Web Technology
Assessment Solution
2023 Fall
NCIT

Sanjaya (Bir Bikram) Shrestha

February 27, 2023

Contents

11 (Q No.1 A) To make an online presence or just transfer any data there are certain predefined rules to follow. Explain any 5 such commonly used protocols that serve different purposes of their own. **16**
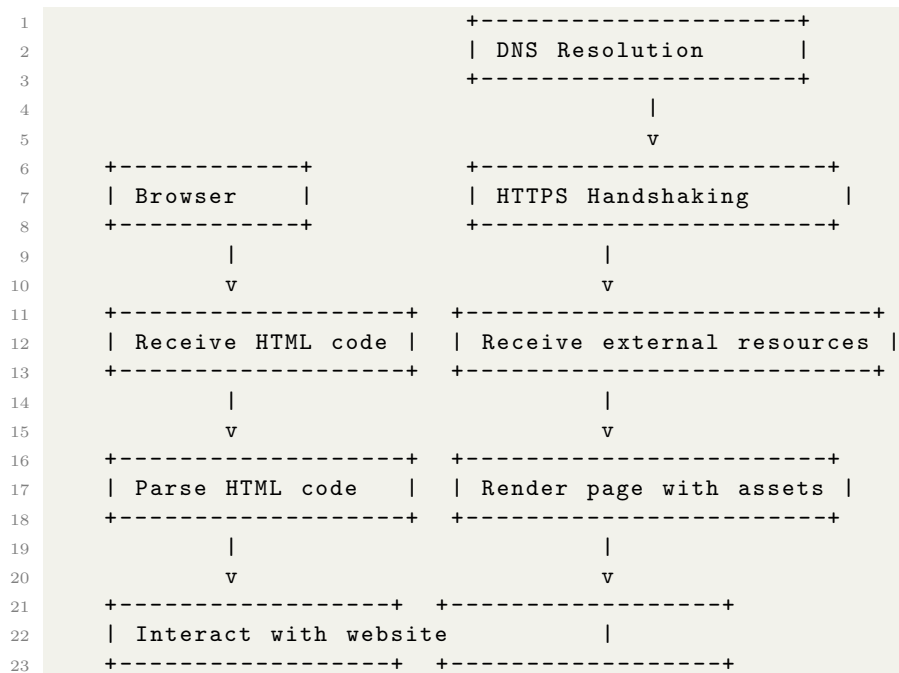
1 (Q No.1 A) To make an online presence or just transfer any data there are certain predefined rules to follow. Explain any 5 such commonly used protocols that serve different purposes of their own.
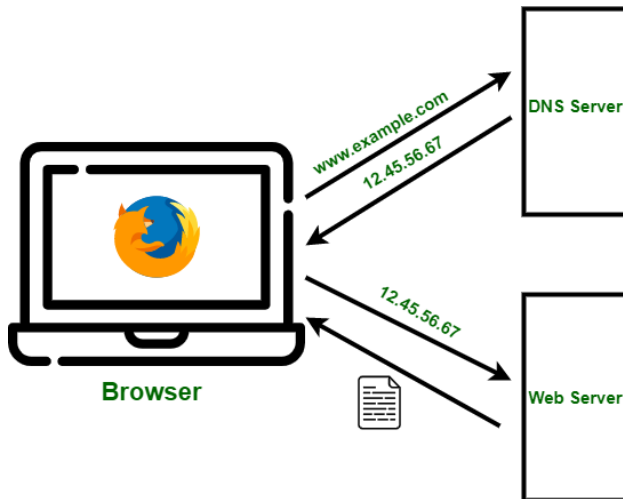
1. **TCP/IP: (Transmission Control Protocol/Internet Protocol)** This is the foundation protocol of the internet and is used for communication between devices over the internet. The web browser sends a request to the website's server using the TCP/IP protocol and then server sends the requested web page back to the web browser using TCP/IP as well.

2. **HTTP: (Hyper Text Transfer Protocol)** This is the protocol used for web browsing and transferring web pages from servers to clients. The browser sends http request to the server hosting the website. The server responds with an HTTP response, which includes the requested web page.

