1.How are inline and block elements different from each other?

ANS - The inline and block elements, both are the display values of HTML elements:

1. The inline element is commonly used inside a line intuitively or a place where we only want to take place needed by the content.
    a. It flows along the text, thus it will not clear previous content like block
    b. It will ignore the weight and height properties with top and bottom margin settings with the exception of left and right margin settings(it follows them).
    c. If floated left or right, It will automatically become a block-level element, subject to all block characteristics
    d. It is also subject to vertical-align property.
        i. Span is an inline element which only takes the space that is needed by the text. Its function is to create a box around the text element.
2. The block element is used for implementing block structure. It has the properties of always starting from a new line and taking the full width and height available on a page or on the parent container.
    a. By Default this will be placed below previous elements in the markup(assumption is made that there are no floats, positioning on surrounding elements)
    b. It can have margins or padding.
    c. It ignores vertical-property unlike inline.
        i. The most common block element is div which is used as a container for other HTML elements having no required attributes but is most commonly used with style, class and id.

2.Explain the difference between visibility:hidden and display:none

ANS - Visibility:hidden hides the element but renders it on the web page. It means that it allocates the space of this element in the web page however does not display the content.

- Example; Test | | Test

On the other hand, Display:none means that the content or the element will not even be rendered on the page, in layman terms, by using this property the element is not displayed and space is also not allocated on the pages. The most common example of using this property is in the drop-down list when we want to keep one row blank indicating null values or nothing selected.

- Example; Test | | Test

3.Explain the clear and float properties.

ANS -

- Float property
    - This property specifies how the element should float in the container. The element is placed along the left or right side of the container.

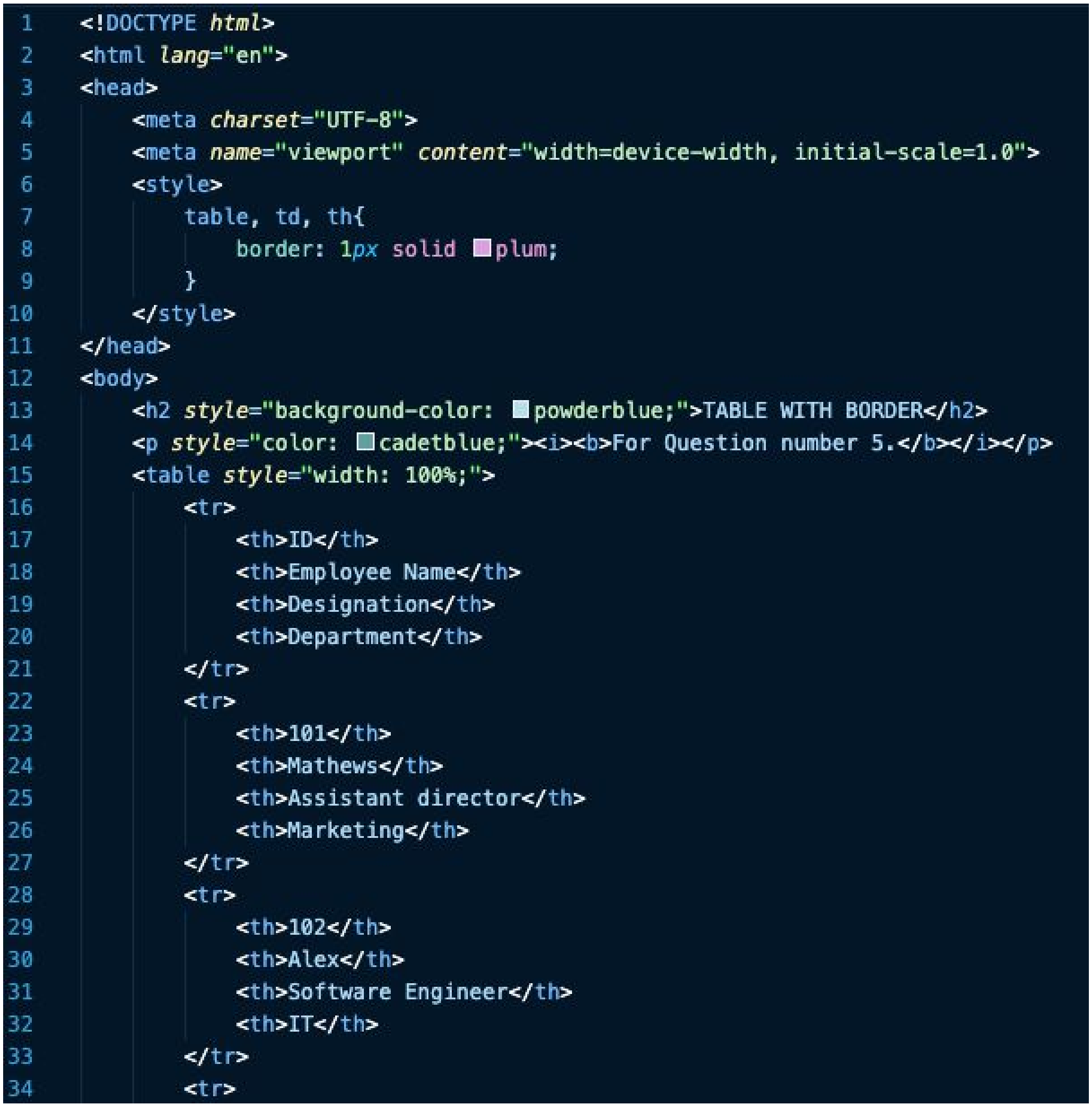
    - This property has 3 possible values:
        - Left - element on the right side of the container.
        - Right - element on the left side of the container.
        - None - there is no floating element in the container.
- Clear property
    - This property is related to the float property as it specifies if an element should be next to the floated elements or if it should move below them. This property applies to both floated and non-floated elements.
    - In layman terms, If an element can fit in the horizontal space next to floated element, it will or you can apply the clear property to that element in the same direction as the float which will move the element below the floated elements
    - The clear property can have following values:
        - None - The element is not moved down to clear the past floats.
        - Left - The element is moved down to clear past left floats.
        - Right - The element is moved down to clear past right floats.
        - Both - The element is moved down to clear past both left and right floats.

