related to  Data Structures, <br> Database Management System,
Aptitude and Verbal analogies.</p>
            </div>
            <div class="row">
                <div class="col-lg-10 offset-lg-1">
                    <form class="intro-newslatter">
                        <input type="text" placeholder="Name">
                        <input type="text" class="last-s" placeholder="E-
mail">
                        <button class="site-btn">Sign Up Now</button>
                    </form>
                </div>
            </div>
        </div>
    </section>
    <section class="categories-section spad" id="courses">
        <div class="container">
            <div class="section-title">
                <h2>Our Course Categories</h2>
            </div>
            <div class="row">
                <!-- categorie -->
                <div class="col-lg-4 col-md-6">
                    <div class="categorie-item">
                        <div class="ci-thumb set-bg" data-
setbg="img/ds.jfif"></div>
                        <div class="ci-text">
                            <h5>Data Structure and Algorithms</h5>
```



```
<p>Practice the frequently asked questions by major companies in coding
section of DSA. Get the best solutions along with their demonstratation.</p>
                            <a href="dsa.html"><span>START NOW</span></a>
                        </div>
                    </div>
                </div>
                <!-- categorie -->
                <div class="col-lg-4 col-md-6">
                    <div class="categorie-item">
                        <div class="ci-thumb set-bg" data-
setbg="img/dbms.jfif"></div>
                        <div class="ci-text">
                            <h5>DBMS</h5>
                            <p>Basic DBMS questions which are generally asked
in technical interview. Give a last minute reading to memorize the general
concepts.</p>
                            <a href="dbms.html"><span>START NOW</span></a>
                        </div>
                    </div>
                </div>
                <div class="w-100"></div>
                <!-- categorie -->
                <div class="col-lg-4 col-md-6">
                    <div class="categorie-item">
                        <div class="ci-thumb set-bg" data-
setbg="img/apti.png"></div>
                        <div class="ci-text">
                            <h5>Aptitude</h5>
                            <p>Solve the questions of quant, data
interpretation and logical reasoning. Frequently asked questions in aptitude
exams are given for the practice.</p>
