

Building a Personal Blog Website with Cloud Computing



HTML WEB CREATE CODE

```
File Edit Selection View Go Run Terminal Help CAD
index.html X app.py # style.css
templates > index.html > html > head > meta
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Travel blog</title>
8   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}>
9 </head>
10
11 <body>
12   <!-- <h1>Welcome to the home page</h1> -->
13   <!-- header starts -->
14   <header>
15     <!-- navbar starts -->
16     <nav class="navbar">
17       
18       <div class="logo">
19         <h1>TRAVEL-BLOG </h1>
20       </div>
21       <!-- <span><img src="" alt=""></span> -->
22       <ul class="links">
23         <li><a href="#">ABOUT</a></li>
24         <li><a href="#">BLOG</a></li>
25         <li><a href="#">TRAVEL TIPS</a></li>
26         <li><a href="#">DESTINATIONS</a></li>
27         <li><a href="#">CONTACT</a></li>
28       </ul>
29
30     </nav>
31     <!-- navbar ends -->
32   </header>
33
34   <!-- header ends -->
35   <!-- home section -->
36   <div class="home">
37   </div>
38 </body>
```

Index.html

Share Code Link Blackbox Search Terminal Output

Ln 5, Col 27 Spaces: 4 UTF-8 CRLF HTML Port: 5500 Blackbox

CASCADING STYLE SHEETS

The screenshot shows a code editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. The title bar displays the file path index.html > # style.css and the file name # style.css. The main editor area contains the following CSS code:

```
static > css > # style.css >
107 header nav ul li a{
108     border-bottom-right-radius: 20px;
109     padding: 5px;
110     font-size: 2em;
111     text-decoration: none;
112     color: #rgb(0, 0, 0);
113 }
114
115 }
116
117 header nav ul li a:hover{
118     background-color: #rgba(225, 231, 229, 0.096);
119 }
120
121
122 /* home section */
123
124 .home{
125     background-color: #aliceblue;
126     height: 90vh;
127     width: 100vw;
128     background-image: url("#{url_for('static', filename='images/bg.jpg')}");
129     /* background-repeat: no-repeat;
130     background-size: cover; */
131
132     /* background-color:rgba(0, 0, 0, 0.714); */
133 }
134
135
136
137 /* responsive */
138 /* ----- */
139
140 @media screen and (max-width: 800px)
141 {
142     header nav ul li{
143         display: none;
144     }
145 }
```

The bottom status bar shows file statistics: Ln 123, Col 8, Spaces: 4, UTF-8, CRLF, CSS, Port: 5500, Blackbox, and a refresh icon.

File Edit Selection View Go Run Terminal Help CAD

```
index.html app.py # style.css
```

```
static > css > # style.css > ...
107 header nav ul li a{
108     border-bottom-right-radius: 20px;
109     padding:5px;
110     font-size:2vmin;
111     text-decoration: none;
112     color:rgb(0, 0, 0);
113 }
114 }
115 }
116
117 header nav ul li a:hover{
118     background-color: rgba(225, 231, 229, 0.096);
119 }
120 }
121
122 /* home section */
123
124 .home{
125     background-color: aliceblue;
126     height:90vh;
127     width:100vw;
128     background-image: url("{{url_for('static',filename='images/bg.jpg')}}");
129     /* background-repeat: no-repeat;
130     background-size: cover; */
131
132     /* background-color:rgba(0, 0, 0, 0.714) ; */
133 }
134 }
135
136
137 /* responsive */
138 /* ----- */
139
140 @media screen and (max-width: 800px)
141 {
142
143     header nav ul li{
144         display: none;
145     }
146 }
```