4.Explain the difference between absolute, relative,fixed and static.

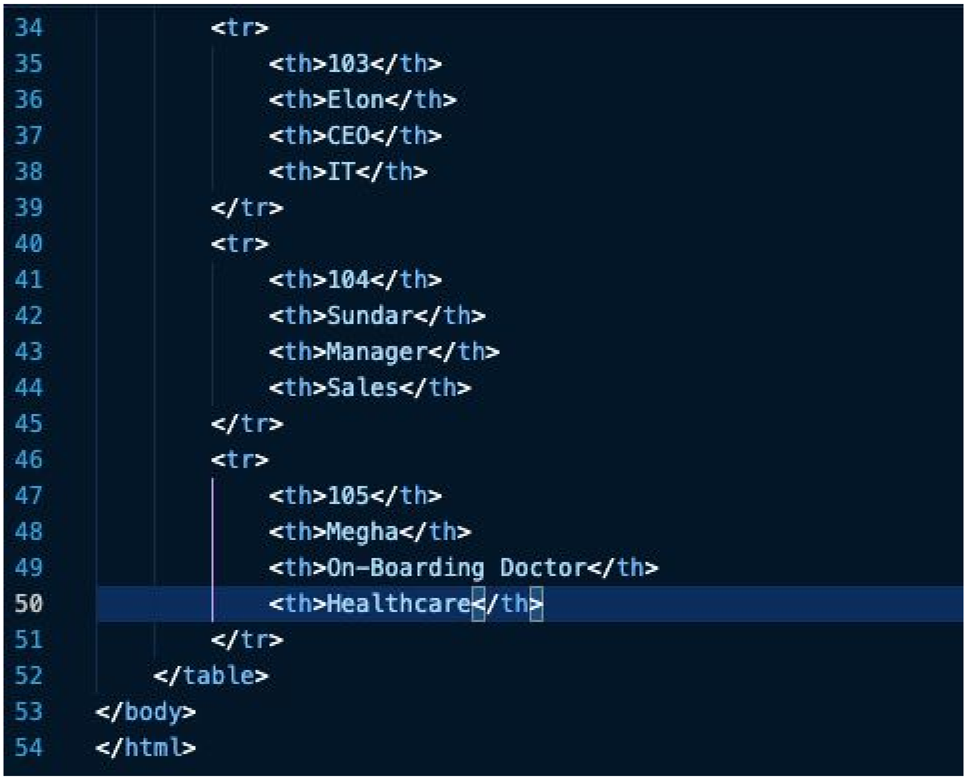
ANS - All the above mentioned properties come under CSS position property which defines how the element is positioned on the web page.