3. **FTP: File transfer protocol** This is the protocol used for file transfer over the internet. When we need to transfer a large file from one computer to another over the internet, you can use the FTP protocol. We need an FTP client program to connect to the FTP server where the file is stored, and then we can download the file using the FTP protocol.

4. **SMTP: (Simple Mail Transfer Protocol)** This is the protocol used for email transmission between servers and clients. We are sending an email from your email client program to someone else's email address. Our email client uses the SMTP protocol to send the email to the email provider's SMTP server. The SMTP server then relays the email to the recipient's email provider using SMTP as well.

5. **DNS: (Domain Name System)** This is the protocol used for translating domain names into IP addresses. When we type in a website URL into your web browser's address bar, the browser sends a request to a DNS server to look up the IP address associated with that URL. The DNS server responds with the IP address, which the web browser then uses to connect to the website's server.

6. **SSH: (Secure Shell)** This is the protocol used for secure remote login and file transfer. If you need to remotely access a computer or server securely, you can use the SSH protocol. You'll need an SSH client program to connect to the remote computer or server, and then you can securely log in and transfer files using SSH.

7. **DHCP: (Dynamic Host Control Protocol)** This is the protocol used for automatically assigning IP addresses to devices on a network. When a device joins a network, it can use DHCP to automatically obtain an IP address from the network's DHCP server. This allows devices to join a network without needing to manually configure IP addresses.

8. **POP: (Post Office Protocol)** This is the protocol used for retrieving email from a server. We using an email client program like Microsoft Outlook or Apple Mail to download our email, we're likely using the POP protocol. POP allows email client to download all of your email from email provider's server and store it on local computer.

2 (Q No.1B) What happens when you enter https://ncit.edu.np into your browser? Describe along with its block diagram.

When we type https://ncit.edu.np into our browser following steps will happen:

1. Browser will initiates a request to the Domain Name System (DNS) to resolve the domain name "https://ncit.edu.np" into an IP address.

2. The DNS responds with the IP address associated with the domain name, which in this case is the IP address of the server hosting the website.

3. Browser will initiates a secure connection with the server using HTTPS protocol, which includes a series of handshaking and authentication steps to ensure the connection is secure and encrypted.

4. The server responds with the HTML code for the home page of the pu.edu.np website.

5. Browser then receives the HTML code and begins to parse it to render the page.

6. The browser sends additional requests for any external files referenced in the HTML code, such as CSS stylesheets, images, and scripts.

7. The server responds with the requested files and the browser uses them to render the page as intended.

8. Then if we interact with the website, such as clicking a link or submitting a form, the browser sends additional requests to the server and the server responds with the appropriate content.

```
1                                    +---------------------+
2                                    | DNS Resolution      |
3                                    +---------------------+
4                                               |
5                                               v
6    +------------+          +-----------------------+
7    | Browser    |          | HTTPS Handshaking     |
8    +------------+          +-----------------------+
9          |                            |
10         v                            v
11   +------------------+   +---------------------------+
12   | Receive HTML code |  | Receive external resources |
13   +------------------+   +---------------------------+
14         |                            |
15         v                            v
16   +------------------+   +-----------------------+
17   | Parse HTML code  |   | Render page with assets |
18   +------------------+   +-----------------------+
19         |                            |
20         v                            v
21   +------------------+   +------------------+
22   | Interact with website            |
23   +------------------+   +------------------+
```

