

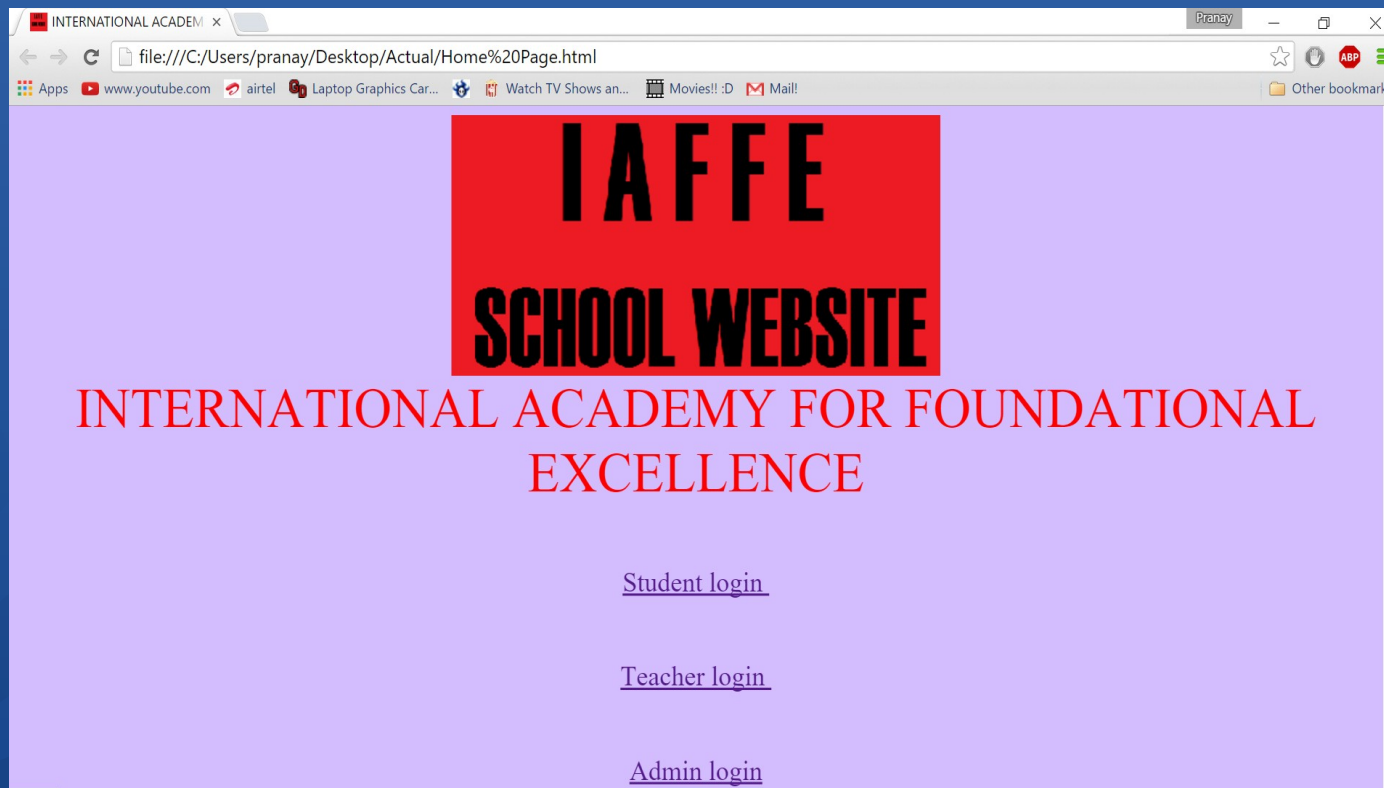
SCHOOL INTRANET

# COMPUTER SCIENCE PROJECT

## Grade 9 – Pranay Venkatesh

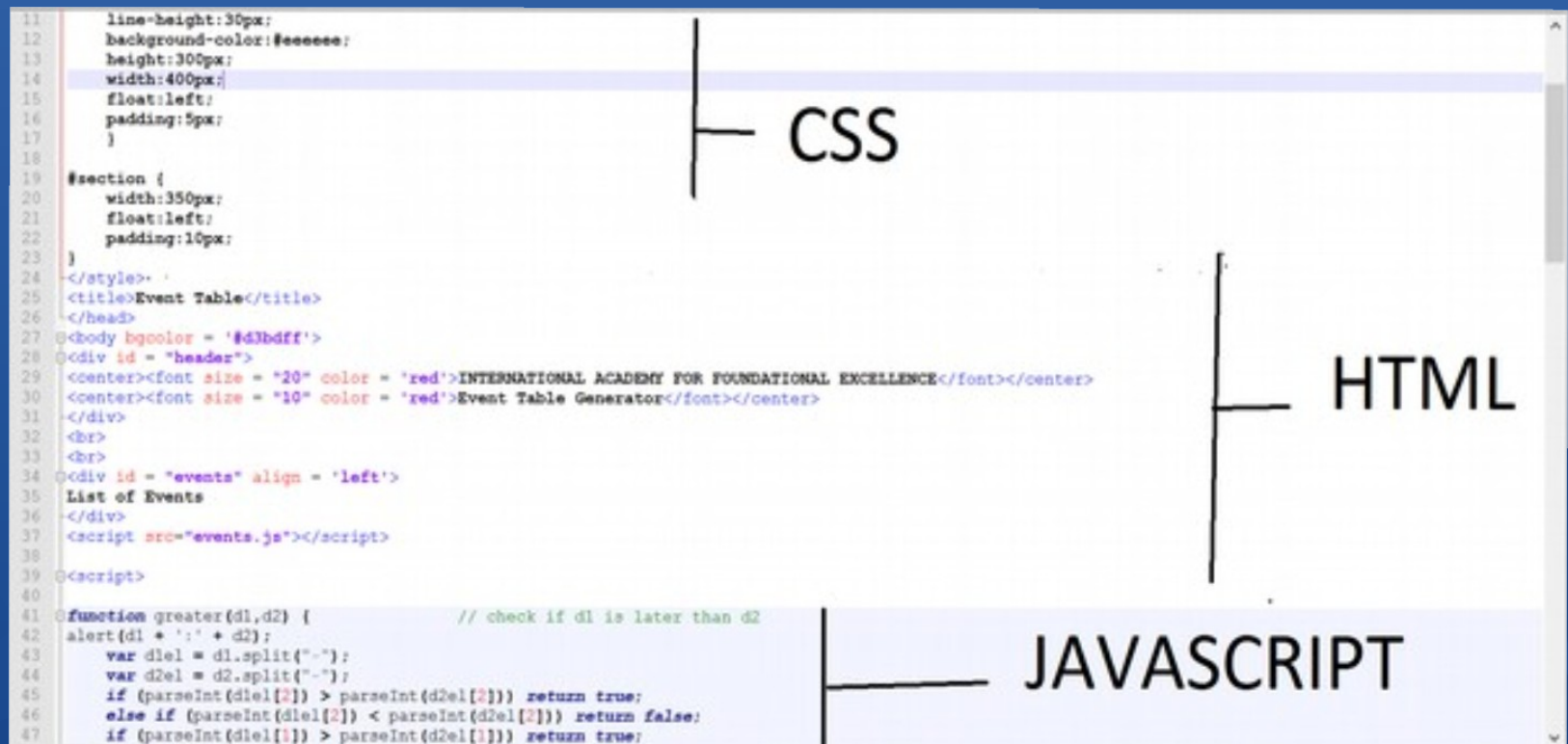
# Topic

- Topic- Creating a website in html
- My website- A school website which allows students, teachers, administrators and visitors to view specific information.



# Tools

- The tools used for this project are-
- HTML
- JAVASCRIPT
- CSS



The image shows a screenshot of a code editor with three sections of code. Handwritten labels with lines pointing to the code are present:

- CSS**: Points to the CSS code block (lines 11-23).
- HTML**: Points to the HTML code block (lines 24-39).
- JAVASCRIPT**: Points to the JavaScript code block (lines 40-47).

```
11  line-height:30px;
12  background-color:#eeeeee;
13  height:300px;
14  width:400px;
15  float:left;
16  padding:5px;
17  }
18
19  #section {
20    width:350px;
21    float:left;
22    padding:10px;
23  }
24  </style>
25  <title>Event Table</title>
26  </head>
27  <body bgcolor = '#d3bdf'>
28  <div id = "header">
29    <center><font size = "20" color = 'red'>INTERNATIONAL ACADEMY FOR FOUNDATIONAL EXCELLENCE</font></center>
30    <center><font size = "10" color = 'red'>Event Table Generator</font></center>
31  </div>
32  <br>
33  <br>
34  <div id = "events" align = 'left'>
35    List of Events
36  </div>
37  <script src="events.js"></script>
38
39  <script>
40
41  function greater(d1,d2) {           // check if d1 is later than d2
42    alert(d1 + ':' + d2);
43    var d1el = d1.split("-");
44    var d2el = d2.split("-");
45    if (parseInt(d1el[2]) > parseInt(d2el[2])) return true;
46    else if (parseInt(d1el[2]) < parseInt(d2el[2])) return false;
47    if (parseInt(d1el[1]) > parseInt(d2el[1])) return true;
```