- **STATIC:** This is the default value which is applied to all elements in order as they appear in the document. We usually use this to set all the values to default, if we mess up somehow.
- **RELATIVE:** The element is positioned relative to its normal position. Setting the top, bottom, right, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.
    - Note - While setting position - relative to an element limits the scope of absolute positioned child elements. This means that any element that is the child of this element can be absolutely positioned within this block.
- **ABSOLUTE:** The element is positioned absolutely to its first positioned parent. The positioning is done, relatively (or absolutely) positioned parent element. In the case where there is no positioned parent element, it will be positioned related directly to the HTML element (the page itself).
    - While using absolute positioning is to make sure it is not overused for preventing maintenance problems.
- **FIXED:** the element is positioned related to the browser window. A fixed position element is positioned relative to the viewport, or the browser window itself. The viewport doesn’t change when the window is scrolled, so a fixed positioned element will stay right where it is when the page is scrolled.

5.Write the HTML code to create a table in which there are 4 columns(ID , Employee Name, Designation, Department) and at least 6 rows. Also do some styling to it.



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        table, td, th{
            border: 1px solid plum;
        }
    </style>
</head>
<body>
    <h2 style="background-color: powderblue;">TABLE WITH BORDER</h2>
    <p style="color: cadetblue;"><i><b>For Question number 5.</b></i></p>
    <table style="width: 100%;">
        <tr>
            <th>ID</th>
            <th>Employee Name</th>
            <th>Designation</th>
            <th>Department</th>
        </tr>
        <tr>
            <th>101</th>
            <th>Mathews</th>
            <th>Assistant director</th>
            <th>Marketing</th>
        </tr>
        <tr>
            <th>102</th>
            <th>Alex</th>
            <th>Software Engineer</th>
            <th>IT</th>
        </tr>
        <tr>
```