                            <a href="#"><span>START NOW</span></a>
                        </div>
                    </div>
                </div>
                <div class="col-lg-4 col-md-6">
                    <div class="categorie-item">
                        <div class="ci-thumb set-bg" data-
setbg="img/verbal.png"></div>
                        <div class="ci-text">
                            <h5>Verbal</h5>
                            <p>Verbal Ability questions and answers with
explantaions for interview and exams. Fully solved examples of verbal
analogies,synonyms and antonyms. </p>
                            <a href="#"><span>START NOW</span></a>
                        </div>
                    </div>
```



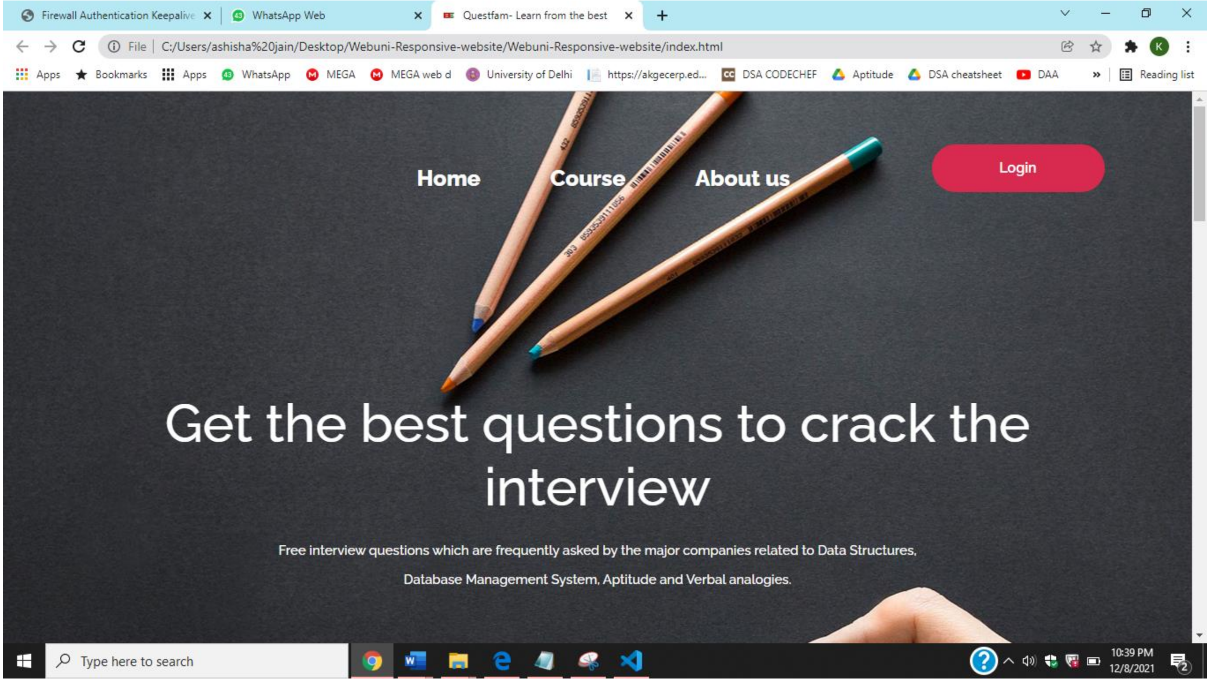
```
            </div>
            </div>
        </div>
    </section>
    <!-- categories section end -->
    <!-- banner section -->
    <section class="banner-section spad">
        <div class="container">
            <div class="section-title mb-0 pb-2">
                <h2>Join Our Community Now!</h2>
            <p>Lets make a change together!</p>
            </div>
            <div class="text-center pt-5">
                <a href="#" class="site-btn">Register Now</a>
            </div>
        </div>
    </section>
    <footer class="footer-section spad pb-0" id="bottom">
        <div class="footer-top">
            <div class="footer-warp">
                <div class="row">
                    <div class="widget-item">
                        <h4>Contact Info</h4>
                        <ul class="contact-list">
                            <li>128/372 Greater Kailash<br>Delhi</li>
                            <li>9854463451</li>
                            <li>yourmail@gmail.com</li>
                        </ul>
                    </div>
                    <div class="widget-item">
                        <h4>Engineering</h4>
                        <ul>
                            <li><a href="">Applied Studies</a></li>
                            <li><a href="">Computer Engineering</a></li>
                            <li><a href="">Software Engineering</a></li>
                            <li><a href="">Informational Engineering</a></li>
                            <li><a href="">System Engineering</a></li>
                        </ul>
                    </div>
                    <div class="widget-item">
                        <h4>Science</h4>
                        <ul>
                            <li><a href="">Applied Studies</a></li>