# Data Structures

- Session: Is a day + period (e.g. Mon 1st period)
- Classroom: Is a Class + Section (e.g. IV A)
- Vector: Is a 32 bit vector, each bit represents a subject – '0' if unavailable, '1' if available.
- Allocation: A vector with exactly one '1'
- Table: Is a  $M \times N$  array of vectors;  $M$  = max no of sessions;  $N$  = max no of classrooms
- Current Table: Working table
- Choice Tables/splitting: Available choices to be explored.
- Stack of choices: Stack of choice tables.

# Time Table

- The website can perform 1 major process- The formation of a Time Table.
- The time table below lists out the allotment of subjects for each period of all 4 classes in the school >>>MAKING

## Time Table

	Monday							Tuesday							Wednesday							Thursday							Friday						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
9A	MA	BI	PH	EN	CH	EN	MA	EN	PH	PH	LA	CH	CH	CH	PH	CH	MA	MA	BI	EN	EN	BI	MA	MA	EN	CH	EN	PH	MA	CH	PH	BI	CH	HI	CH
8A	HI	EN	CH	PH	BI	MA	EN	MA	HI	EN	PH	EN	HI	MA	EN	BI	EN	PH	MA	MA	CH	CH	HI	LA	LA	HI	HI	HI	BI	HI	MA	MA	LA	LA	HI
7A	PH	HI	EN	MA	MA	BI	CH	LA	LA	HI	EN	MA	EN	HI	MA	MA	CH	HI	EN	PH	BI	EN	EN	HI	MA	MA	PH	BI	CH	MA	HI	EN	MA	EN	LA
6A	EN	MA	MA	BI	EN	CH	PH	HI	MA	LA	HI	HI	LA	BI	HI	EN	BI	EN	CH	HI	MA	MA	LA	PH	HI	EN	MA	LA	EN	LA	LA	CH	HI	PH	MA