```
        <tr>
            <th>103</th>
            <th>Elon</th>
            <th>CEO</th>
            <th>IT</th>
        </tr>
        <tr>
            <th>104</th>
            <th>Sundar</th>
            <th>Manager</th>
            <th>Sales</th>
        </tr>
        <tr>
            <th>105</th>
            <th>Megha</th>
            <th>On-Boarding Doctor</th>
            <th>Healthcare</th>
        </tr>
    </table>
</body>
</html>
```

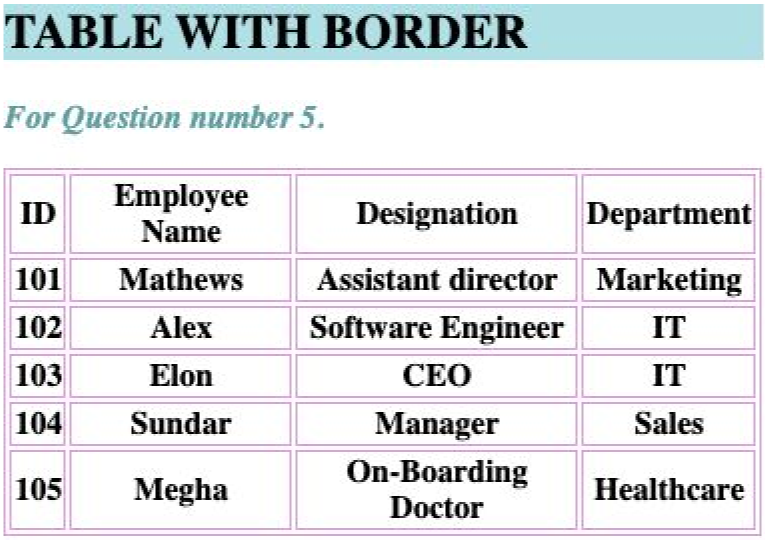


TABLE WITH BORDER

For Question number 5.

ID	Employee Name	Designation	Department
101	Mathews	Assistant director	Marketing
102	Alex	Software Engineer	IT
103	Elon	CEO	IT
104	Sundar	Manager	Sales
105	Megha	On-Boarding Doctor	Healthcare

6. Why do we use meta tags?

ANS - Meta tags are snippets of text that describe a page's content; the meta tags don't appear on the page itself, but only in the page's source code. Meta tags are essentially little content descriptors that help tell search engines what a web page is about. The only difference between tags you see (on a blogpost, say) and tags you can't see is location: meta tags only exist in

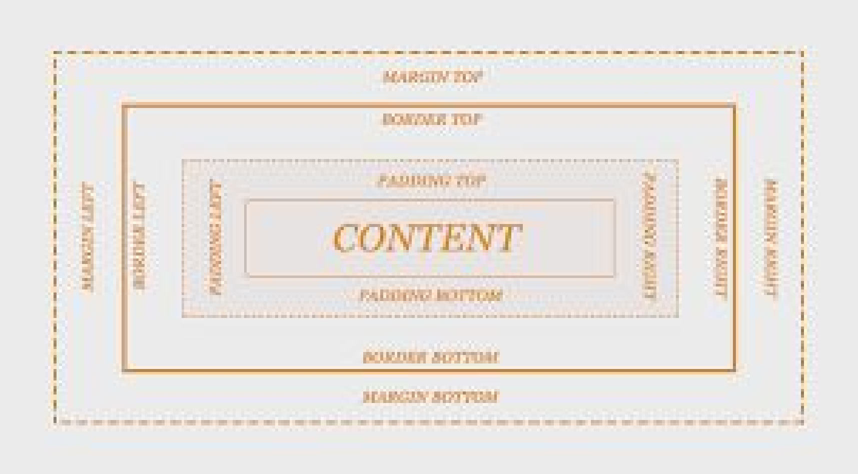
HTML, usually at the “head” of the page, and so are only visible to search engines (and people who know where to look). The “meta” stands for “metadata,” which is the kind of data these tags provide – data about the data on your page.
We use meta tags because they make it easier for search engines to determine what your content is about, and thus are vital for SEO(Search Engine Optimization) for retailers.

7. Explain box model.

ANS - The term “Box Model” is used while talking about design and layout. In this concept all HTML elements are considered as boxes.
The CSS box model is essentially a box that wraps around every HTML element. It consists of:

1. Margins
2. Borders
3. Padding
4. The actual content.

The image below illustrates the box model:



- **Content:** The content of the box, where text and images appear.
- **Padding:** Clears an area around the content. The padding is transparent.
- **Border:** A border that goes around the padding and content.
- **Margin:** Clears an area outside the border. The margin is transparent.

8. What are the different types of CSS Selectors?

ANS - CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors
  - It selects the elements based on the element name.
  - CSS element selector: uses the id attribute of an HTML element to select a specific element.
    - Example: `p { text-align: center; color: red;}`

    - CSS id selector: The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character, followed by the id of the element.
        - Example: `#para1 { text-align: center; color: red;}`
    - CSS class selector: selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.
        - Example: `#para1 { text-align: center; color: red;}`
    - CSS grouping selector: selects all the HTML elements with the same style definitions. To group selectors, we separate each selector with a comma.
        - Example: `h1, h2, p { text-align: center; color: red;}`
    - CSS universal selector: (*) selects all HTML elements on the page.
        - Example: `* { text-align: center; color: blue;}`
- Combinator selectors: A combinator is something that explains the relationship between the selectors. Between the simple selectors, we can include a combinator.
    - There are four different combinators in CSS:
        - descendant selector (space)
        - child selector (>)
        - adjacent sibling selector (+)
        - general sibling selector (~)
    - Example: div > p { background-color: yellow; }
- Pseudo-elements selectors: A pseudo-class is used to define a special state of an element.
    - For instance:
        - Style an element when a user mouses over it
        - Style visited and unvisited links differently
        - Style an element when it gets focus.
    - Example: selector::pseudo-element { property: value; }
- Attribute selectors: The [attribute] selector is used to select elements with a specified attribute.
    - Eg: a[target="_blank"] { background-color: yellow; }

9. Define Doctype.

ANS - The HTML document type declaration, also known as DOCTYPE, is the first line of code required in every HTML or XHTML document. The DOCTYPE declaration is an instruction to the web browser about what version of HTML the page is written in. This ensures that the web page is parsed the same way by different web browsers. In HTML 5, the declaration is simple:

```
<!DOCTYPE html>
```

In HTML 4.01, the DOCTYPE declaration refers to a document type definition (DTD). A DTD defines the structure and the legal elements of an XML document.

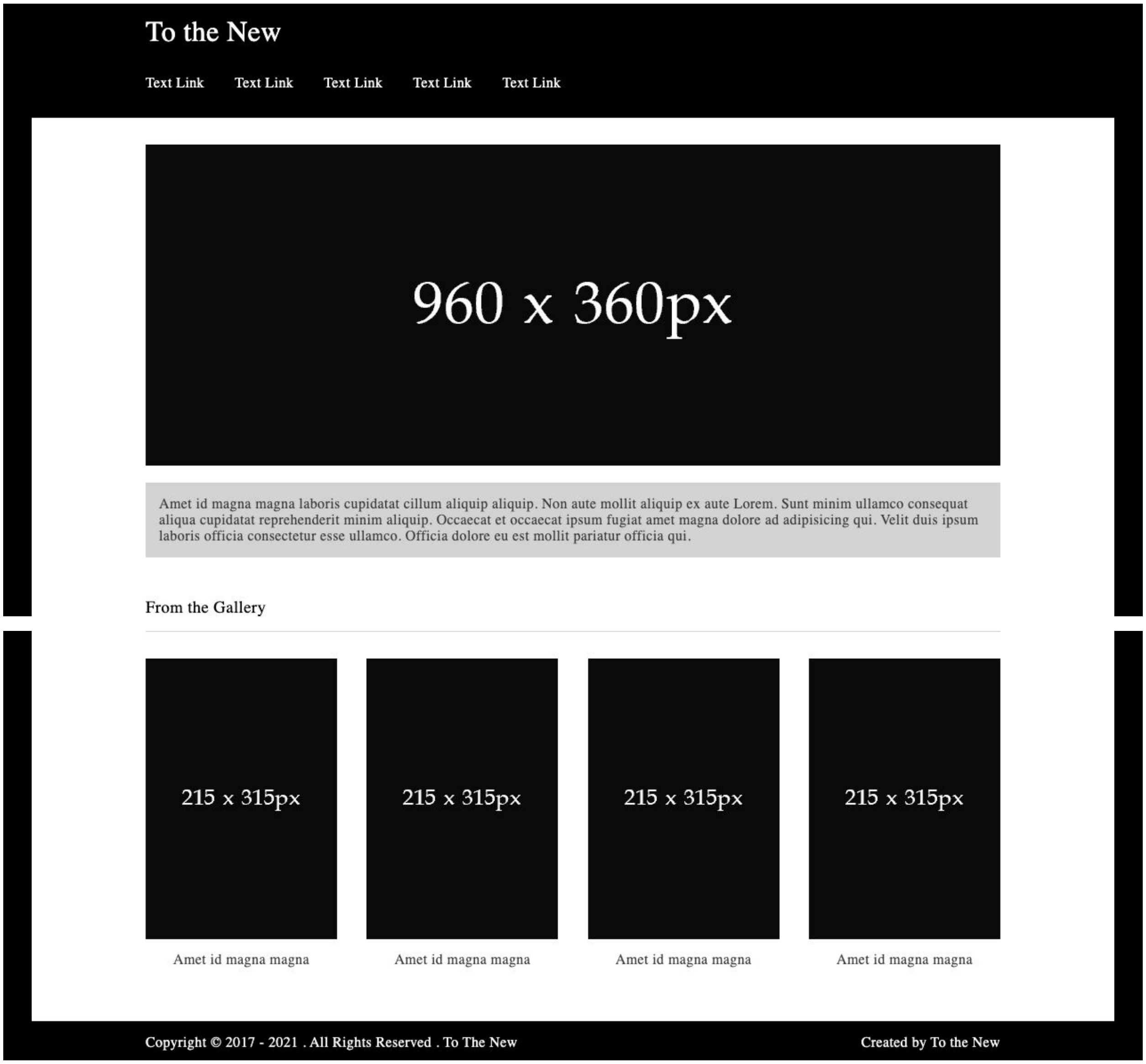
10. Explain 5 HTML5 semantic tags.

ANS The 5 HTML5 semantic tags are as follows:

- Article tag:
    - The <article> element specifies independent, self-contained content. An article should make sense on its own, and it should be possible to distribute it independently from the rest of the web site.
    - Examples of where an <article> element can be used:
        - Forum post
        - Blog post
        - Newspaper article
- Header Tag
    - The <header> element represents a container for introductory content or a set of navigational links. A <header> element typically contains:
        - one or more heading elements (<h1> - <h6>)
        - logo or icon
        - authorship information
    - You can have several <header> elements in one HTML document. However, <header> cannot be placed within a <footer>, <address> or another <header> element.
- Footer Tag
    - The <footer> element defines a footer for a document or section. A <footer> element typically contains:
        - authorship information
        - copyright information
        - contact information
        - Sitemap
        - back to top links
        - related documents
    - You can have several <footer> elements in one document.
- Nav Tag