                            <li><a href="">Computer Engineering</a></li>
                            <li><a href="">Software Engineering</a></li>
                            <li><a href="">Informational Engineering</a></li>
                            <li><a href="">System Engineering</a></li></ul>
```



```
                    </div>
                    <div class="widget-item">
                        <h4>Development</h4>
                        <ul>
                            <li><a href="">Applied Studies</a></li>
                            <li><a href="">Computer Engineering</a></li>
                            <li><a href="">Software Engineering</a></li>
                            <li><a href="">Informational Engineering</a></li>
                            <li><a href="">System Engineering</a></li>
                        </ul>
                    </div>
                    <div class="widget-item">
                        <h4>Newsletter</h4>
                        <form class="footer-newslatter">
                            <input type="email" placeholder="E-mail">
                            <button class="site-btn">Subscribe</button>
                            <p>*We don’t spam</p>
                        </form>
                    </div>
                </div>
            </div>
        </div>
        <div class="footer-bottom">
            <div class="footer-warp">
                <ul class="footer-menu">
                    <li><a href="#">Terms & Conditions</a></li>
                    <li><a href="#">Register</a></li>
                    <li><a href="#">Privacy</a></li>
                </ul>

            </div>
        </div>
    </footer>
    <!-- footer section end -->


    <!--====== Javascripts & Jquery ======-->
    <script src="js/jquery-3.2.1.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/mixitup.min.js"></script>
    <script src="js/circle-progress.min.js"></script>
    <script src="js/owl.carousel.min.js"></script>
    <script src="js/main.js"></script>
</html>
```

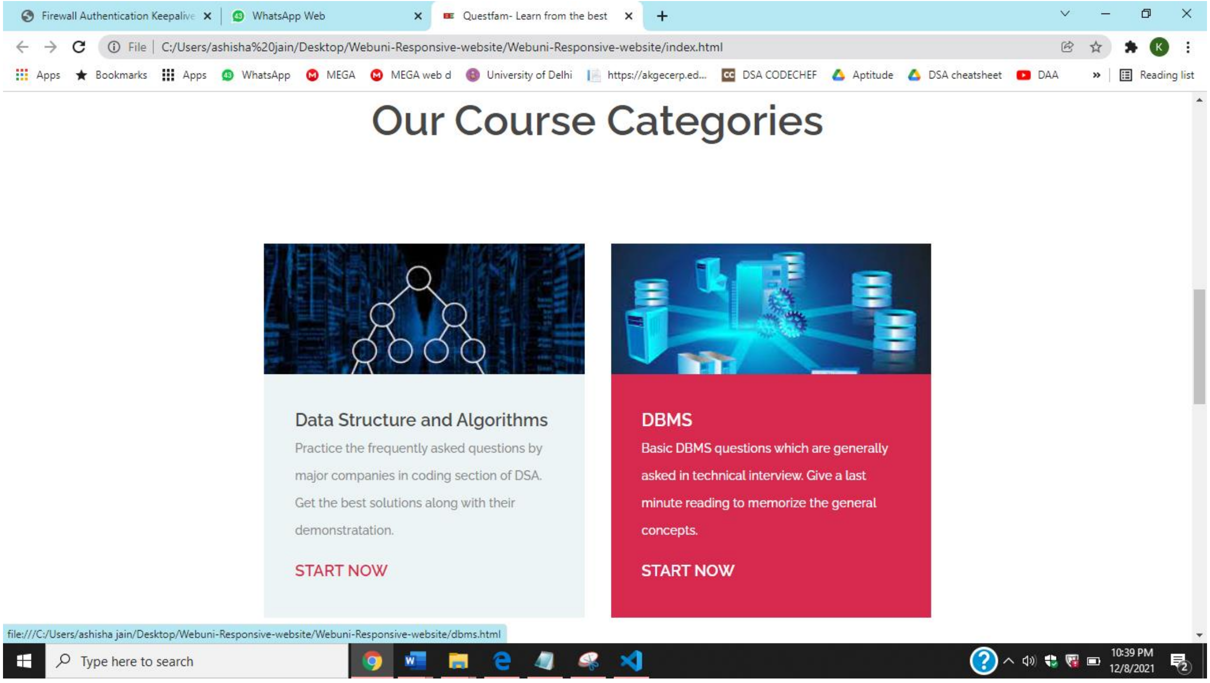
2.7 Snapshot of the project

<u>Home page</u>



Get the best questions to crack the interview

Free interview questions which are frequently asked by the major companies related to Data Structures, Database Management System, Aptitude and Verbal analogies.


Firewall Authentication Keepalive
WhatsApp Web
Questfam- Learn from the best
C:/Users/ashisha%20jain/Desktop/Webuni-Responsive-website/Webuni-Responsive-website/index.html
Apps
Bookmarks
Apps
WhatsApp
MEGA
MEGA web d
University of Delhi
https://akgecerp.ed...
