

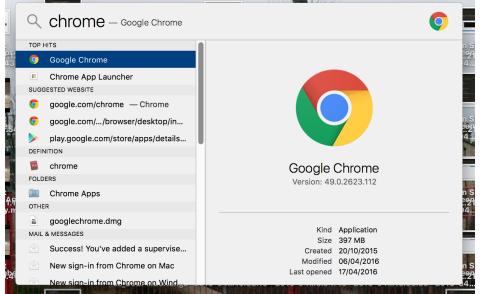
**DOCUMENTATION**

1: Download and installation .....	2
2: Instructions.....	7
2.1: Starting the rails app.....	7
2.2: Navigation.....	9
2.2.1: Landing page.....	9
2.2.2: General Navigation.....	10
2.2.2: Dashboard.....	13
2.2.3: All Students page.....	17
2.2.4: Upload File page.....	18
2.2.5: View Files page.....	19
2.2.6: Seating Plan Generator.....	20
3: Troubleshooting.....	27
3.1: Issues & Solutions .....	27
3.2: Installing xCode Command Line Tools .....	28
4: Glossary.....	29

# 1: Download and Installation

On Mac, on your keyboard press the keys cmd + spacebar to search for your web browser. E.G Chrome, Safari etc. And open Browser.

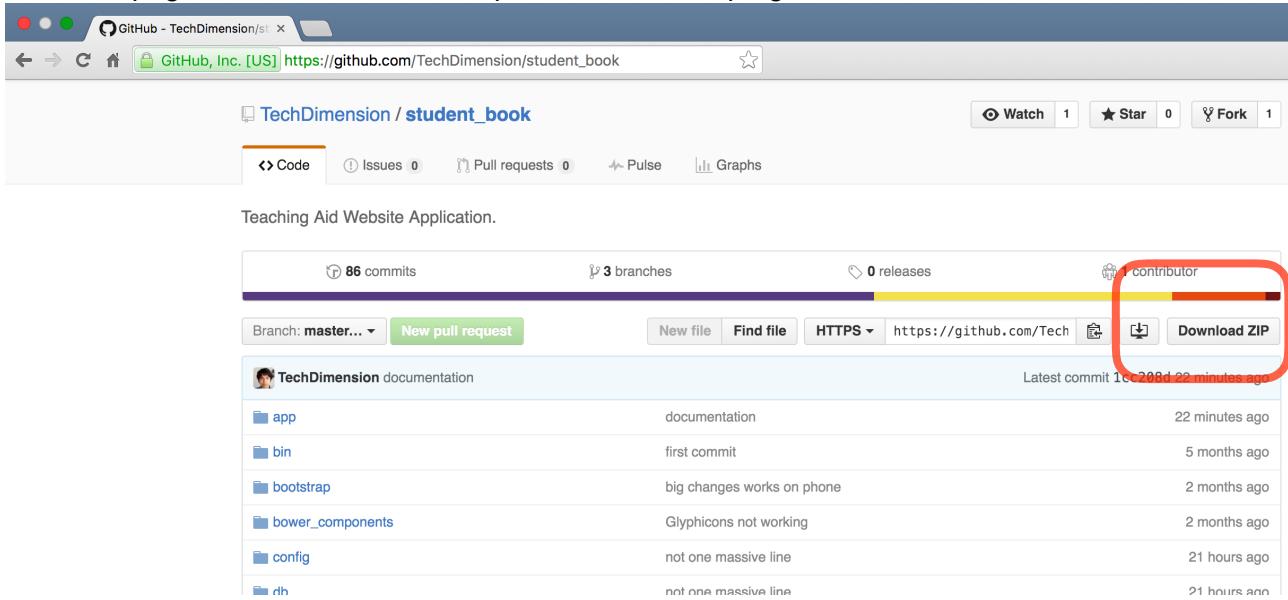
Or Find Manually and open browser.



