CS 251: Outlab Lab 02: [Presentation] HTML, CSS, JavaScript, Inkscape

- Handed out: 7/31 Due: 8/3 11pm (except for challenge problem)
- Please write (only if true) the honor code. If you used any source (person or thing) explicitly state it. You can find the honor code on the web page.

Overview

The goal of this lab is to make you get a grip on the basics of HTML pages on the Internet.

Pre-tasks

Look up `http://www.w3schools.com/` to get started with HTML, CSS and JS. Similarly look up `http://inkscape.org/doc/basic/tutorial-basic.html` to get started with Inkscape.

The Tasks

The purpose of this task is to able to present your work on a web page You might have seen a gazillion web pages; wait, we are not going to make a super duper page. We will do just enough to get the basics of a web page. A modern web page has these principal components: CSS, Images, Javascript, and Forms. We won't do forms yet but we will do other stuff.

1. Basic Web page stub: This page will have the following components

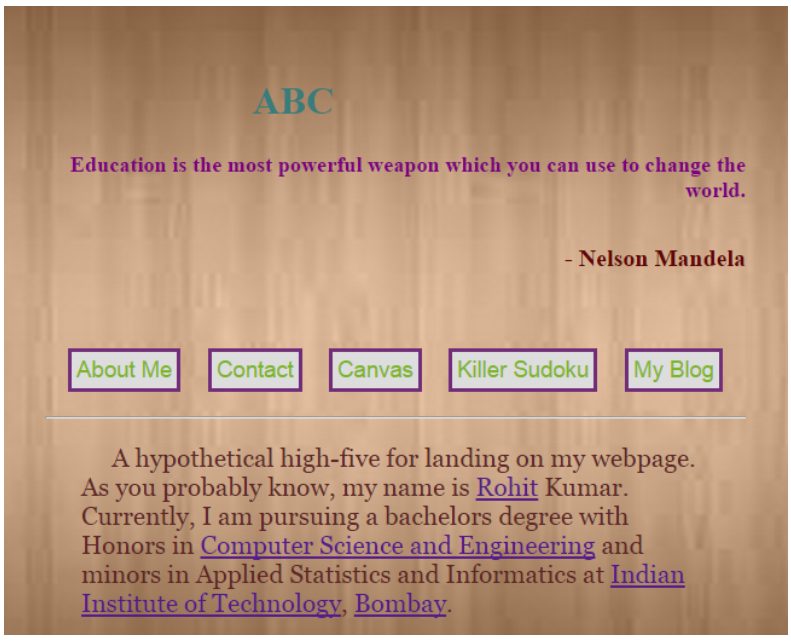


Figure 1: A starting web page

- The landing page will be a template for all things to come. See Figure 1 for a representation. This page will be named `index.html`. It is essential that you also place links to your lab group in this page (not shown in the figure).

- The next page will be an "About me" page (check Figure 2). It will introduce you to the audience: Where are you from? What are you doing at IITB? Anything you want to say about yourself? Name this page `aboutMe.html`

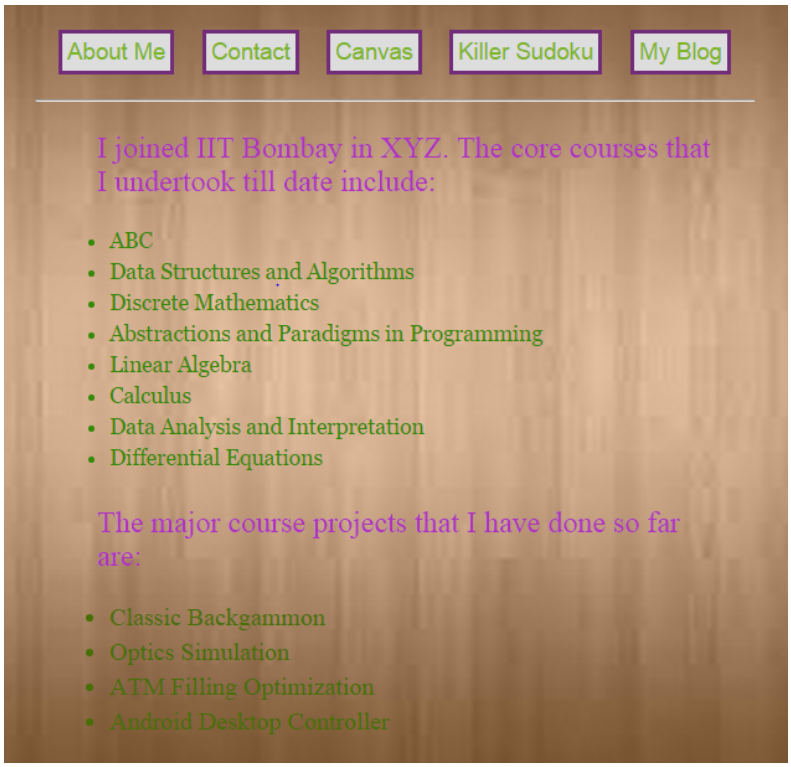


Figure 2: About me page

- Make another page called `group.html` where put details (i.e., name, roll number, group number, group name) of you group members. Also, put a picture of each group member. Clicking on the image must open their personal webpages. Note: This page, i.e., `group.html` has to be the same for the group.
- After you get the basics working, simply move your entire directory (folder) to another folder. Does everything work as well? If not, rework the links.
- Your pages must contain and use sensibly at least the following html tags : <title>, <head>,<body>, , <p>, <ul>,<ol>, <img>, <a href>, <a name>
  Feel free to use more.

How we will score you: This section has 30 marks.

`Note:` Moving the folder should not result in any error. If it does, you get a zero here.

2. CSS: A basic guideline is to separate content and form. The markup language originally tells the browser what to write, and ALSO how to mark them up, i.e., display. This is considered bad practice. Instead, the Cascading Style Sheet (CSS) is the preferred way to decorate a page.

   Learn to style the html elements by using inline styling, included style blocks in the "same" html file, and separate style file. Style one html element each a) inline, b) using a style block within in the html file and c) using a separate CSS style file. The style file must be called `index.css`.

   Use the same stylesheet for all webpages in this task.

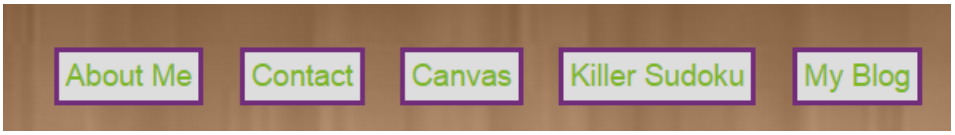


Figure 3: Note how style elements with CSS has been used to beautify these buttons.

How we will score you: This section has 30 marks

- The style elements should be confined only to `aboutMe.html` page so that it is easier for us to grade. Only this page will be checked for decorations.
- 10 each for the three parts mentioned above.

3. Javascript: A web page like the one you have so far is pretty static. We want to be able to interact with the web page. One way of doing this is using JavaScript. This is a long topic but we want to do the real basics.

   - Create a new page called `contactMe.html`. Introduce the `<script>` tag to enable JavaScript, and create a *hover* functionality on this page.

     Create a button. If you hover on the button, your contact information should be revealed to the users and the color of the element should change. Feel free to place whatever little or large information here, we won't worry about the content. However, on taking the mouse away from the element, the contact information should disappear. Your code should be in a file called `index.js`. There might be a variety of ways of making this feature happen, be minimalistic.

How we will score you: This section has 10 marks.

4. Canvas: This is a relatively new feature (HTML5). Create a file called `mycanvas.html` that has a HTML Canvas. This canvas should contain a simple drawing of, say, a leaf. It should have **at least 3** vector elements (not an image).

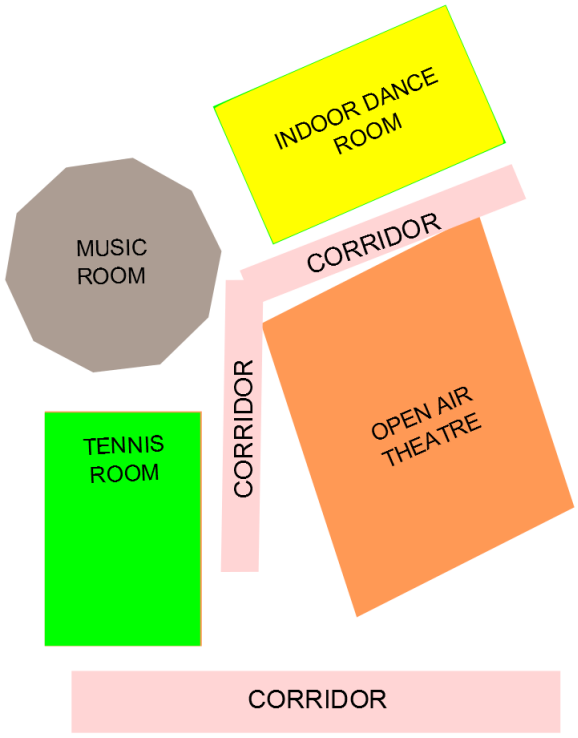
   For inspiration take a look at `http://davidwalsh.name/canvas-demos`.

