[**Screen**](https://developer.mozilla.org/en-US/docs/Web/API/Screen)**:-**

**=>The Window property screen returns a reference to the screen object associated with the window. The screen object, implementing the Screen interface, is a special object for inspecting properties of the screen on which the current window is being rendered.**

[**Examples**](https://developer.mozilla.org/en-US/docs/Web/API/Window/screen#examples)

**if (screen.pixelDepth < 8) {**

**// use low-color version of page**

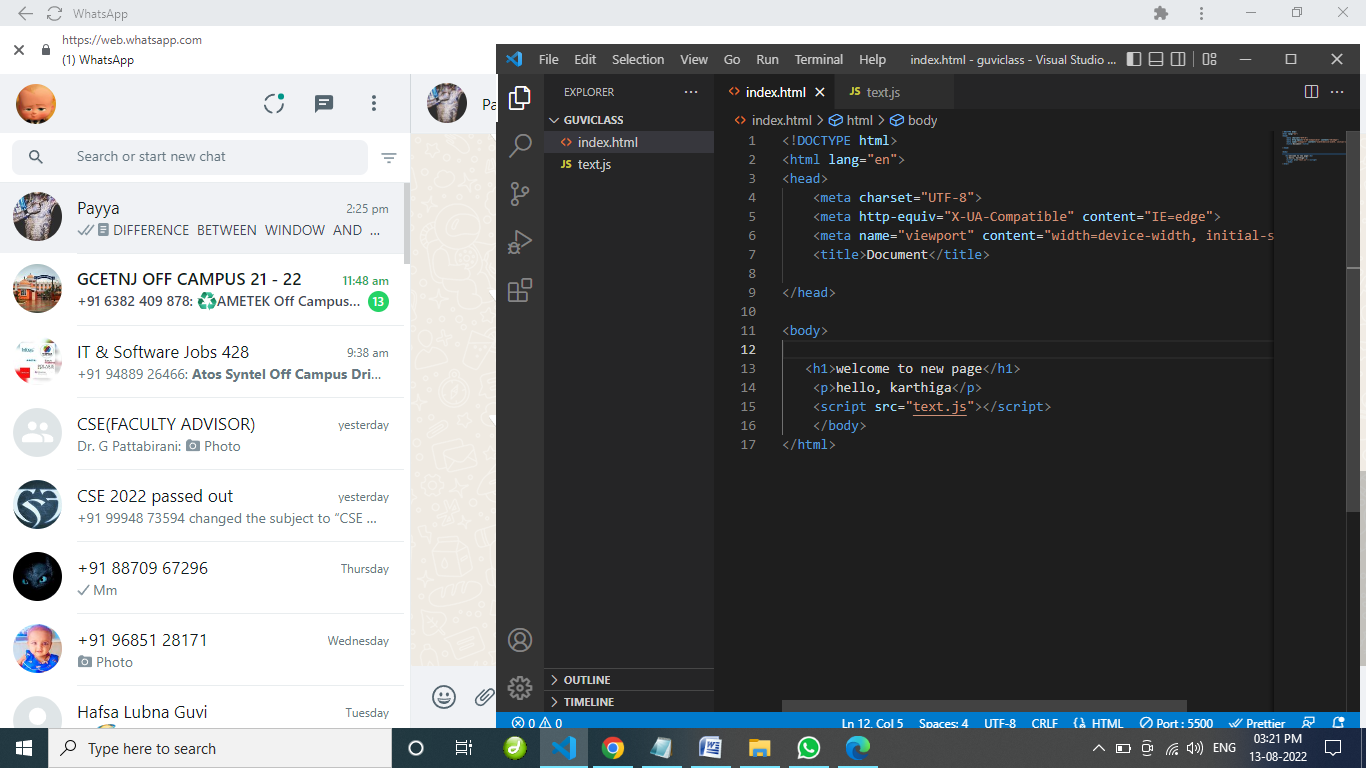
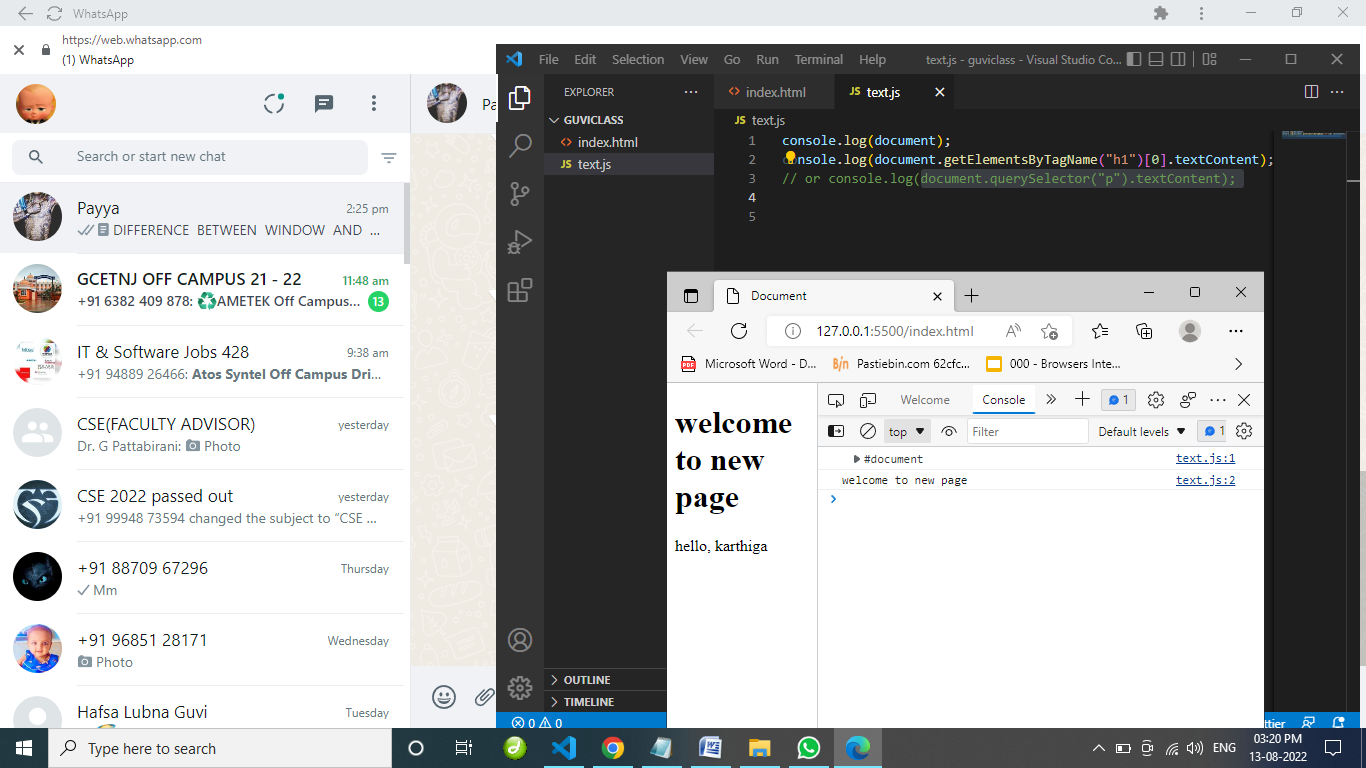
**} else {**