Share Code Link Blackbox Search Terminal Output

In 123, Col 8 Spaces: 4 UTF-8 CRLF CSS Port: 5500 Blackbox

```
File Edit Selection View Go Run Terminal Help CAD □ □ □ □ ×
```

```
index.html app.py # style.css
```

```
static > css > # style.css > ...
```

```
74     position: relative;
75 }
76
77 header nav .menu-btn{
78     position: absolute;
79     width: 0;
80     right: 2%;
81     cursor: pointer;
82 }
83
84 /* ul */
85 header nav ul{
86
87     height: 100%;
88     width: 70%;  
/* background-color: grey; */  
     display: flex;  
     /* margin-right: 10px; */  
     justify-content: space-around;  
     align-items: center;
89 }
90
91 header nav ul li{
92     /* display: none; */  
     list-style-type: none;
93 }
94
95 header nav ul li a{
96     border-bottom-right-radius: 20px;  
     padding: 5px;  
     font-size: 2vmin;  
     text-decoration: none;  
     color: □rgb(0, 0, 0);
97 }
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
```

In 123, Col 8 Spaces: 4 UTF-8 CRLF CSS ⚙ Port: 5500 Blackbox

The screenshot shows a code editor interface with a dark theme. The top bar includes tabs for 'index.html', 'app.py', and '# style.css'. A tooltip is displayed over the line 'box-sizing: border-box;', providing information about the 'width' and 'height' properties and linking to the MDN Reference. The code in 'style.css' defines a header with a height of 10vh and a logo section with flexbox styling.

```
static > css >
1  /*<
2
3  *{
4
5    Syntax: content-box | border-box
6
7    MDN Reference
8
9    box-sizing: border-box;
10
11 /* header */
12 header{
13
14   font-size: 2.5vmin;
15   width:100vw;
16   height: 10vh;
17   /* background-color: #fff; */
18
19   /* padding:0 10px; */
20
21
22 }
23
24
25 .logo{
26   /* color:rgb(0, 255, 110); */
27   display: flex;
28   justify-content:start;
29   align-items:center;
30   height:100%;
31   width:30%;
32   cursor: pointer;
33   color:■rgb(255, 255, 255);
34   margin-left:2vmin;
35
36   /* background-color: yellow; */
37 }
38
39 .logo h1{
```

Ln 123, Col 8 Spaces: 4 UTF-8 CRLF CSS Port: 5500 Blackbox

```
File Edit Selection View Go Run Terminal Help CAD index.html app.py # style.css
static > css > # style.css > ...
130     /* background-repeat: no-repeat;
131     background-size: cover; */
132
133     /* background-color:rgba(0, 0, 0, 0.714); */
134 }
135
136
137 /* responsive */
138 /* ----- */
139
140 @media screen and (max-width: 800px)
141 {
142
143     header nav ul li{
144         display: none;
145     }
146
147     header nav ul{
148         width:0;
149     }
150
151     header nav .logo{
152         width:100%;
153         justify-content: center;
154     }
155
156     header nav .menu-btn{
157         width:5vmin;
158     }
159
160 }
161
162
163
164
165
166
167
168
169     /* </style> */
```

Ln 123, Col 8 Spaces: 4 UTF-8 CRLF CSS ⚙ Port: 5500 Blackbox

FLASK CODE

The screenshot shows a code editor interface with a dark theme. The title bar reads "FLASK CODE". The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and CAD. The toolbar has icons for file operations like Open, Save, and Print, along with a search icon.

The left sidebar contains project files: index.html, app.py, and style.css. The main editor area displays the following Python code:

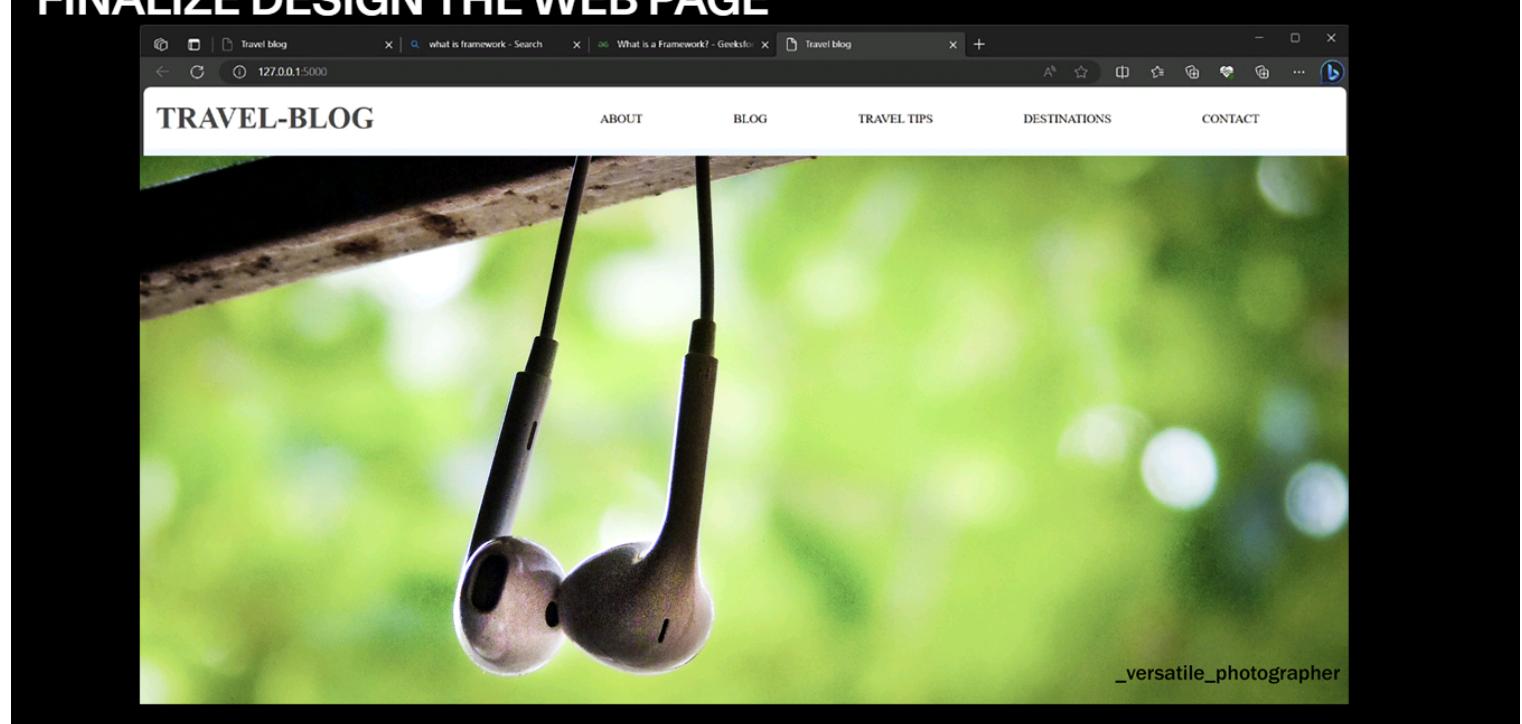
```
index.html
app.py > ...
# style.css

from flask import Flask, render_template
# Create a Flask application instance
app = Flask(__name__)
# Define a route and the corresponding function to handle it
@app.route('/')
@app.route('/home')
def hello_world():
    return render_template('index.html')

# Run the Flask app
if __name__ == '__main__':
    app.run(debug=True)
```

The status bar at the bottom shows: Line 18, Col 1 | Spaces: 4 | UTF-8 | CRLF | Python 3.12.0 (venv: venv) | Port: 5500 | Blackbox |

FINALIZE DESIGN THE WEB PAGE



Introduction to Cloud Computing

What is Cloud Computing?

Cloud computing is the delivery of computing services, including servers, storage, databases, networking, software, analytics, and intelligence, over the internet (the cloud) to offer faster innovation, flexible resources, and economies of scale. In this section, we will explore the fundamentals of cloud computing and its benefits.

Benefits of Cloud Computing

- Scalability and Flexibility
- Cost Savings
- Reliability and Availability
- Increased Collaboration



Considerations for Choosing a Cloud Provider

When choosing a cloud provider, there are several factors to consider. These include:

- Cost and pricing model
- Reliability and uptime guarantees
- Security and compliance
- Scalability and flexibility
- Ease of use and management

Popular Cloud Providers

Some of the most popular cloud providers include:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)
- IBM Cloud
- Oracle Cloud Infrastructure (OCI)

Setting Up a Development Environment

Before you can start creating your website, you need to set up a development environment. This will allow you to write and test your code locally before deploying it to the cloud.