    - The <nav> element defines a set of navigation links.
    - Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major blocks of navigation links. Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.
- Aside Tag
    - The <aside> element defines some content aside from the content it is placed in (like a sidebar). The <aside> content should be indirectly related to the surrounding content.

11. Create HTML for web-page.jpg (check resources, highest weightage for answers)

ANS - Code is attached as a file in folder 1 filename 1.html



To the New

Text Link Text Link Text Link Text Link Text Link

Amet id magna magna laboris cupidatat cillum aliquip aliquip. Non aute mollit aliquip ex aute Lorem. Sunt minim ullamco consequat aliqua cupidatat reprehenderit minim aliquip. Occaecat et occaecat ipsum fugiat amet magna dolore ad adipisicing qui. Velit duis ipsum laboris officia consectetur esse ullamco. Officia dolore eu est mollit pariatur officia qui.

From the Gallery

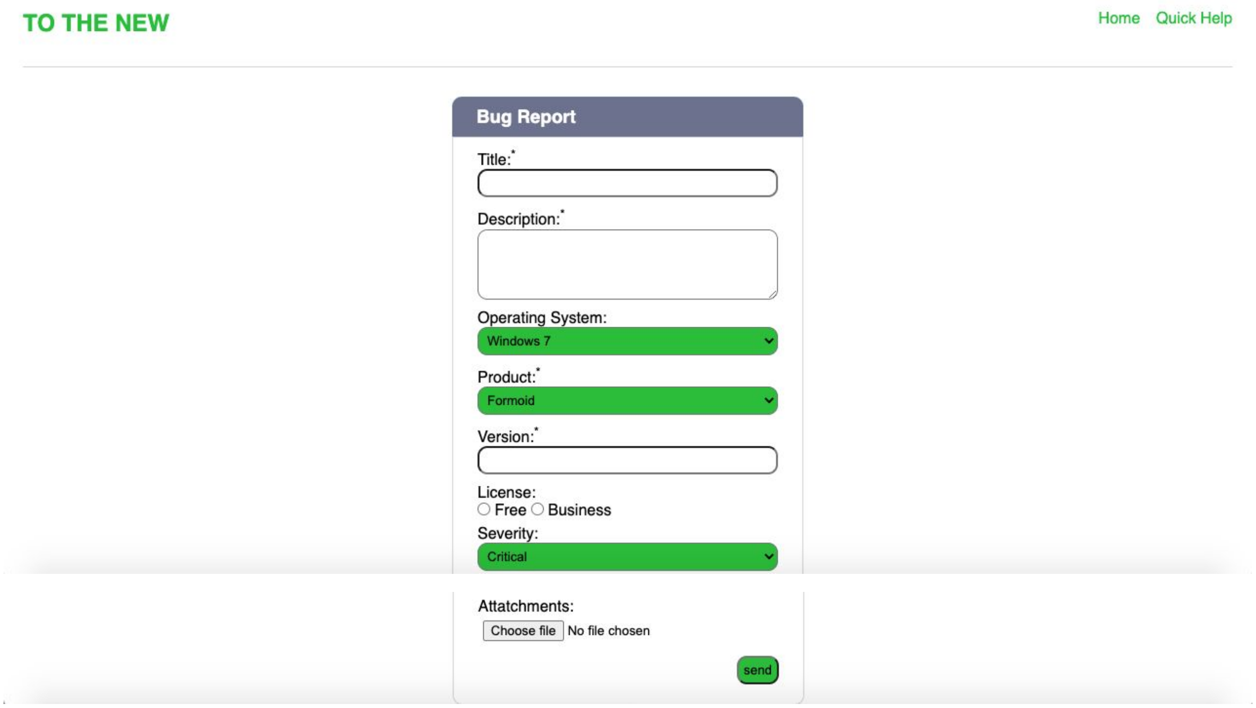
Amet id magna magna Amet id magna magna Amet id magna magna Amet id magna magna

Copyright © 2017 - 2021 . All Rights Reserved . To The New

Created by To the New

12. Create HTML for form.png (check resources, highest weightage for answers)

ANS - Code is attached as a file named as web1.html



Bug Report

Title:*

Description:*

Operating System: Windows 7

Product:* Formoid

Version:*

License:
○ Free ○ Business

Severity: Critical

Attachments: