1. A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

For example: <div> , <h1-h6> , <canvas> , <article> etc.

An inline element does not start on a new line and only takes up as much width as necessary.

For example: <span>, <input> , <img> , <a> etc.

1. **Display: none** , is commonly used with JavaScript to hide and show elements without deleting and recreating them.

**Visibility: hidden** also hides an element. However, the element will still take up the same space as before. The element will be hidden, but still affect the layout:

1. The **Float** property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The **Clear** property specifies what elements can float beside the cleared element and on which side.

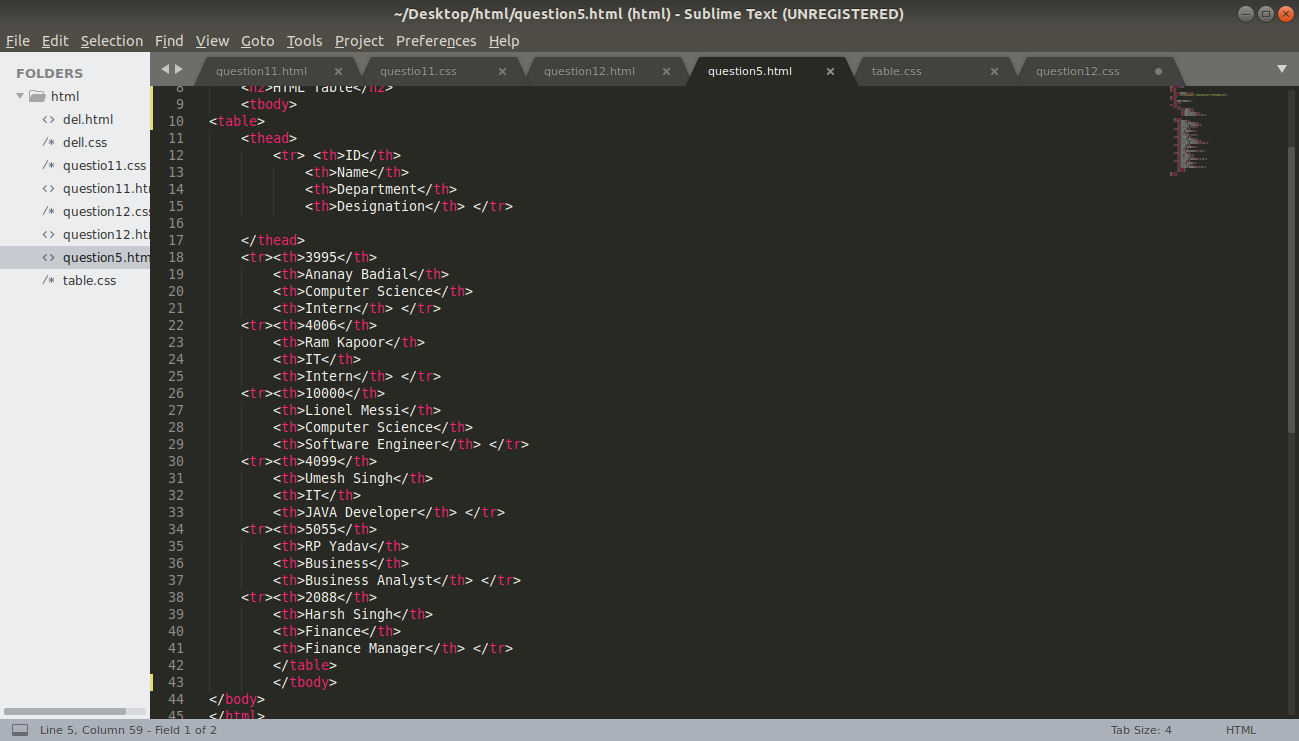
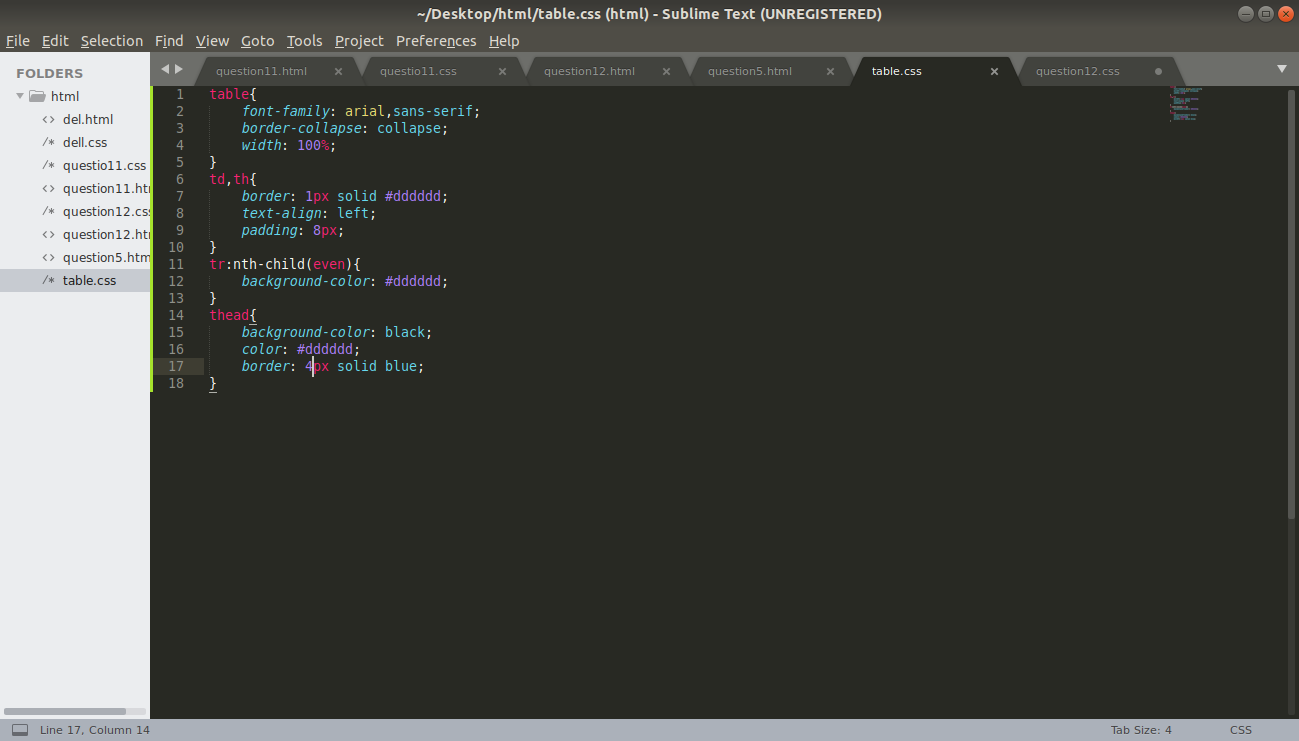
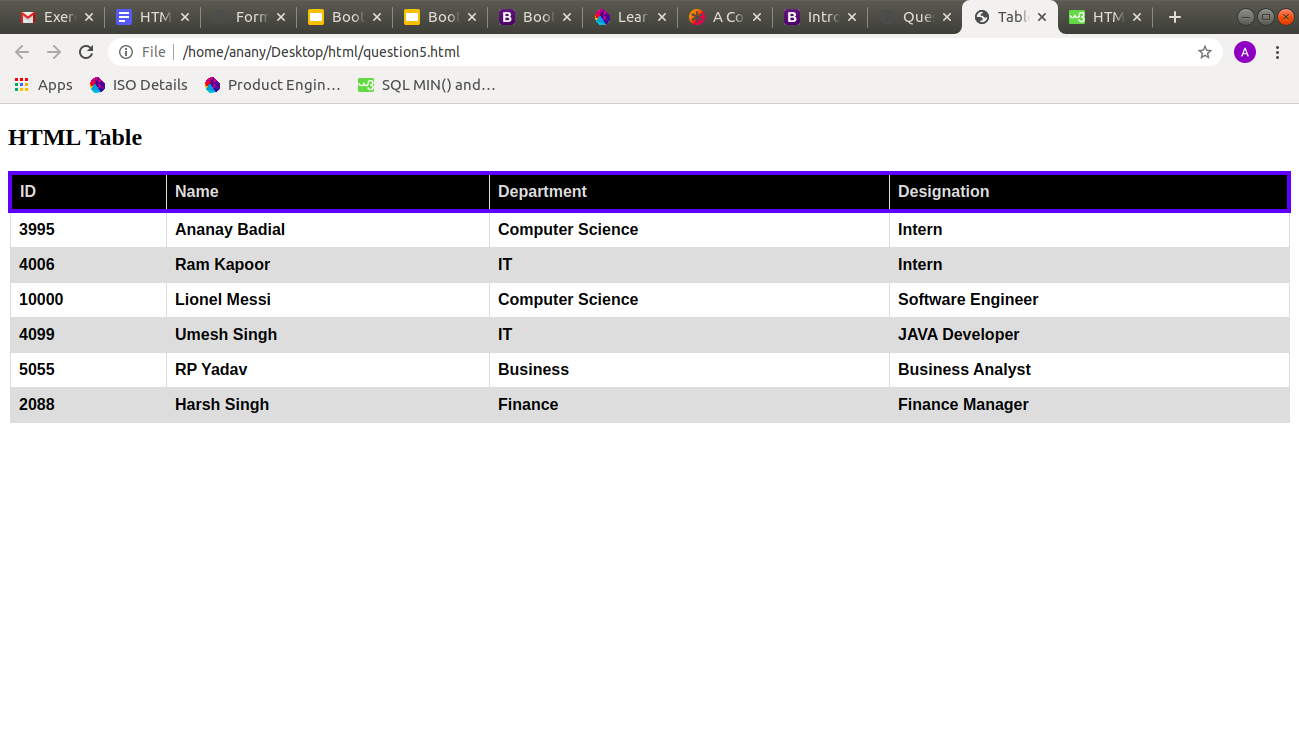
1. Static: An element with position: static is not positioned in any special way; it is always positioned according to the normal flow of the page.

Relative: An element with position: relative is positioned relative to its normal position.Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position.

Fixed: An element with position: fixed is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.

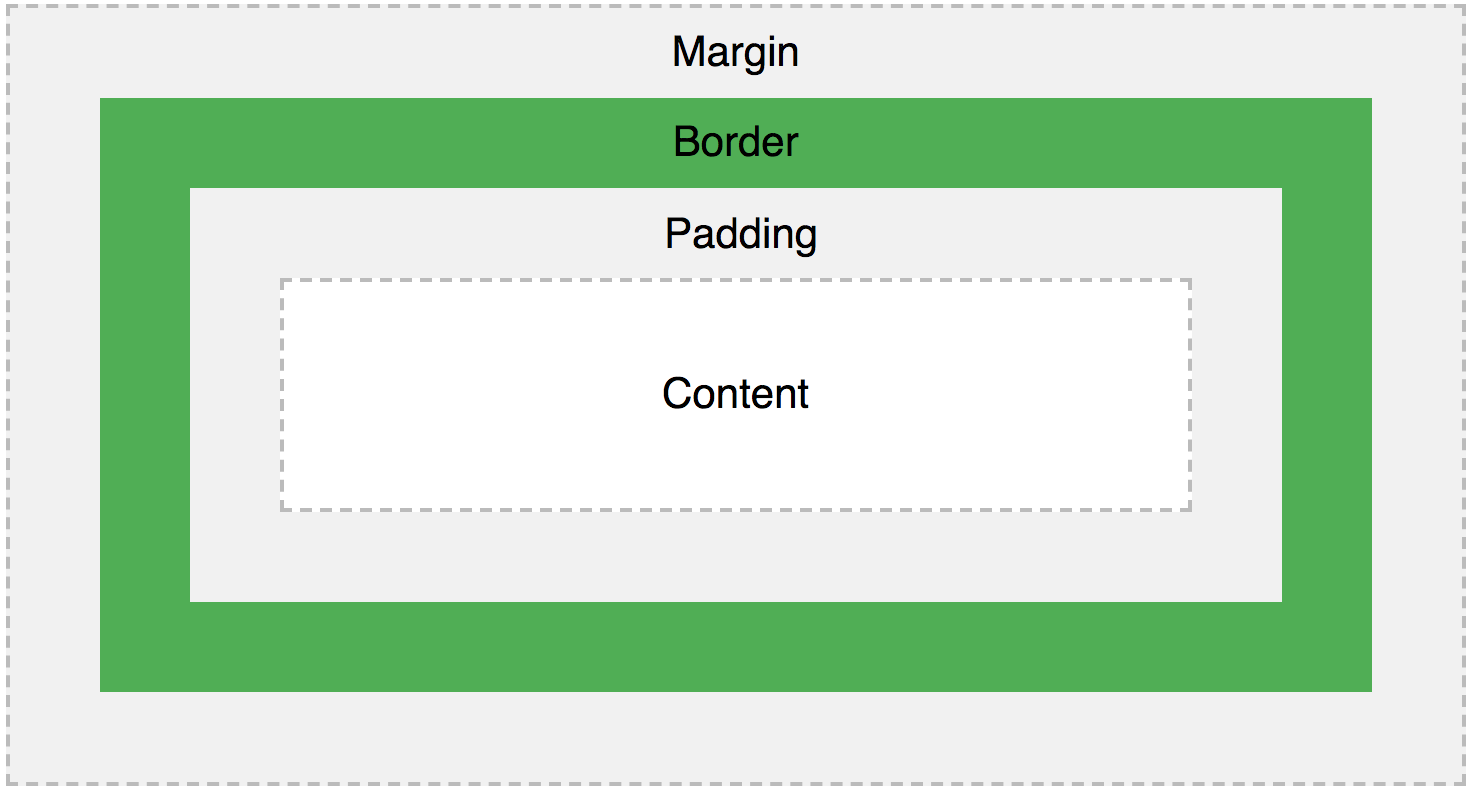
Absolute: An element with position: absolute is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