How we will score you: This section has 20 marks.

5. Inkscape: A picture is worth a 1000 words, and these days no document is considered good unless you have pictures (Contrast images vs pictures vs graphic art). Now pictures like the one you have created above are all fine, but to create more complex ones in vector format you need a tool like `inkscape`. In this task, you will be using `inkscape` to make a model of the student activity center (SAC). Think this is tough? Have you used the `instimap?` on your Android phone? Well, many parts of the app required painstaking hours on the computer with a tool like `inkscape`. And we use the app without thinking twice.

   - Make a diagram indicating the layout of the ground floor of SAC. You may have to visit the SAC for this :)
   - Indicate corridors and rooms properly
   - Display this image on the web page. Name the image as `sac.png`

   An example is shown below. This example is very crude so do not to copy the exact design but be creative!



How we will score you: This section carries 30 marks.

- Here we want a basic 2D schematic like the one shown. You probably want to show what facilities there are in the various rooms. This part carries 15 marks. You should create this file in a the `svg` vector format and upload the vector file also.
- We are expecting something more interesting here. How about using your long hours in engineering drawing and making a 3D projection (e.g. isometric). Apart from this, at least one extra information should be shown (e.g., when clicked). For example, we'd like to see the whole building so that we know where the Table-Tennis Room is and where the Dance Room is. Or if you click a part of the image and something interesting should happen. See `http://www.cse.iitb.ac.in/page191?Building=KR&floor=0` and `http://www.cse.iitb.ac.in/page191?Building=KR&floor=1` for inspiration. This part carries another 15 marks.

6. Challenge question.

   Your assignment is complete without answering the challenge question. The challenge question will be graded after the regular questions, and only if all regular questions have been done with good effort. The deadline for this will be posted later, and this part carries an extra 50 bonus.

   (Not all challenge questions will carry bonus).

   All of you most likely have played sudoku and some of you might be experts! For those of you who are not familiar with the game see http://en.wikipedia.org/wiki/Sudoku. In this part we will try to make a variant of the game called Killer Sudoku. This has to built using HTML and Javascript.

   Have a look at :

- `https://en.wikipedia.org/wiki/Killer_sudoku`
- `http://www.killersudokuonline.com/`

Note that you don't have to generate a question every time a user comes. You have to store the sudoku given in the figure 4 below and "load" it.

This part carries 50 marks. How we will score you:

- A sudoku board where you can enter numbers (10 marks)
- Representing different cages (grouping of cells) by a dotted line or different colors and showing the sum of numbers in that cage (20 marks)
- Verifying the solution (This can be done on some button click) (20 marks)
- **Clearly** write the instructions for playing the game in the readme file. Inadequate instructions may attract penalty.
- Other cool things like music, reset, etc. (extra credit goes to your Karma. We will record it but if you don't do this you can still earn 100 marks out of 100)

The board should look something like this

Figure 4

Submission Guidelines

Here are some general principles in doing an assignment.

1. First, my belief is that you learn best when you do things, YOURSELF. You also learn when you do things in a group, but it becomes hard for me to give marks to figure out who has done what. So as a compromise, please work within the group, and not across a group.

2. When you submit, please document individual percentages such as Student 1: 80%, Student 2:100%, Student 3:10%. In this example, the second student will get full marks (10/10) (assuming that the submission covered all parts) and the first student will receive 8/10.

3. You are going to create a web page that documents your work in this task. However, *don't make it public*. Submit it on moodle as mentioned below. Use the ASC format for your roll number (i.e., with leading zero if applicable). Be sure to note that Moodle does not have infinite size, so prune your submission. DO NOT SEND JUNK FILES THAT ARE IRRELEVANT IN THE ASSIGNMENT (mac users especially to note). Any links within the web page should be to a local file – not absolute links. Also tgz the entire contents, but do not include any binary files.

4. Do include a readme.txt (telling me whatever you want to tell me). Do include group members (name, roll number), group number, honour code, citations etc.

5. The folder and its compressed version should both be named `lab02_groupXY_final` for example folder should be named `lab02_group07_final` and the related `tar.gz` should be named `lab02_group07_final.tar.gz`

How We will Grade You

The number of points per task appears below

1. Basic web page: 30
2. CSS: 30
3. Javascript: 10
4. Canvas: 20
5. Inkscape: 30
6. Killer Sudoku: 50 (bonus, extended deadline)

Missing/Incomplete readme etc. $\Rightarrow$ Will attract penalty points