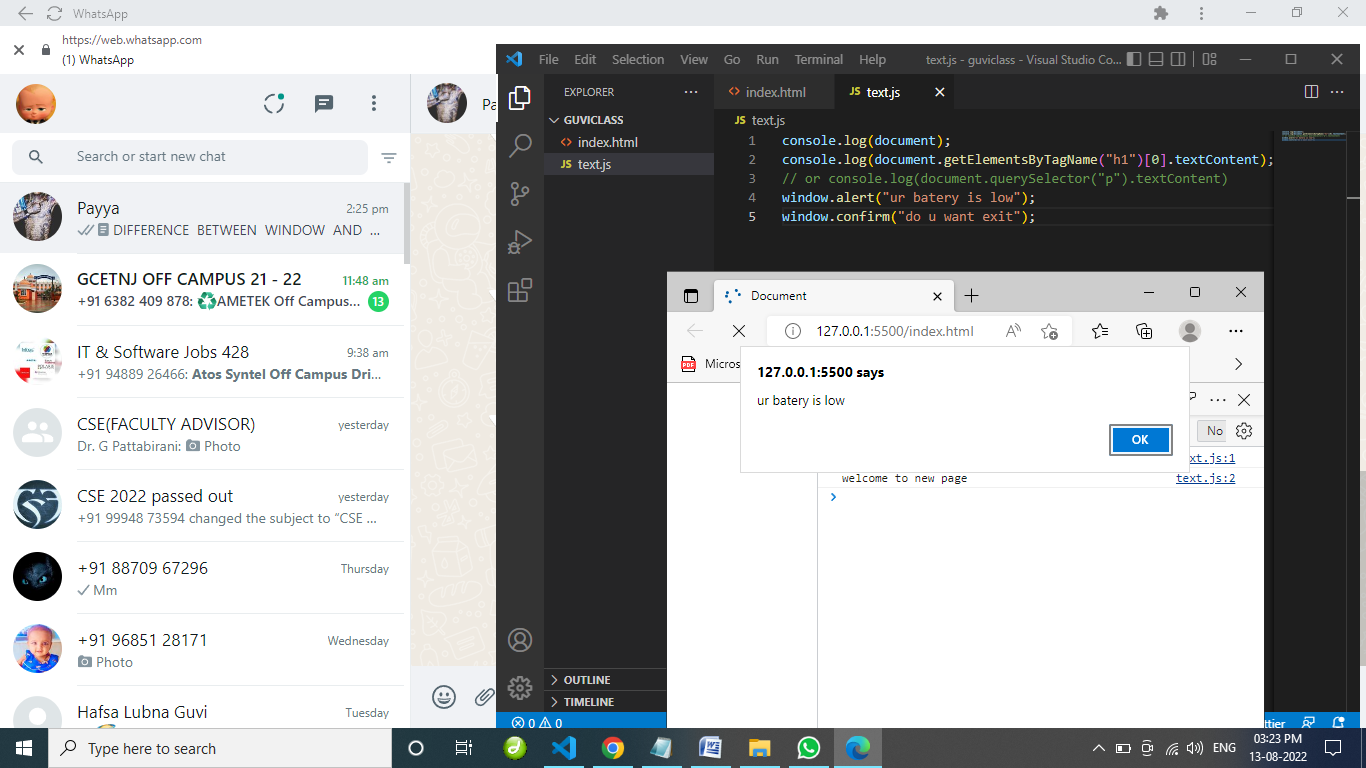
**// use regular, colorful page**

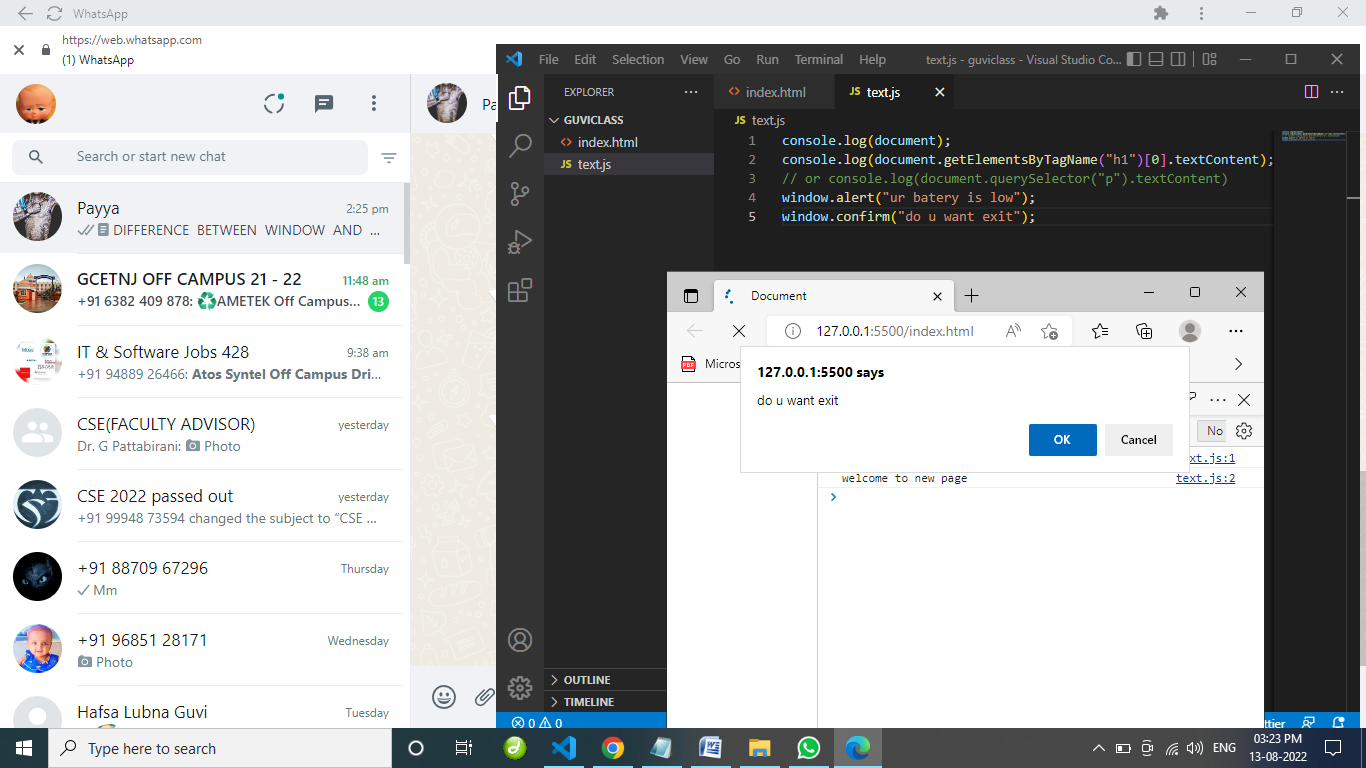
**}**

**=>The window object also has a screen object with properties describing the physical display:**

* **screen properties width and height are the full screen**
* **screen properties availWidth and availHeight omit the toolbar.**



RESULT FOR WINDOW WILL LOOK LIKE THIS: ****



Window is the very first object that is loaded in the browser.

It is used to give alret message to the user.

**Syntax of the window is:**

window.propertyname;

**For example:**

I want to show the pop-up alert message:-

**So that i give,**

window.alert("ur batery is low");

**And also i give confirm message ,**

window.confirm("do u want to exsit");

THIS IS SCREENSHARE OF HTML FILE.

RESULT FOR DOCUMENT WILL LOOK LIKE THIS:

Screenshot for document objects ,how its work like,

Document is a object, used to access the DOM.

It is loaded inside the window.

**Syntax of the document is:**

document.propertyname;

**For example:**

i want to read the h1 tag content:-

So that we have to give like this inside the console.log(),

**This is one type,**

document.getElementsByTagName("h1")[0].textContent;

**The another type is,**

document.querySelector("h1").textContent;