5.



6. The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable. Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

7.



All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.

Explanation of the parts in 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. The different CSS Selectors are:

1. **Id Selector**:The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element!
2. **Class Selector**: The 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.
3. **Universal Selector (\*)**: The universal selector (\*) selects all HTML elements on the page

**Grouping Selector**: The grouping selector selects all the HTML elements with the same style definitions.

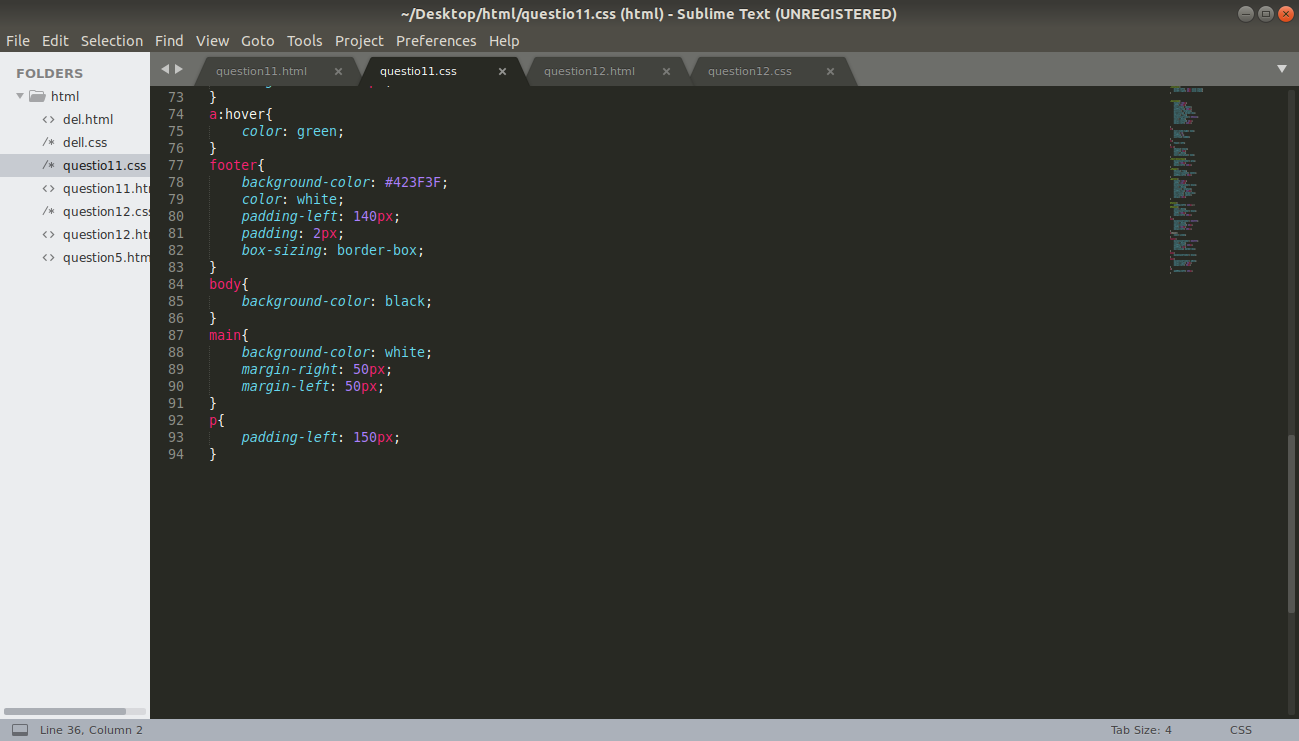
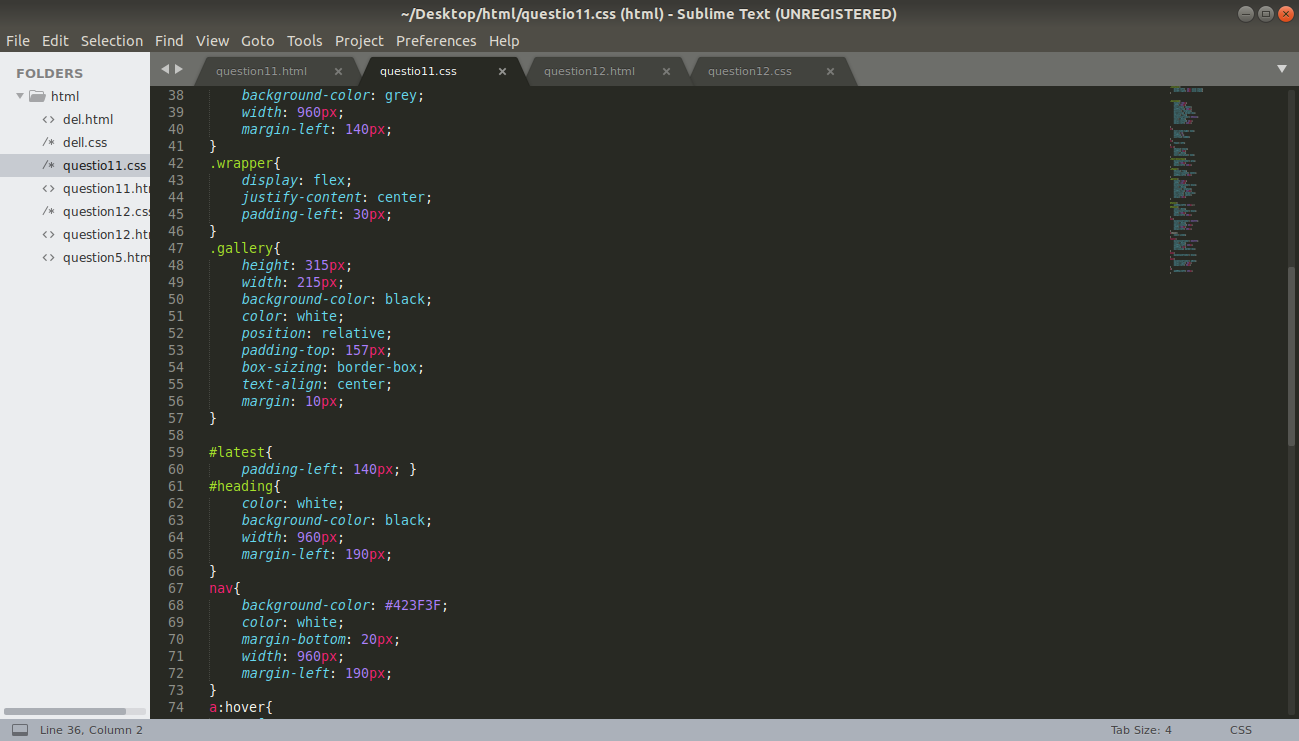
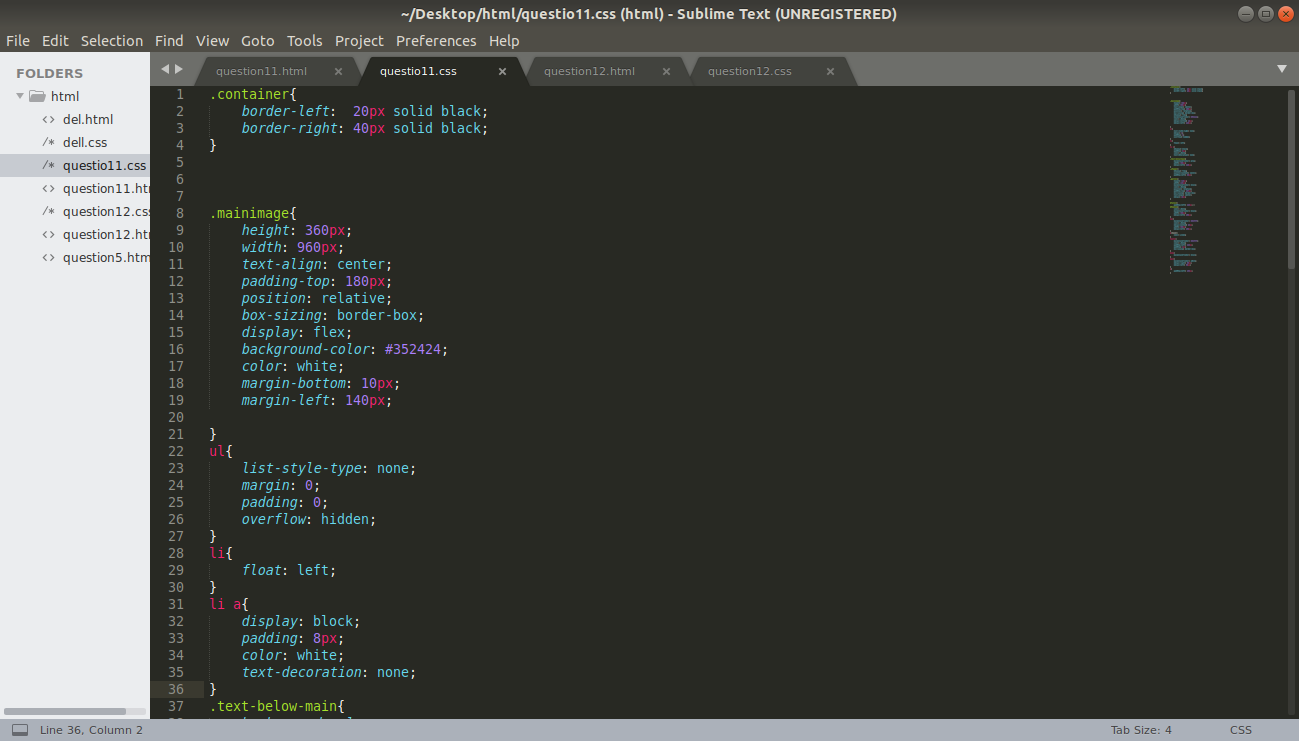
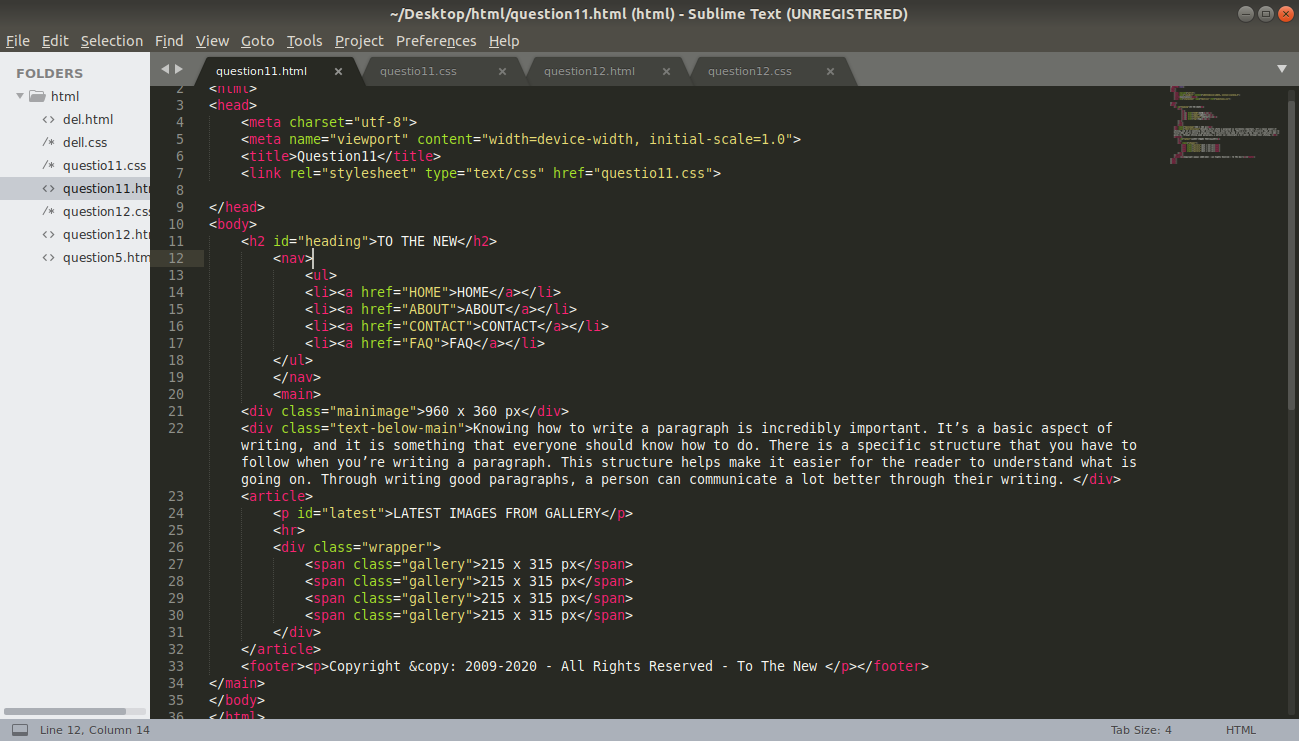
9. The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

