CS 251: Talking the Web Lingo: Outlab

- Due: 06/08 10:00 PM Released 5:30 PM. 02/08
- 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 Piazza.

Overview

This outlab gives you a taste of some of the web technologies. We start with JavaScript (JS) to implement some dynamic functionality in web pages. We will also use JS to abort an XSS attack (Read about about XSS here to get an idea about the attack.) Our second task involves adding some beautification elements to the website so that it looks aesthetically elegant. No web page is complete without a picture, so while images are cool, we also want you to work with vector graphics.

This lab has 4 components. Add the pages you create for this outlab to the website you created in the inlab making sure that you add the relevant pages to the navbar.

- **Beauty**: Polishing your inlab pages by making them more 'beautiful'.
- **JS form validation**: You will have to create a HTML form (with some specified elements) and use JS for validation. We touch upon a well known web attack.
- **Accordions**: Accordions are collapsible CSS elements. Refer to the first example on this page. However, we expect you to implement a simplified version of the same. The goal of this part is to have a basic understanding of CSS and JS.
- **Inkscape**: We agree that the new CSE building is confusing. Help us and others by preparing a floor plan of the New CSE Building using Inkscape. Refer to this page for context. Keep in mind creativity elements – e.g. what is the different from the InstiMap on android and Google maps? We don't want all the stuff – just enough for people to get around. Where is the washroom? What is the seating capacity? Is the room wireless enabled?

Let's delve deeper!

Beautification

It is said that beauty is a state of the mind. We expect you to use some CSS theme to beautify the website you created for your inlab task. There are numerous free and open-source themes with Bootstrap and Materialize CSS being the most popular ones. You are free to use either one of them. If you're feeling adventurous, feel free to use a amalgam of both. A quick web search should direct you to their respective home pages.

Task A: Add the CSS wrapper to the 5 web pages from the inlab task. (Make the other outlab web pages also using the same wrapper.)

Assembly language of the web: JavaScript

The above is definitely not an understatement! JavaScript, commonly called JS, has become a fundamental part of web page design. JavaScript is a programming language that can be included on web pages to make them more interactive. One can use it to check or modify the contents of web pages to write dynamic page content. JS operates at the client side, e.g. browsers. There are other languages that support the server side operations, e.g. PHP (now considered old, but note that Facebook still uses a lot of PHP, albeit disguised)

Task B: Create an HTML form with a text field, a password field, an email field and other optional fields. Use the `POST` method to transfer the form data. The link for the form action will be released later and will be intimated to you through Piazza. On submitting the form, the server will check the correctness of the inputs and echo back all the values it received. You are not allowed to use JQuery or any external libraries for this. The code needs to be in native JavaScript. You will need to conform to the following specifications:

- The number of characters in the text field should be in the range 6-50, both inclusive. Use `"text"` as the `name` of this field
- The password should not be visible and thus suitably masked. It should be 8-20 characters in length with at least one special character from the set `{$, !, @, _}`. Use `"pass"` as the `name` of the element
- Use `"email"` as the `name` of the email field. The email address should conform to the following specifications:
  - It should be of the form **local-part@domain-part** where
  - *local-part* can contain only alphanumeric characters in addition to the special characters `{., _}` with length of the string >2
  - *domain-part* should contain only alphabetical characters other than a period(`"."`). There should be at least one period(`"."`). Each substring enclosed within the `"."` characters in this part should have at least 2 alpha-characters between them. The substring of domain-part before the first `"."` should have at least 3 alpha-characters

Having done this, you are expected to secure the form against the most basic type of XSS attack.

A simple XSS attack is when JS code is injected between the `<script>` and `</script>` tags. For example, if the browser receives `<script>alert("Hacked")</script>` as in input, then an alert box will pop up showing the message `Hacked`.

The problem arises because the code between the `script` tags is identified by the browser as JS and it executes that. A solution to this is to make the `script` tags lose their meaning, i.e., the browser does not treat them as structures (or 'structs') carrying code within them but instead just simple pieces of text. A simple way to do that is to encode special HTML characters. Figure out what they are and how to encode them in a HTML safe manner. Do not use any predefined functions for this. You have to write code for this.

Extra credits: Try implementing on the fly form validation as the user is filling out the form. Refer to the form on this page for example. You can earn up to 10 points for attempting this task. Try filling in fields of the form incorrectly to see the error messages.