DSA CODECHEF
Aptitude
DSA cheatsheet
DAA
Reading list
START NOW
START NOW
file:///C:/Users/ashisha jain/Desktop/Webuni-Responsive-website/Webuni-Responsive-website/dbms.html
Type here to search
10:39 PM
12/8/2021

Our Course Categories

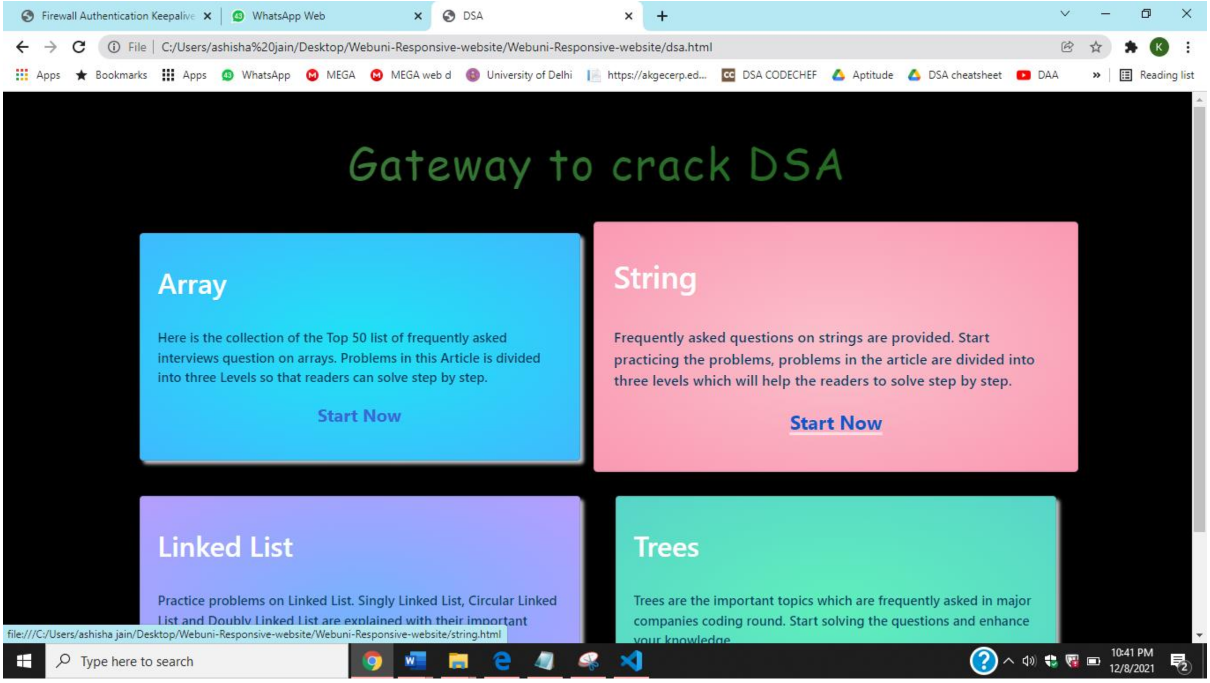
Data Structure and Algorithms

Practice the frequently asked questions by major companies in coding section of DSA. Get the best solutions along with their demonstratation.

DBMS

Basic DBMS questions which are generally asked in technical interview. Give a last minute reading to memorize the general concepts.

Data Structure and algorithm page


Firewall Authentication Keepalive
WhatsApp Web
DSA
File | C:/Users/ashisha%20jain/Desktop/Webuni-Responsive-website/Webuni-Responsive-website/dsa.html
Apps
Bookmarks
Apps
WhatsApp
MEGA
MEGA web d
University of Delhi
https://akgecerp.ed...
DSA CODECHEF
Aptitude
DSA cheatsheet
DAA
Reading list
file:///C:/Users/ashisha jain/Desktop/Webuni-Responsive-website/Webuni-Responsive-website/string.html
Type here to search
10:41 PM
12/8/2021

Gateway to crack DSA

Array

Here is the collection of the Top 50 list of frequently asked interviews question on arrays. Problems in this Article is divided into three Levels so that readers can solve step by step.

Start Now

String

Frequently asked questions on strings are provided. Start practicing the problems, problems in the article are divided into three levels which will help the readers to solve step by step.

Start Now

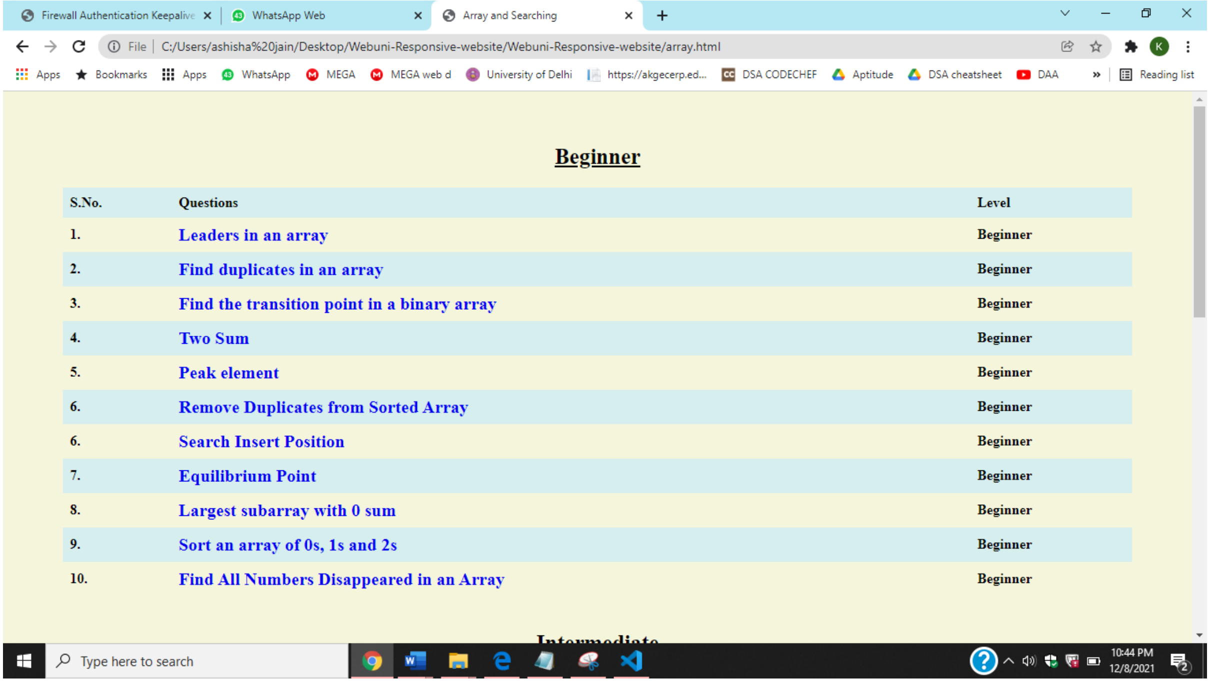
Linked List

Practice problems on Linked List. Singly Linked List, Circular Linked List and Doubly Linked List are explained with their important

Trees

Trees are the important topics which are frequently asked in major companies coding round. Start solving the questions and enhance your knowledge

Array page

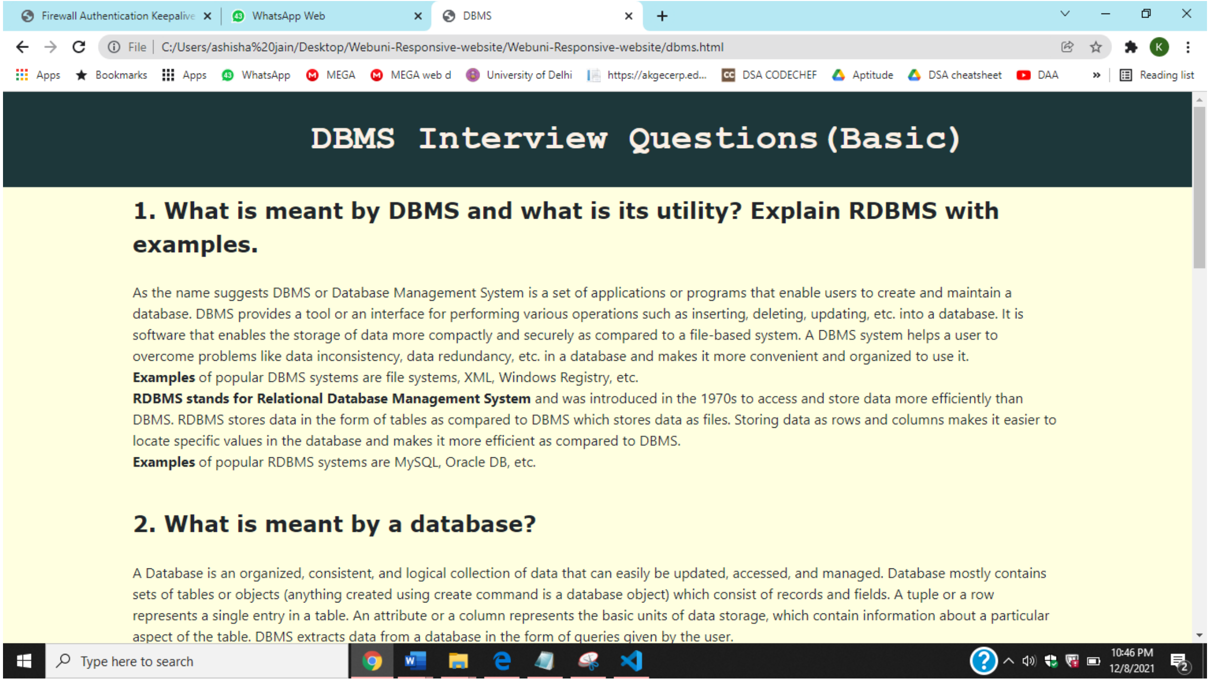


Beginner

S.No.	Questions	Level
1.	Leaders in an array	Beginner
2.	Find duplicates in an array	Beginner
3.	Find the transition point in a binary array	Beginner
4.	Two Sum	Beginner
5.	Peak element	Beginner
6.	Remove Duplicates from Sorted Array	Beginner
6.	Search Insert Position	Beginner
7.	Equilibrium Point	Beginner
8.	Largest subarray with 0 sum	Beginner
9.	Sort an array of 0s, 1s and 2s	Beginner
10.	Find All Numbers Disappeared in an Array	Beginner

Intermediate

DBMS Page



DBMS Interview Questions(Basic)

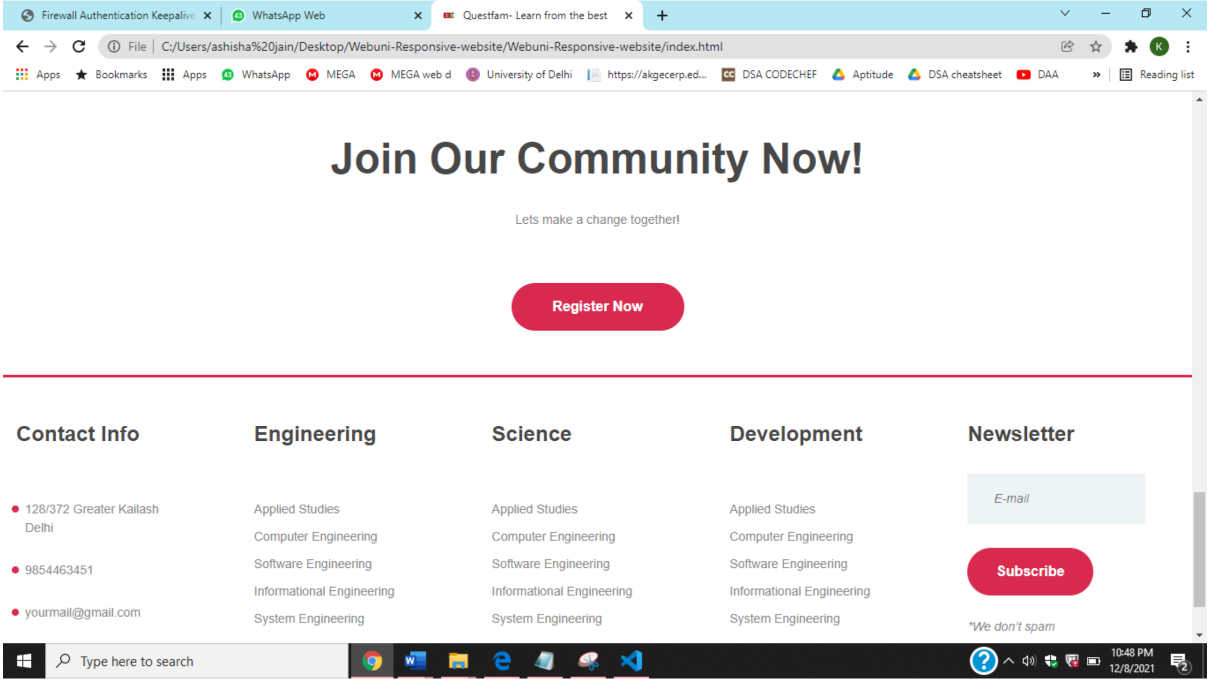
1. What is meant by DBMS and what is its utility? Explain RDBMS with examples.

As the name suggests DBMS or Database Management System is a set of applications or programs that enable users to create and maintain a database. DBMS provides a tool or an interface for performing various operations such as inserting, deleting, updating, etc. into a database. It is software that enables the storage of data more compactly and securely as compared to a file-based system. A DBMS system helps a user to overcome problems like data inconsistency, data redundancy, etc. in a database and makes it more convenient and organized to use it.
Examples of popular DBMS systems are file systems, XML, Windows Registry, etc.
RDBMS stands for Relational Database Management System and was introduced in the 1970s to access and store data more efficiently than DBMS. RDBMS stores data in the form of tables as compared to DBMS which stores data as files. Storing data as rows and columns makes it easier to locate specific values in the database and makes it more efficient as compared to DBMS.
Examples of popular RDBMS systems are MySQL, Oracle DB, etc.

2. What is meant by a database?

A Database is an organized, consistent, and logical collection of data that can easily be updated, accessed, and managed. Database mostly contains sets of tables or objects (anything created using create command is a database object) which consist of records and fields. A tuple or a row represents a single entry in a table. An attribute or a column represents the basic units of data storage, which contain information about a particular aspect of the table. DBMS extracts data from a database in the form of queries given by the user.

Footer of home page



Join Our Community Now!

Lets make a change together!

Contact Info

- 128/372 Greater Kailash Delhi
- 9854463451
- yourmail@gmail.com

Engineering

Applied Studies
Computer Engineering
Software Engineering
Informational Engineering
System Engineering

Science

Applied Studies
Computer Engineering
Software Engineering
Informational Engineering
System Engineering

Development

Applied Studies
Computer Engineering
Software Engineering
Informational Engineering
System Engineering

Newsletter

*We don't spam

CHAPTER:3 Conclusion

3.1 Summary

An interview is a stepping stone to landing a job opportunity of lifetime. Interview is a prominent part of getting jobs nowadays, and to prepare you to compete from the best of best, a good start is required. To gain an insight on this process, we have designed an interview website page, that enlists al questions from important subjects like Data Structure and Algorithm, Data Base Management System, Aptitude and Verbal. These subjects make up the crux of an interview.

We aim to prepare candidates and aspiring students to excel in their respective field by cracking the interview. The questions are prepared keeping in mind the merit, level and area of expertise for the users. The questions have been designated to beginner, intermediate and advanced level to keep the users at par with their level as well as preparing them for the next level.

This website is an ode to all the current as well as soon to be students who have to go through the grueling process of an interview, and have no idea how to go about the preparation of that process. We aim to please the users and provide them with a safe, distraction-free environment for studies. Also aiding in the operation are features of knowing answers of that question on the spot. The user doesn’t have to wander out in search of answers and waste their time.

The panels shown in the website allow smooth transition and help the users to navigate easily throughout the website as well as indicate the respective subjective fields. The user is also given the option of going back to the previous or home page whenever required. There are also additional features of ‘Sign up’ and ‘Register’. The technologies used in the website are HTML, CSS, React, JQuery and Bootstrap. We hope that this website will be an easy and convenient platform for users for interview preparation.