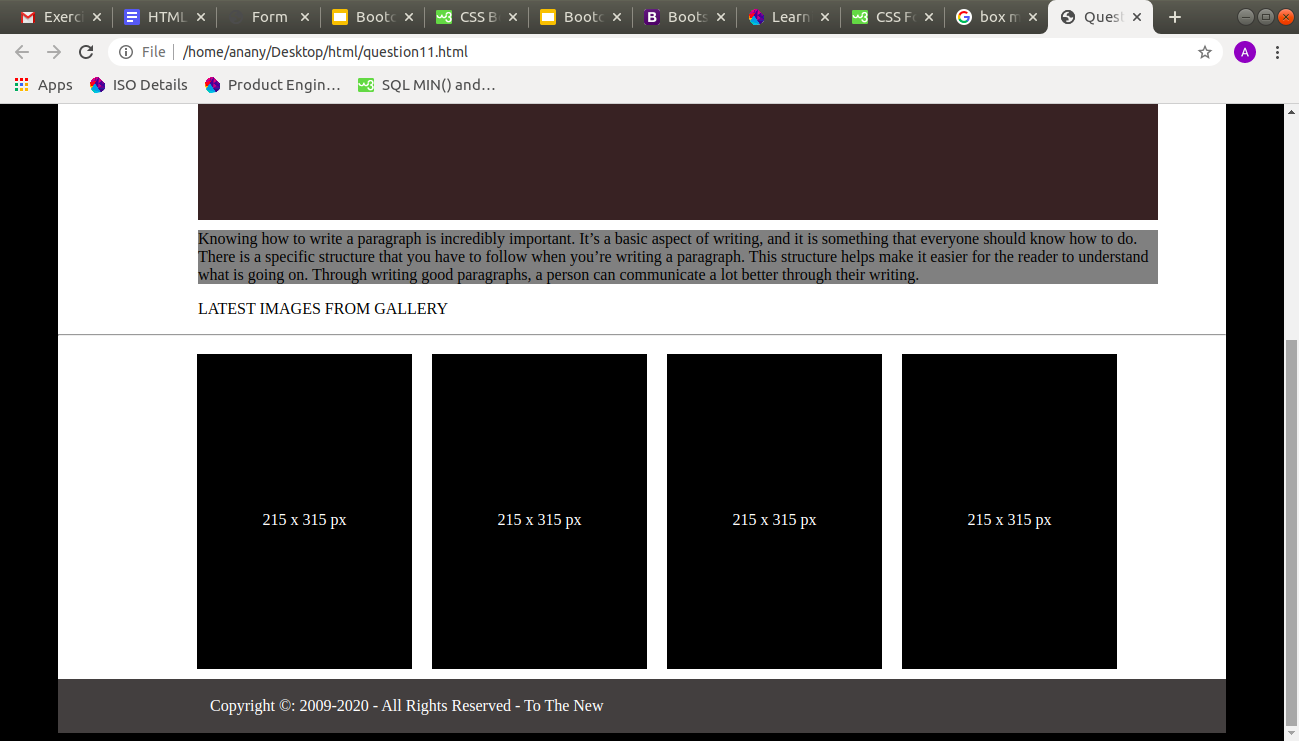
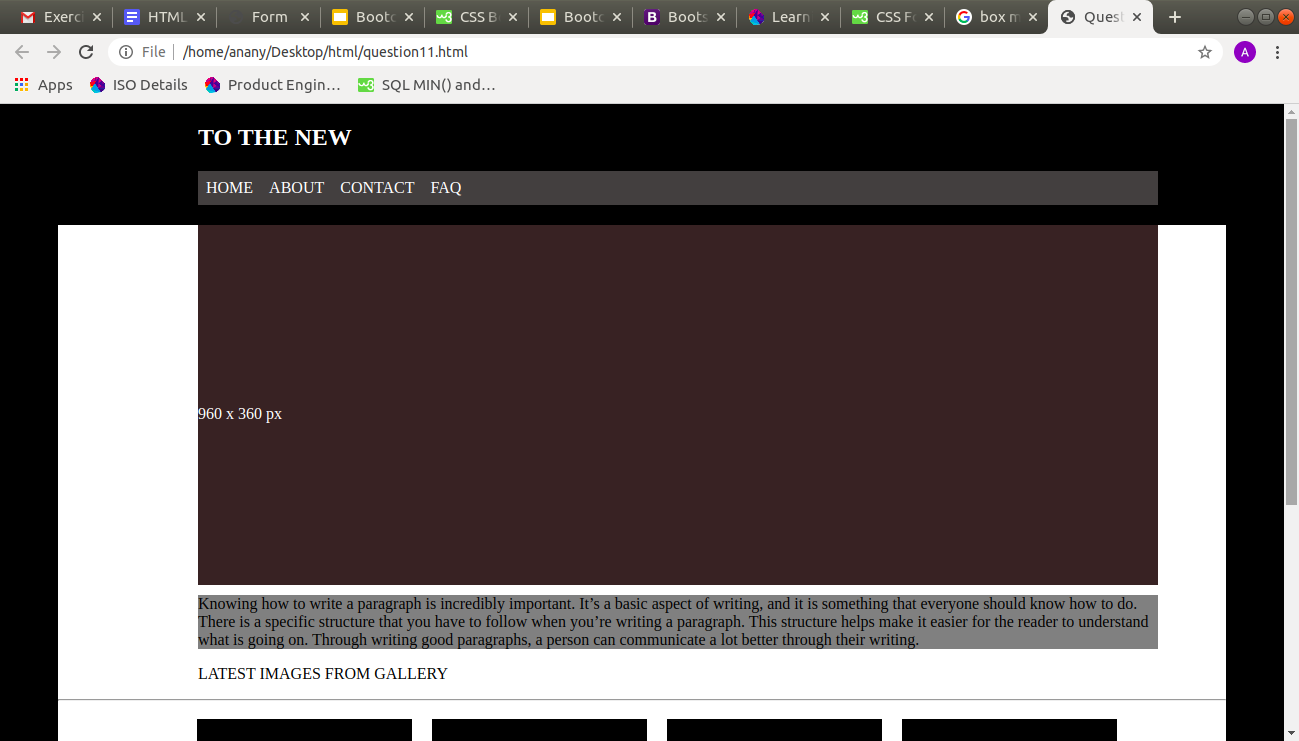
In HTML 4.01, the <!DOCTYPE> declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly. HTML5 is not based on SGML, and therefore does not require a reference to a DTD. We must always add the <!DOCTYPE> declaration to your HTML documents, so that the browser knows what type of document.

10. **Semantic tags used in HTML are:**

1. <article> : The <article> element specifies independent, self-contained content. An article should make sense on its own, and it should be possible to read it independently from the rest of the web site.
2. <header> : The <header> element specifies a header for a document or section.The <header> element should be used as a container for introductory content.
3. <footer> : The <footer> element specifies a footer for a document or section. A <footer> element should contain information about its containing element.
4. <nav>: The <nav> element defines a set of navigation links.
5. <aside>: The <aside> element defines some content aside from the content it is placed in (like a sidebar). The <aside> content should be related to the surrounding content.

11.





12.