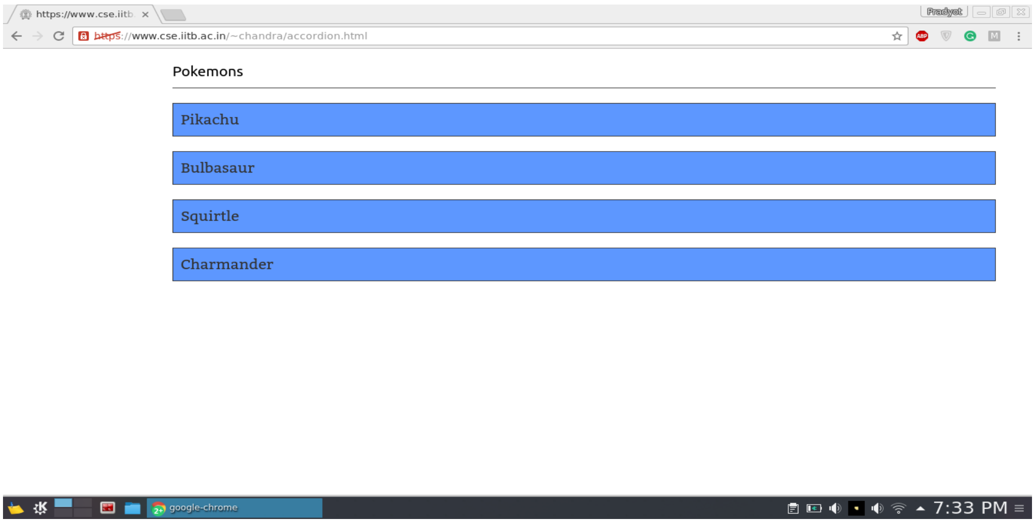
On the go: Accordion

Since Pokémon Go is literally on the streets these days, let's walk in that direction.

Task C: Create a self-stylized accordion with the following specifications:

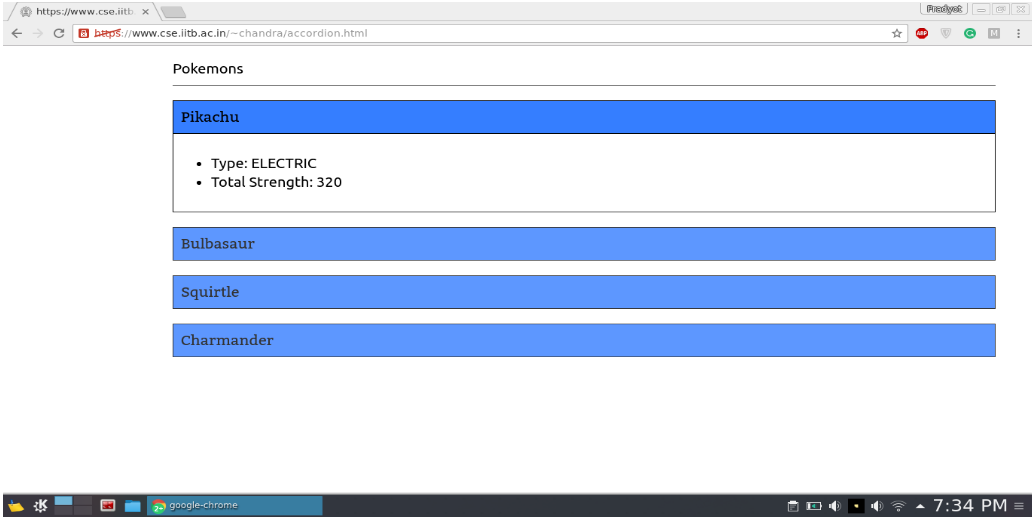
- Create at least 4 collapsible tiles
- Header of each section should contain the name of a Pokémon
- On expanding the section, the type, final evolution state of the Pokémon (if any) and four most common attacks of the Pokémon should be made visible

Just to reiterate, you are not allowed to use any existing themes or templates for this. Read the references provided and figure out how to go about doing this. Here are some screen shots for your reference. (Note: If you have no idea about Pokemons you can use your the 4 largest metros. Interpret 'attack' and 'evolution state' in your own way but be prepared to defend your interpretation.)



Pokemons

- Pikachu
- Bulbasaur
- Squirtle
- Charmander



Pokemons

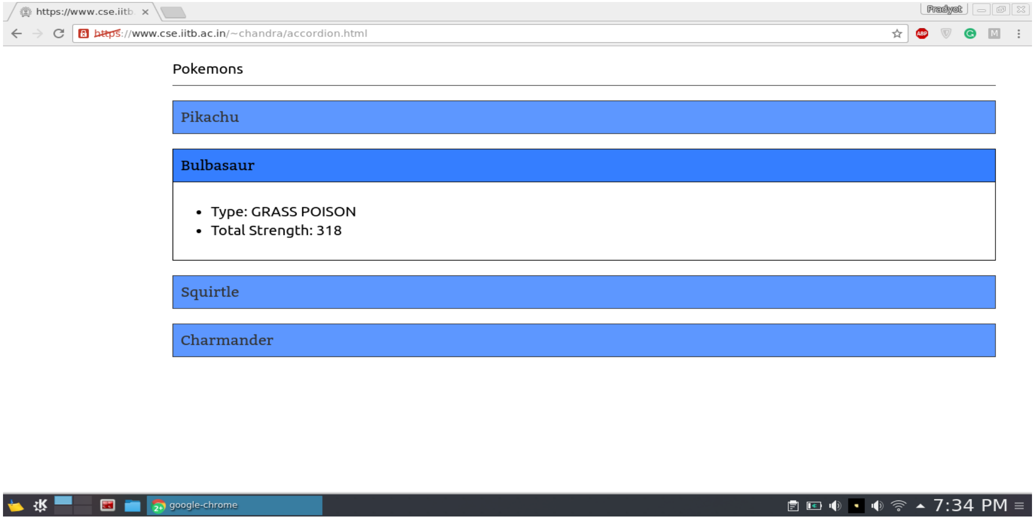
Pikachu

- Type: ELECTRIC
- Total Strength: 320

Bulbasaur

Squirtle

Charmander



Pokemons

Pikachu

Bulbasaur

- Type: GRASS POISON
- Total Strength: 318

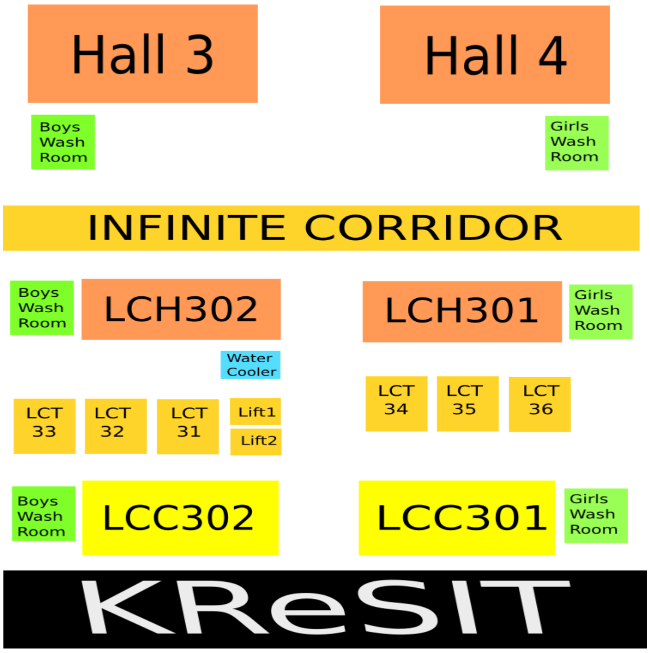
Squirtle

Charmander

Draw freely: Inkscape

The floor plan of the New CSE Building can be found here. The edifice contains 6 major floors namely basement, ground floor, first floor, ..., fifth floor. Let's annotate the basement floor by the number -1, ground floor by 0, and the successive floors by their numeric values. The map on the CSE homepage has lot of details. We want a simplified map of the floors with the major rooms drawn using Inkscape. You can omit the staircases and other structures that do not contribute to the blueprint of the floor. Do not worry much about the beautification of your layout; instead focus on the overall layout.

Task D: If your group number is `X` then you need to work out the floor plan of floor `X%7 - 1` using Inkscape. Embed the image as a SVG file into an HTML page. Look at the following image from one of the earlier course offerings.



Extra credit: Inkscape lets us edit and modify PDF files, a task very commonly desired, in addition to creating vectored images.

Go to this url to access one of the forms being used by IIT Bombay for summer course registration. The `Semester` section is hard-coded to `3`. We want you to change the number by a comma-separated list of numbers `(3, 5, 7)` which the student can change based on his/her current semester. Also, figure out a way to add your team name as a watermark into the same PDF using Inkscape. Upload the modified PDF. Make sure that you have properly annotated the updated page so that we can spot your changes.

You can go all mercenary and look for more such forms with some mistakes that you think can be corrected/generalized. Each correction will fetch you 2 additional points with your maximum points being upper-bounded by 10, i.e., at most 5 such tasks. For us to believe that you actually did the task, provide a link to the original PDF too.

Submission Guidelines

1. 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) and the first student will receive 8/10.

2. Submit the HTML, CSS and JS files for all the web pages along with the images that you have used in them.

3. For the PDF editing task, label the edited form as `IITBForm.pdf`

4. For the PDF editing bonus task, label the files as `Fi.pdf` and their corresponding edits as `Fi_edit.pdf` where i the index number of the file. Add these files in the folder `pdf_bonus`

5. If you have attempted either of the two bonus tasks mention that in the readme.txt

6. 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. This is the place for the reflection essay.

7. The folder and its compressed version should both be named `lab02_groupXY_outlab` for example folder should be named `lab02_group07_outlab` and the related `tar.gz` should be named `lab02_group07_outlab.tar.gz`

How We Will Grade You

The specific number of points per task (A, B, C, D) appears below:

1. Adding the CSS wrapper to the website will fetch you up to 20 points. If you feel good about your website, then you shouldn't have any problem earning the 20 points

2. Task B carries a total of 40 points. You will be awarded 30 points for for completing the basic form validation tasks. Another 10 points will be awarded for the XSS attack prevention task

3. The task on accordion carries a total of 10 points: 2.5 points each for each of the 4 collapsible tiles

4. The final 30 points can be earned by doing Task D

On attempting the first extra credit task(form validation), you can earn an additional 10 points. For the second task, you can earn up to a maximum of 10 additional points. In essence, you can earn up to 20 points by attempting additional tasks.