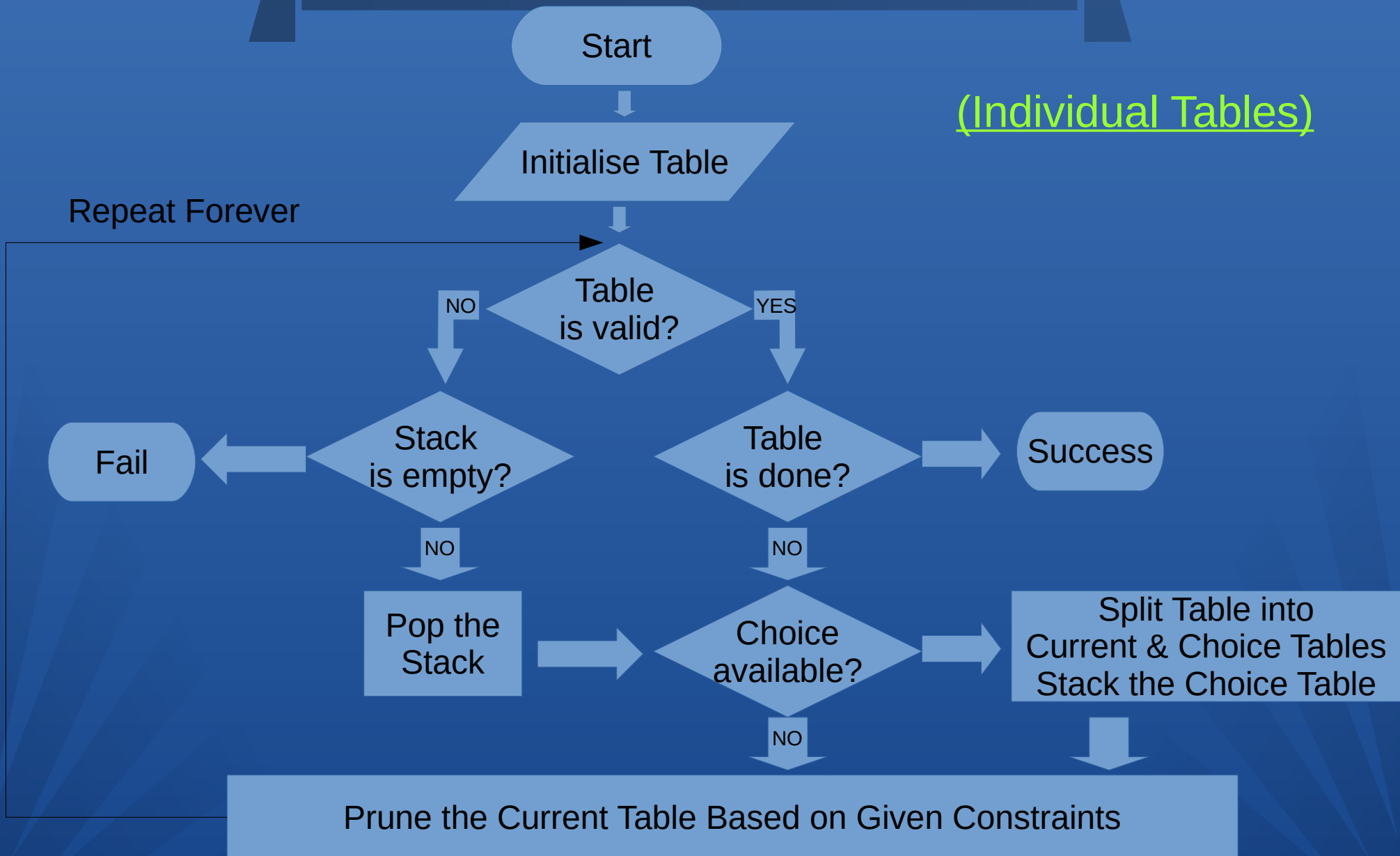
☐ Keep ☐ Get Another

# Making of time table

- The time table is made using a simple algorithm.
- Based on inputs entered by the administrator – i.e. the teacher of each subject for each class, the subjects each class has, and the number of hours for the teacher and the subject, it forms the master table.
- The flowchart following this slide describes the process of creating the time table. [\(flowchart\)](#)

# Creation process

(Individual Tables)



# Individual tables

- Student Time Table: [\(go back\)](#)
- Scan one row of the Table data structure and format the entries into a 2D table with periods as columns and days as rows
- Teacher Time Table:
  - Scan one column of the Table data structure. There can be at most one entry for any of the subjects that is allocated to the teacher.
  - Time tables are in the form of **one html file** for each teacher and one for each class

**Time Table for Seetha**

	1	2	3	4	5	6	7
<b>Monday</b>	6A/HI	7A/HI	Free	6A/HI	7A/HI	6A/HI	8A/HI
<b>Tuesday</b>	6A/HI	Free	Free	Free	6A/HI	Free	Free
<b>Wednesday</b>	Free	Free	Free	Free	9A/HI	8A/HI	Free
<b>Thursday</b>	9A/HI	7A/HI	8A/HI	7A/HI	7A/HI	6A/HI	6A/HI
<b>Friday</b>	6A/HI	Free	Free	8A/HI	7A/HI	8A/HI	9A/HI

**Time Table for class 9A**

	1	2	3	4	5	6	7
<b>Monday</b>	MA	EN	CH	MA	PH	BI	BI
<b>Tuesday</b>	EN	CH	MA	PH	MA	BI	EN
<b>Wednesday</b>	BI	CH	EN	MA	HI	PH	EN
<b>Thursday</b>	HI	BI	PH	BI	MA	BI	LA
<b>Friday</b>	EN	CH	MA	BI	EN	PH	HI



# Vector

- The vector is an integer having 32 bits, each bit having a value of 1 or 0.
- The first bit is used to identify that the session has one subject allocated.
- The other 31 bits are used to identify whether a subject is available or not.
- 1 means that the subject is available and 0 means unavailable. (go back)

Unallocated

```
0100001000101000
0101000101001100
```

Allocated

```
1000000000000000
0000000000000000
```

# Splitting

- In a cell where the available choices is more than 1, the table gets split into the current and choice table.
- 1 value is picked randomly from a cell with the least number of choices and allocated to the cell in the current table.
- In the choice table, the remaining options are left and the table is pruned.
- (example)

# Example of Splitting

Unallocated Current Table entry

Unallocated

0 0 0 0 ... 1 1 0 1 1 1 1

SPLIT / MODIFICATION



Allocated Current Table entry

Allocated

1 0 0 0 ... 0 0 0 0 0 1 0

Choice Table entry

Unallocated

0 0 0 0 ... 1 1 0 1 1 0 1

[\(go back\)](#)

# Pruning

- In order to meet the given constraints, the function “prune()” is applied.
- “prune()” eliminates choices which do not meet constraints.
- For example, prune teacher constraints makes sure that the teacher can only teach one class per session.

```
295 function prune(cr,sn) { // prunes the table entries based on a set of constraints
296     if (currTable.getDone(cr,sn)) return;
297     currTable.setDone(cr,sn);
298     teacherConstraint(cr,sn); // apply the teacher constraints
299     teacherTooManyClassesConstraint(cr,sn); // if teacher has been teaching continuously for some time
300     subjectDoneConstraint(cr,sn); // check if the number of hours/week of the subject is done
301     // sameSubjectRepeatConstraint(cr,sn); // Same subject can't repeat more than twice per day
302     sameSubjectDistributedConstraint(cr,sn);
303     // Same subject can't repeat in neighbouring days ("6A"h/wk subjects)
304 }
305
306 function teacherConstraint(cr,sn) { // teacher can be allocated to only one classroom per session
307     var subject = findSubject(currTable.get(cr,sn)); // get the single subject allocated to cell
308     var teacher = Allocation.allocTable[cr][subject].teacher;
309     // find teacher associated with this class/subject
310
311     var tSubjects = Allocation.subjects(teacher);
312     // tSubjects will hold the subjects taught by this teacher
313
314     for (var i=0; i<numCRs; i++) { // remove any subject allocated to this teacher from the session
315         if (i == cr) continue; // don't do anything if it is the current classroom
316         currTable.set(i,sn,currTable.get(i,sn) & (~tSubjects)); // remove tSubjects from the cell
317         if (singleChoice(currTable.get(i,sn))) prune(i,sn); // prune from cell <i,sn>
318     }
319 }
```

# Event List

- Event list is entered by an administrator.

<b>List of Events</b>  20-01-2016 : UT  19-1-2016 : UT-1  17-1-2016 : UT-1	<table><tr><td rowspan="5">EVENT</td><td>Event Name:</td><td>UT-1</td></tr><tr><td>Event Date</td><td>17/01/2016</td></tr><tr><td>Teacher Responsible</td><td>Kunti</td></tr><tr><td>Class Involved</td><td>9A</td></tr><tr><td colspan="2"><input type="button" value="Confirm"/> <input type="button" value="Reset"/></td></tr></table>	EVENT	Event Name:	UT-1	Event Date	17/01/2016	Teacher Responsible	Kunti	Class Involved	9A	<input type="button" value="Confirm"/> <input type="button" value="Reset"/>	
EVENT	Event Name:		UT-1									
	Event Date		17/01/2016									
	Teacher Responsible		Kunti									
	Class Involved		9A									
	<input type="button" value="Confirm"/> <input type="button" value="Reset"/>											

Events are then allotted to classrooms and teachers

## List of Events for class 9A

20-01-2016:UT  
17-1-2016:UT-1  
17-01-2016:UT-1

**Time Table :**

## List of Events for Kunti

17-01-2016:UT-1

**Time Table**

# Logins

- There are three basic logins for 3 different purposes and 3 different kinds of users.
- Teacher login – Teachers alone can see their events and time table.
- Student Login – Students can see their class time table and class event list.
- Administrator Login – The administrators can generate time tables and event lists.

Class:	<input type="text"/>
Password:	<input type="password"/>
<input type="button" value="Login"/>	<input type="button" value="Reset"/>

# Summary

- The website allows 3 logins – student, teacher, admin
- The admin can generate global event lists and time tables.
- These can be viewed individually by teachers and students of a class as a whole.

# THANK YOU