3   (Q No.2A) Write HTML code for the following


```html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8     <style>
9         input {
10             margin: 5px;
11         }
12         fieldset {
```

```
13              border: 1px dashed black;
14          }
15
16      </style>
17  </head>
18  <body>
19      <form action="">
20          <fieldset>
21              <legend align="center">Registration</legend>
22              <fieldset>
23                  <legend align="center">Personal Info</legend>
24                  <label for="firstname">Firstname</label>
25                  <input type="text" name="firstname" id="firstname">
26
27                  <label for="lastname">Lastname</label>
28                  <input type="text" name="lastname" id="lastname">
29
30                  <label for="age">Age</label>
31                  <input type="number" name="age" id="age"> <br>
32
33                  <label for="email">Email</label>
34                  <input type="email" name="email" id="email">
35
36                  <label for="phone_number">Phone </label>
37                  <select name="phone_code" id="phone_code">
38                      <option value="nep" selected>+977</option>
39                      <option value="india">+91</option>
40                      <option value="us">+1</option>
41                  </select>
42                  <input type="phone_number" name="phone_number" id="phone_number">
    <br>
43
44                  <label for="address">Address </label>
45                  <select name="address" id="address">
46                      <option value="ktm" selected>KTM</option>
47                      <option value="pokhara">Pokhara</option>
48                      <option value="butwal">Butwal</option>
49                  </select>
50
51                  <label for="gender">Gender</label>
52                  <input type="radio" name="gender" id="male" value="male"> Male
53                  <input type="radio" name="gender" id="female" value="female">
    Female
54                  <input type="radio" name="gender" id="other" value="other"> Other
55
56
57              </fieldset>
58
59              <fieldset>
60                  <legend align="center">Credential</legend>
61
62                  <label for="username">Username</label>
63                  <input type="text" name="username" id="username"> <br>
64
65                  <label for="password">Password</label>
66                  <input type="text" name="password" id="password">
67
68              </fieldset>
69
```

```
70            <fieldset>
71                <legend align="center">Interest</legend>
72
73                <label for="interest">Hobby</label>
74                <input type="checkbox" name="interest" id="interest" value="
      cricket"> Cricket
75                <input type="checkbox" name="interest" id="interest" value="
      football"> Football
76                <input type="checkbox" name="interest" id="interest" value="movie"
      > Movie
77
78            </fieldset>
79
80            <label for="faculty">Faculty</label>
81            <input type="checkbox" name="faculty" id="faculty" value="it"> IT
82            <input type="checkbox" name="faculty" id="faculty" value="bca"> BCA
83            <input type="checkbox" name="faculty" id="faculty" value="engineering"
      > Engineering <br>
84
85            <label for="university">University</label>
86            <input type="text" name="university" id="university"><br>
87
88            <label for="photo">Photo</label>
89            <input type="file" name="photo" id="photo" accept="image/png"><br>
90
91            <label for="transcripts">Transcripts</label>
92            <input type="file" name="transcripts" id="transcripts" accept="image/
      png"> <br>
93
94            <label for="message">Message</label>
95            <textarea name="message" id="message" cols="40" rows="15" placeholder=
      "Type your message here"></textarea>
96 <br>
97            <input type="submit" value="Submit">
98            <input type="reset" value="Reset">
99            <input type="button" value="Ok">
100       </fieldset>
101   </form>
102 </body>
103 </html>
```

4 (Q No.2B) How can you map images in HTML? Does a hyperlink apply to text only? Explain the logic behind it and its types in detail.

To map an image in HTML, you can use the <map> and <area> elements to define clickable regions on the image. Here's an example:

```
1   <img src="https://fakeimg.pl/300/" alt="My Image" usemap="#myMap">
2
3   <map name="myMap">
4   <area shape="rect" coords="0,0,100,100" href="page1.html" alt="Page 1">
5   <area shape="rect" coords="100,0,200,100" href="page2.html" alt="Page 2">
```

```
6    <area shape="circle" coords="150,150,50" href="page3.html" alt="Page 3">
7    </map>
```

Listing 1: Image map

When the user clicks on one of the mapped areas, the browser will navigate to the URL specified in the "href" attribute. This allows you to create interactive images that can act as navigation menus, image maps, and more.

No, hyperlinks in HTML do not apply to text only. We can create hyperlinks for any element that can have a clickable area, including images, buttons, and even entire divs.

A hyperlink is created using an ¡a¿ tag, which stands for "anchor." The <a> tag requires a "href" attribute, which specifies the URL that the link should point to. We can also include text or other elements within the <a> tag, which will serve as the clickable area for the hyperlink.

Here's an example of how to create a hyperlink for text in HTML:

```
1    <a href="https://example.com">Click here to visit Example.com</a>
```

For email links also we can use the hyperlink. Email links are created using the mailto: protocol, like this:

```
1    <a href="mailto:example@example.com">Send an email to Example</a>
```

For images we can use hyperlink like this:

```
1    <a href="www.google.com"><img src="./google.png" alt="Click here to go to
     google"> </a>
```

Any clickable content inside the <a> tag works for hyperlink.

5   (Q No.3A) What are the overriding rule in CSS? Explain different levels of stylesheets with their hierarchy of implementation.

In CSS, the overriding rule is used to determine which style declarations should be applied to an element when there are multiple rules that could potentially apply.

The overriding rule in CSS can be summarized as follows:

1. Styles defined using the !important keyword take precedence over all other styles, regardless of where they are defined in the CSS file.

2. Styles defined inline (i.e. using the style attribute in an HTML element) take precedence over styles defined in an external CSS file or in the <head> section of the HTML document.

3. Styles defined closer to the element in the HTML document take precedence over styles defined further away. For example, styles defined in an ID selector take precedence over styles defined in a class selector, and styles defined in a class selector take precedence over styles defined in a tag selector.

4. Styles defined with more specific selectors take precedence over styles defined with less specific selectors. For example, a style defined with a selector of #my-div p (which targets all <p> elements within an element with the ID of my-div) would take precedence over a style defined with a selector of p (which targets all <p> elements on the page).

5. Styles defined later in the CSS file take precedence over styles defined earlier in the file, provided that all other factors are equal.

By: Sanjaya (Bir Bikram) Shrestha

There are 3 levels of stylesheets in CSS:

1. Inline Styles: These styles are applied directly to an HTML element using the style attribute. Inline styles have the highest specificity, meaning they override any other styles applied to the same element.

```
1        <p style="color: red;">This text is red.</p>
2
```

Listing 2: Inline Style

2. Internal Styles: These styles are defined within the head section of an HTML document using the style tag. Internal styles have a higher specificity than external styles, but lower than inline styles.

```
1
2        <head>
3    <style>
4        p {
5        color: blue;
6        }
7    </style>
8    </head>
9    <body>
10   <p>This text is blue.</p>
11   </body>
12
```

Listing 3: Internal Style

3. External Styles: These styles are defined in an external CSS file and are linked to an HTML document using the link tag. External styles have the lowest specificity, but they are also the most reusable and easier to maintain.

```
1
2           <head>
3       <link rel="stylesheet" href="style.css">
4       </head>
5       <body>
6       <p class="red-text">This text is red.</p>
7       </body>
8
```

Listing 4: External Style

6 (Q No.3B) Design a box model. What will be the actual height and width of content if padding=margin=border=height=width of content = 20px?

The CSS box model is a container that contains multiple properties including borders, margin, padding, and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model. Box-Model has multiple properties

By: Sanjaya (Bir Bikram) Shrestha

in CSS.

```
┌─────────────────────────────────────┐
│              margin                  │
│  ┌───────────────────────────────┐   │
│  │            border             │   │
│  │  ┌─────────────────────────┐  │   │
│  │  │         padding         │  │   │
│  │  │   ┌─────────────────┐   │  │   │
│  │  │   │     content     │   │  │   │
│  │  │   └─────────────────┘   │  │   │
│  │  └─────────────────────────┘  │   │
│  └───────────────────────────────┘   │
└─────────────────────────────────────┘
```
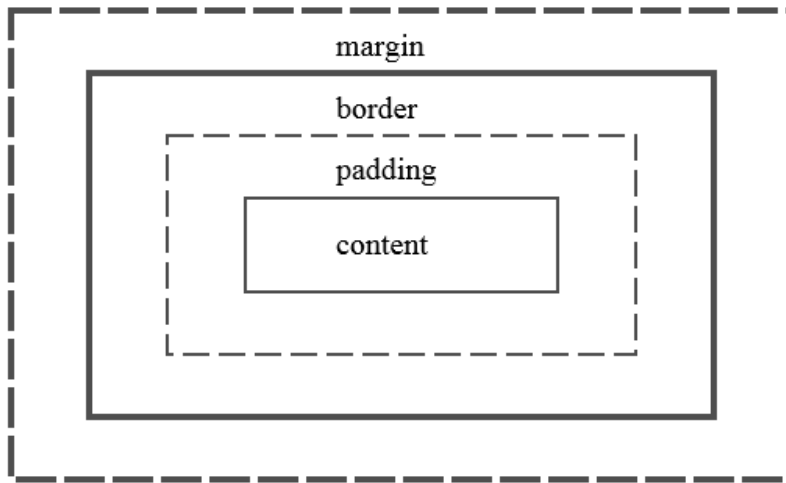
If the padding, margin, border, height, and width of an element's content are all set to 20 pixels, then the actual height and width of the content will be 0 pixels.

This is because the total size of the element would be calculated as follows:

Total size = height + padding + border + margin

Total size = 20px + 20px + 20px + 20px

Total size = 80px

Since the total size is greater than the height and width of the content (which are both set to 20 pixels), there is no space left over for the content itself. As a result, the content will be completely hidden and will not be visible on the page.

By: Sanjaya (Bir Bikram) Shrestha

7 (Q No.C) Write HTML and CSS code for the following O/P


```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <style>
9
10 header {
11     padding: 30px;
12     background-color: gray;
13     text-align: center;
14     font-size: 22pt;
15     color: white;
16 }
17
18 nav {
19     margin-top: 2px;
20     background-color: gray;
21     padding: 2px;
22     display: block;
23 }
24
25 li {
26     list-style-type: none;
27     display: inline;
28     margin: 10px;
29 }
30 li a {
31     text-decoration: none;
32     color: white;
33 }
34
35 li input[type=search] {
```

By: Sanjaya (Bir Bikram) Shrestha

```
36        float: right;
37        padding: 6px;
38        border: none;
39        }
40
41        .wrapper {
42            display: flex;
43        }
44
45        .box {
46            /* height: 200px; */
47            width: 100%;
48            background-color: beige;
49            margin-left: 2px;
50            text-align: center;
51            padding: 100px 0;
52        }
53
54        footer {
55            text-align: center;
56            background-color: gray;
57            padding: 20px;
58            color: white;
59            font-size: 16pt;
60        }
61        </style>
62 </head>
63 <body>
64     <header>NCIT COLLEGE</header>
65     <nav>
66
67        <ul>
68            <li><a href="#home">Home</a></li>
69            <li><a href="#contact">Contact</a></li>
70            <li><a href="#about">About</a></li>
71            <li><input type="search" name="search" id="search" placeholder="Search
    "></li>
72        </ul>
73        </nav>
74
75     <section class="wrapper">
76        <div class="box box1"> Lorem ipsum </div>
77        <div class="box box2"> Lorem ipsum </div>
78        <div class="box box3">Lorem ipsum </div>
79     </section>
80
81     <footer>
82        Ncit college footer
83     </footer>
84 </body>
85 </html>
```

Listing 5: Html css for given design

8 (Q No.4A) How events are handled in JavaScript? Write the JS code for the following o/p. When the user provides quantity it gives total.

Product	Price	Quantity
MOMO	200	
Pizza	200	
Total		

In JavaScript, events are used to handle user interactions or system events, such as a mouse click or a keypress, and trigger corresponding actions.

To handle events in JavaScript, you can use event listeners. An event listener is a function that is executed in response to an event. Here's an example:

```
1     // Get a reference to the element you want to listen for an event on
2 const button = document.querySelector('button');
3
4 // Add an event listener to the element
5 button.addEventListener('click', function() {
6   // Do something when the button is clicked
7   alert("Hello World!!!");
8 });
9 In this example, we get a reference to a button element using document.
    querySelector(), and then add an event listener to it using the
    addEventListener() method. We pass in two arguments to addEventListener(): the
    type of event we want to listen for (in this case, 'click'), and the function
    we want to execute when the event occurs.
10
11 When the button is clicked, the function passed as the second argument to
    addEventListener() will be executed. Inside this function, we have alert
    function which will create a pop box when clicked.
```

Here is the JS code for the given scenario to display the total when quantity are given.

```
1 <html lang="en">
2 <head>
3
4     <title>Document</title>
5 </head>
6 <body>
7     <table border="1">
8         <tr>
9             <th>Product</th>
10            <th>Price</th>
11            <th>Quantity</th>
12        </tr>
13        <tr>
14            <td>MOMO</td>
15            <td id="momoprice">200</td>
```

By: Sanjaya (Bir Bikram) Shrestha

```
16          <td><input type="number" name="momo" id="momo" onkeyup="total(event)">
    </td>
17
18      </tr>
19      <tr>
20          <td>Pizza</td>
21          <td id="pizzaprice">200</td>
22          <td><input type="number" name="pizza" id="pizza" onkeyup="total(event)
    "></td>
23      </tr>
24  </table>
25
26  <button>Total</button> <input type="text" name="total" id="total" readonly>
27
28
29  <script>
30
31
32      function total(event) {
33          let momo = document.getElementById("momo").value
34          let pizza = document.getElementById("pizza").value
35          let momoPrice = document.getElementById("momoprice").innerText
36          let pizzaPrice = document.getElementById("pizzaprice").innerText
37          let total = document.getElementById("total")
38
39          if(momo == ""){
40              momo = 0;
41          }
42          if(pizza == ""){
43              pizza = 0;
44          }
45
46          console.log(momo);
47          console.log(pizza);
48          console.log(momoPrice);
49          console.log(pizzaPrice);
50
51          let totalPrice =( parseInt(momo) * parseInt(momoPrice)) + (parseInt(
    pizza) * parseInt(pizzaPrice))
52
53          console.log(totalPrice);
54          total.value = totalPrice
55      }
56  </script>
57 </body>
58 </html>
```

Listing 6: JS code to display total

Here we have used onKeyUp function to handle the event, when key is pressed the event is triggered which will call the total() function in the JS, where we had created reference to the both quantity and price field, and then we checked if value is null then set to 0 for default value, then after converting to integer we calculated the total value then display in the total box.

9 (Q No.4B) What do you mean by DOM in JavaScript? Differences between DOM 0 and DOM 1 and DOM 2 methods with example.

In JavaScript, the DOM (Document Object Model) is an API that provides a way for JavaScript code to interact with HTML and XML documents as a tree of objects. The DOM represents the structure of a document as a tree of nodes and provides methods and properties for manipulating those nodes.

The differences between the DOM: DOM Level 0, DOM Level 1, and DOM Level 2 are:

- DOM Level 0: This refers to the original, basic DOM API that was built into early versions of web browsers. It includes a few global objects and methods, such as document, window, and alert(), but doesn't provide a formalized API for working with the document tree. Instead, it allows directly manipulate the properties of the document and its elements using JavaScript. Example of using DOM Level 0:

```
1    // Accessing the document object
2    const doc = document;
3
4    // Accessing an element and changing its text content
5    const heading = document.getElementById('heading');
6    heading.textContent = 'New Heading';
7
```

- DOM Level 1: This introduced a formalized API for working with the document tree, including methods for accessing and manipulating elements and their attributes, as well as event handling. This version of the DOM is widely supported in modern browsers. Example of using DOM Level 1:

```
1    // Accessing an element by ID and changing its text content
2    const heading = document.getElementById('heading');
3    heading.textContent = 'New Heading';
4
5    // Adding an event listener to a button element
6    const button = document.getElementById('myButton');
7    button.addEventListener('click', function() {
8    console.log('Button clicked!');
9    });
10
11
```

- DOM Level 2: This added additional features to the DOM, such as support for CSS styling and more advanced event handling. This version of the DOM is also widely supported in modern browsers. Example of using DOM Level 2:

```
1    // Accessing an element's style properties
2    const box = document.getElementById('myBox');
3    box.style.backgroundColor = 'red';
4    box.style.width = '100px';
5    box.style.height = '100px';
6
7    // Adding a mouseover event listener to an element
8    const button = document.getElementById('myButton');
9    button.addEventListener('mouseover', function() {
```

By: Sanjaya (Bir Bikram) Shrestha

```
10          this.style.backgroundColor = 'blue';
11          });
12
13
```

10 **(Q No.5A) Create a registration form as given. Also validates it with JavaScript. (All input should be fulfilled else *required message will display, the name must be of 8 characters, email should in the proper format. the phone must be of 10 digits, password include only number and must be of 8 digits, and starting and ending with 0, Confirm password and Password must match).**

11 **(Q No.1 A) To make an online presence or just transfer any data there are certain predefined rules to follow. Explain any 5 such commonly used protocols that serve different purposes of their own.**