Choose Your Text Editor

The first step is to choose a text editor. There are many options available, both free and paid. Some popular choices include:

- Visual Studio Code
- Sublime Text
- Atom

Choose the one that works best for you and download it onto your computer.

Install a Local Server Environment

Next, you'll need to install a local server environment. This will allow you to run your website on your own computer. Some popular options include:

- XAMPP (for Windows, Mac, and Linux)
- MAMP (for Mac)

Download and install the one that is appropriate for your operating system. Once installed, start the local server environment to ensure it is running properly.

Creating a Basic HTML Page

HTML (Hypertext Markup Language) is the standard markup language used to create web pages. In this section, we will cover the basics of creating a simple HTML page.

Step 1: Create a New File

The first step is to create a new file on your computer. Open a text editor such as Notepad or Sublime Text and create a new file.

Step 2: Add the HTML Boilerplate

The HTML boilerplate is the basic structure of an HTML page. It includes the doctype declaration, the HTML element, head element, and body element.

```
<!DOCTYPE html>

<html>
  <head>  <title>My First HTML Page</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

This code will create a basic HTML page with a title of 'My First HTML Page' and a heading that says 'Hello, World!'.

Step 3: Save and View Your Page

Save your file with a .html extension, such as 'index.html'. Open the file in your web browser to view your new HTML page.

Designing with CSS

CSS is a powerful tool for designing the look and feel of your website. It allows you to control the layout, fonts, colors, and other visual elements of your pages. Here are some tips for using CSS effectively:

- Use a consistent color scheme and typography to create a cohesive design.
- Keep your code organized and easy to read by using classes and IDs to target specific elements.
- Use responsive design techniques to ensure that your site looks great on all devices.



Adding Navigation

Navigation is an essential component of any website, helping users to easily navigate to different pages and sections. In this section, we will learn how to add navigation to our website using HTML and CSS.

Creating a Navigation Bar

The first step in adding navigation is to create a navigation bar. We can do this by using an unordered list (``) and styling it with CSS to create a horizontal bar. Here's an example:

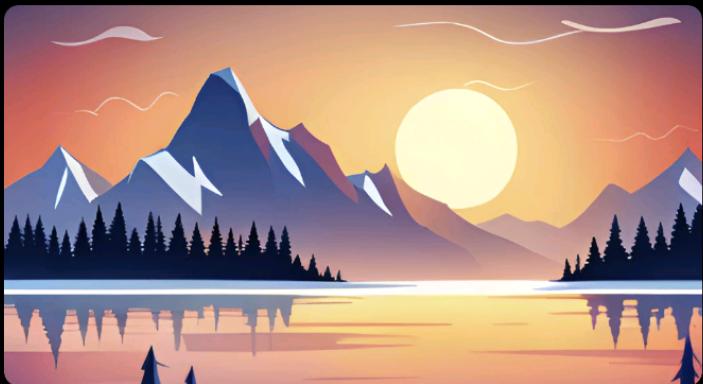
In this example, we have created a `<nav>` element to contain our navigation bar. Inside the `<nav>` element, we have an unordered list (``) with four list items (``), each containing a link (`<a>`) to a different page on our website. We have also added an ID attribute to each list item to style them individually with CSS.

Styling the Navigation Bar

Now that we have created our navigation bar, we can style it with CSS to make it look more visually appealing and user-friendly. Here's an example of some CSS styles we can use:

In this example, we have used CSS to style our navigation bar with a dark background color, white text, and a flexbox layout. We have also added some margin to our list items to give them some breathing room. Feel free to experiment with different styles to create a navigation bar that fits your website's aesthetic.

Creating a Blog Post Template



Layout

A blog post template should have a clear and consistent layout to make it easy for readers to navigate. The template should include a featured image, title, author, date, and social sharing buttons. Additional elements such as a header image, related posts, and a comments

Typography

Choosing the right typography is important for readability and aesthetic appeal. A simple and legible font should be used for the body text, while a more decorative font can be used for headings and subheadings. The font size and line spacing should be adjusted for

Color Scheme

The color scheme should be consistent with the overall design of the website. A neutral background color can be used for the body of the post, while brighter colors can be used for headings and other elements. The color scheme should be chosen to enhance readability

Adding Dynamic Content with JavaScript



What is Dynamic Content?

Dynamic content refers to website elements that change based on user input or other factors. This can include things like pop-up messages, interactive forms, and live updates of data.

Using JavaScript for Dynamic Content

JavaScript is a popular programming language used for creating dynamic content on the web. It allows developers to create interactive elements that respond to user input and update in real-time.

Integrating with a CMS

A content management system (CMS) can greatly simplify the process of managing and publishing content on your website. There are many CMS options available, such as WordPress, Drupal, and Joomla. Integrating a CMS into your cloud computing personal blog website can provide a user-friendly interface for creating and editing content, as well as a range of additional features.



Choosing a CMS

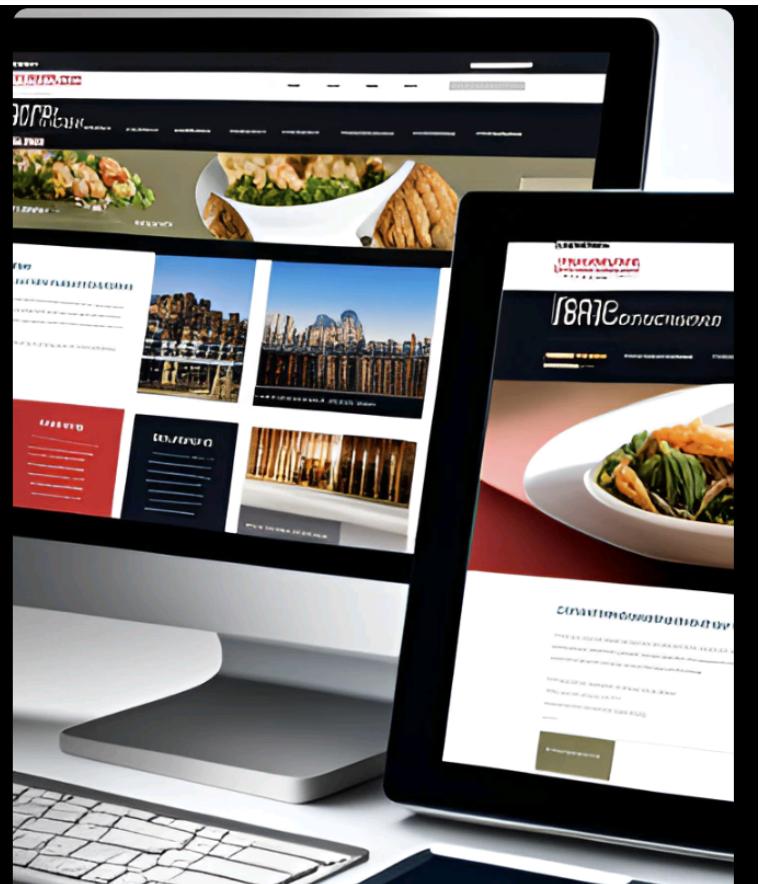
When choosing a CMS, consider factors such as ease of use, customization options, and available plugins and themes. It's also important to ensure that the CMS you choose is compatible with your cloud provider and development environment.

Implementing Responsive Design

In today's world, where users access websites from a variety of devices with different screen sizes, it is essential to ensure that your website looks good and functions well on all devices. Responsive design is the approach that allows us to achieve this goal. With responsive design, we can create a website that automatically adjusts its layout and content to fit the screen size of the device being used to access it.

To implement responsive design, we use a combination of HTML, CSS, and JavaScript. We start by designing a flexible layout that can adapt to different screen sizes. We then use CSS media queries to apply different styles to the layout based on the screen size of the device. Finally, we use JavaScript to add interactivity and dynamic features to the website.

- Design a flexible layout that can adapt to different screen sizes.
- Use CSS media queries to apply different styles to the layout based on the screen size of the device.



Adding Social Sharing Features

Social sharing features allow your readers to easily share your content on their favorite social media platforms. This can help increase your reach and drive more traffic to your website.

Choosing Which Platforms to Include

When deciding which social media platforms to include, consider your target audience and where they are most active. Some popular options include:

- Facebook
- Twitter
- LinkedIn
- Pinterest
- Instagram



Implementing SEO Best Practices



Keyword Research

Start by researching relevant keywords and phrases to target in your content. Use tools like Google Keyword Planner or SEMrush to find popular search terms related to your topic.



On-Page Optimization

Ensure that your website's content is optimized for search engines. This includes using relevant keywords in page titles, meta descriptions, and header tags, as well as optimizing images and videos with alt text.



Analytics and Monitoring

Use tools like Google Analytics to track your website's traffic and monitor keyword rankings. Analyze this data to make informed decisions about future content and optimization strategies.

Setting Up Analytics

Analytics are a crucial component of any website, allowing you to track user behavior and gain insights into how your site is performing. By setting up analytics on your cloud computing personal blog website, you can monitor traffic, track user engagement, and identify areas for improvement.

Choosing an Analytics Provider

There are many analytics providers to choose from, each with its own strengths and weaknesses. Some popular options include Google Analytics, Adobe Analytics, and Piwik. When choosing an analytics provider, consider factors such as cost, ease of use, and the specific features that are important to you.



Setting Up Analytics

Analytics are a crucial component of any website, allowing you to track user behavior and gain insights into how your site is performing. By setting up analytics on your cloud computing personal blog website, you can monitor traffic, track user engagement, and identify areas for improvement.

Choosing an Analytics Provider

There are many analytics providers to choose from, each with its own strengths and weaknesses. Some popular options include Google Analytics, Adobe Analytics, and Piwik. When choosing an analytics provider, consider factors such as cost, ease of use, and the specific features that are important to you.

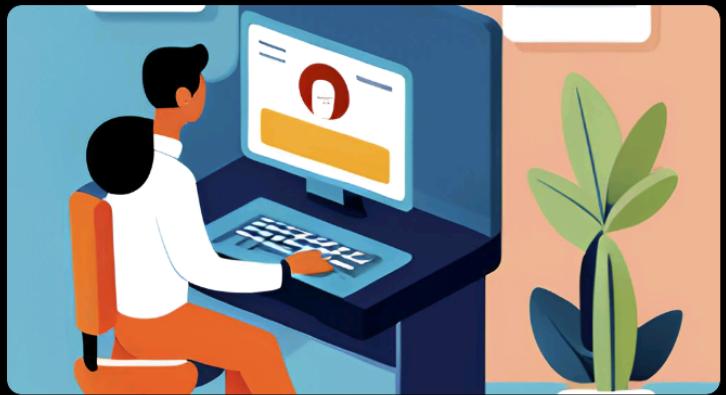


Implementing User Authentication



User authentication is an essential feature for any website that requires user accounts. It allows users to securely log in and access their account information, as well as perform actions that require authentication, such as making purchases or submitting comments.

To implement user authentication,



Password Security

It is important to ensure that user passwords are stored securely in the database. This can be done by using a hashing algorithm such as bcrypt to encrypt the password before storing it. It is also a good practice to enforce password complexity requirements, such as a

Session Management

Once a user has logged in, their session needs to be managed to ensure that they remain authenticated as they navigate the site. This can be done by setting a session cookie with a unique identifier for the user, and checking this identifier on subsequent

Creating a Commenting System

Why Add a Commenting System?

A commenting system allows your readers to engage with your content, share their thoughts and opinions, and start a conversation. This can help to build a community around your blog and increase user engagement.

Choosing a Commenting System

There are many third-party commenting systems available, such as Disqus and Facebook Comments. These systems integrate easily with your website and offer features like spam protection and moderation tools.

Creating a Custom Commenting System

If you prefer to create your own commenting system, you can use a server-side language like PHP and a database like MySQL to store and retrieve comments. You will also need to implement user authentication and moderation tools to prevent spam and inappropriate content.