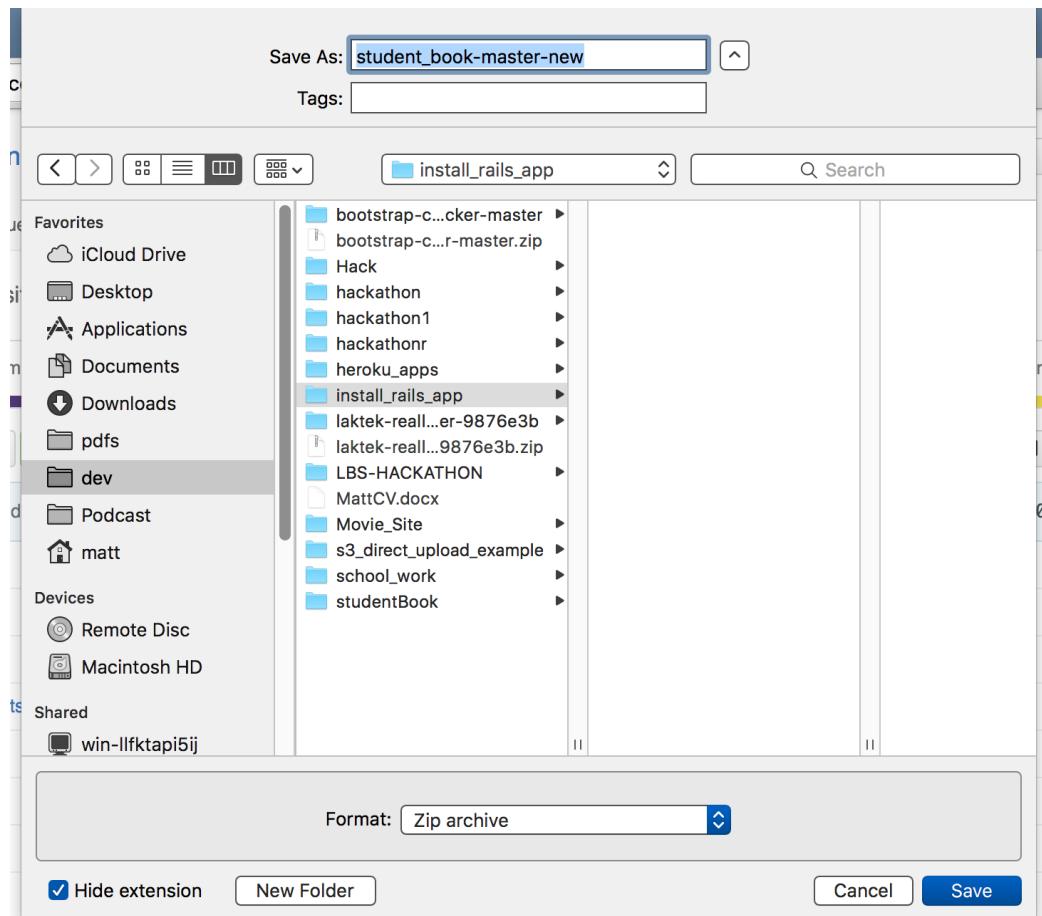
Open your preferred browser, and enter this url.  
[https://github.com/TechDimension/student\\_book](https://github.com/TechDimension/student_book)



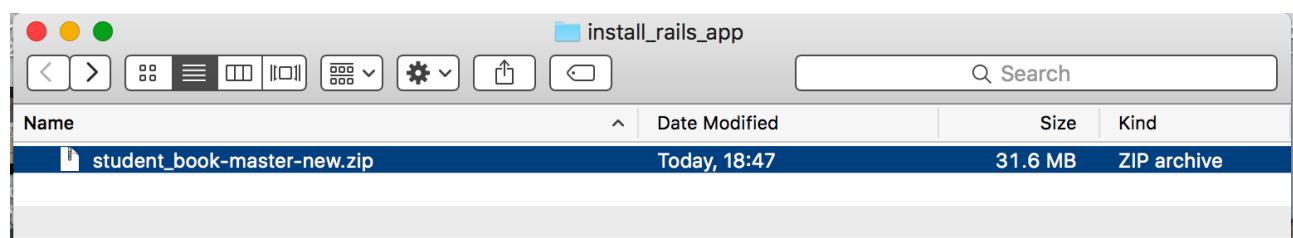
Enter the page, and click download Zip located at the top right .



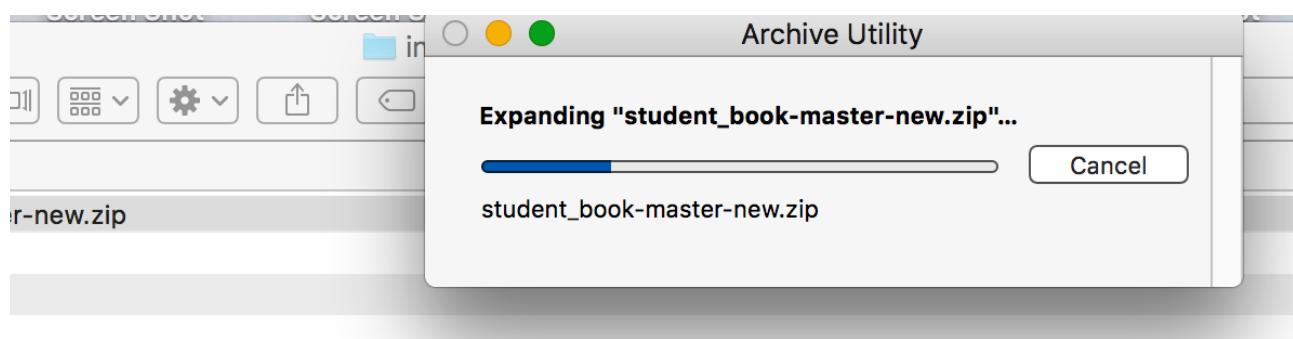
You can decide to save ZIP file as whatever you want, as long as you remember where you place it.

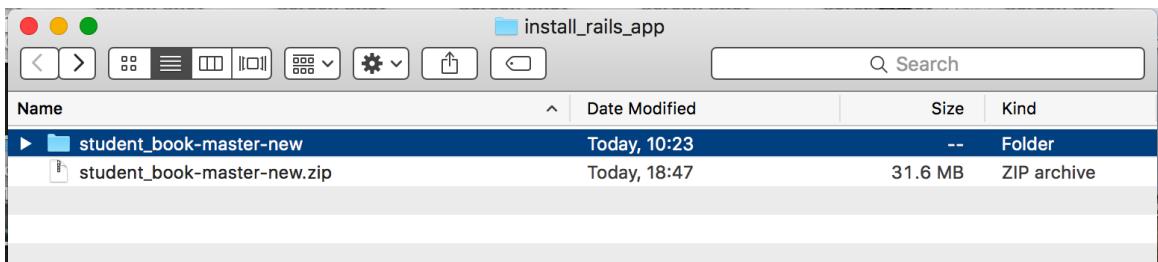


Navigate in your file browser to the ZIP file

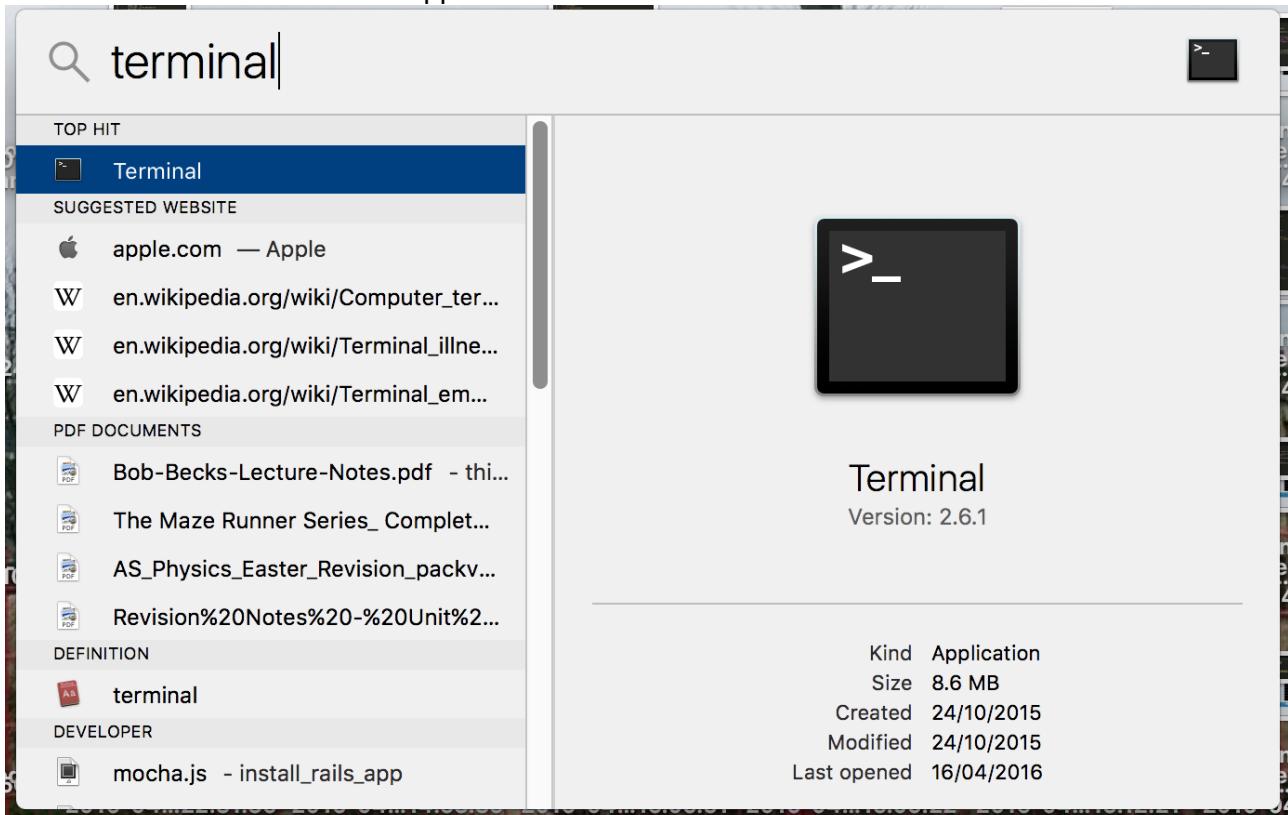


Double click the ZIP file to extract the data to the current folder.





Search for the Mac OS X Terminal application by pressing the Command-Spacebar combination (which Apple calls "Spotlight Search") and searching for "Terminal." Or look in the Applications/ Utilities/ folder for the Terminal application.



In your terminal, locate the folder where you installed the rails app.

```
[Matthews-MattBook-Pro:~ matt$ ls
Applications          Movies           coursework-pics.jpg
Desktop              Music            furnacha
Documents             Pictures         vehicles
Downloads            Public           vehicles.sqlpro
Library              Sites
[Matthews-MattBook-Pro:~ matt$ cd documents/dev/install_rails_app/
[Matthews-MattBook-Pro:install_rails_app matt$ ls
student_book-master-new      student_book-master-new.zip
[Matthews-MattBook-Pro:install_rails_app matt$ cd student_book-master-new
Matthews-MattBook-Pro:student_book-master-new matt$ gem install rails]
```

Next type in Console.  
**gem install rails**

```
Matthews-MattBook-Pro:student_book-master-new matt$ gem install rails
Fetching: activesupport-4.2.6.gem (100%)
Successfully installed activesupport-4.2.6
Fetching: actionview-4.2.6.gem (100%)
Successfully installed actionview-4.2.6
Fetching: actionpack-4.2.6.gem (100%)
Successfully installed actionpack-4.2.6
Fetching: activejob-4.2.6.gem (100%)
Successfully installed activejob-4.2.6
Fetching: actionmailer-4.2.6.gem (100%)
Successfully installed actionmailer-4.2.6
Fetching: activemodel-4.2.6.gem (100%)
Successfully installed activemodel-4.2.6
Fetching: activerecord-4.2.6.gem (100%)
Successfully installed activerecord-4.2.6
Fetching: railties-4.2.6.gem (100%)
Successfully installed railties-4.2.6
Fetching: rails-4.2.6.gem (100%)
Successfully installed rails-4.2.6
Parsing documentation for activesupport-4.2.6
Installing ri documentation for activesupport-4.2.6
Parsing documentation for actionview-4.2.6
Installing ri documentation for actionview-4.2.6
Parsing documentation for actionpack-4.2.6
Installing ri documentation for actionpack-4.2.6
Parsing documentation for activejob-4.2.6
Installing ri documentation for activejob-4.2.6
Parsing documentation for actionmailer-4.2.6
Installing ri documentation for actionmailer-4.2.6
Parsing documentation for activemodel-4.2.6
Installing ri documentation for activemodel-4.2.6
Parsing documentation for activerecord-4.2.6
Installing ri documentation for activerecord-4.2.6
Parsing documentation for railties-4.2.6
Installing ri documentation for railties-4.2.6
Parsing documentation for rails-4.2.6
Installing ri documentation for rails-4.2.6
```

After gem install rails is complete. Now type in console.  
**Bundle install.**

```
Matthews-MattBook-Pro:student_book-master-new matt$ bundle install
Using rake 11.1.2
Using i18n 0.7.0
Using json 1.8.3
Using minitest 5.8.4
Using thread_safe 0.3.5
Using builder 3.2.2
Using erubis 2.7.0
Using mini_portile2 2.0.0
Using rack 1.6.4
Using mime-types 2.6.2
Using arel 6.0.3
Using execjs 2.6.0
Using bcrypt 3.1.10
Using debug_inspector 0.0.2
Using sass 3.4.22
Using byebug 6.0.2
Using choice 0.2.0
Using coffee-script-source 1.9.1.1
Using thor 0.19.1
Using commonjs 0.2.7
Using concurrent-ruby 1.0.1
Using multi_json 1.11.2
Using tilt 2.0.2
Using libv8 3.16.14.13
Using pg 0.18.4
Using bundler 1.11.2
Using rails-assets-adminlte 2.3.2
Using ruby-graphviz 1.2.2
Using rails_serve_static_assets 0.0.5
Using rails_stdout_logging 0.0.4
Using rdoc 4.2.0
Using ref 2.0.0
Using spring 1.3.6
Using sqlite3 1.3.10
Using tzinfo 1.2.2
Using nokogiri 1.6.7.2
Using rack-test 0.6.3
Using mail 2.6.3
Using autoprefixer-rails 6.3.3
Using uglifier 2.7.2
Using binding_of_caller 0.7.0
Using bootstrap-sass 3.2.0.0
Using font-awesome-sass 4.5.0
Using coffee-script 2.4.1
Using less 2.6.0
Using sprockets 3.6.0
Using rails_12factor 0.0.3
Using sdoc 0.4.1
Using therubyracer 0.12.2
Using activesupport 4.2.5
Using loofah 2.0.3
Using rails-deprecated_sanitizer 1.0.3
Using globalid 0.3.6
```

Next type in console

```
rake db:schema:load
```

```
----  
Matthews-MattBook-Pro:student_book-master-new matt$ rake db:schema:load  
-- create_table("groups", {:force=>:cascade})  
  -> 0.0043s  
-- create_table("students", {:force=>:cascade})  
  -> 0.0009s  
-- add_index("students", ["group_id"], {:name=>"index_students_on_group_id"})  
  -> 0.0121s  
-- create_table("user_files", {:force=>:cascade})  
  -> 0.0010s  
-- create_table("users", {:force=>:cascade})  
  -> 0.0009s  
-- add_index("users", ["email"], {:name=>"index_users_on_email", :unique=>true})  
  -> 0.0008s  
-- initialize_schema_migrations_table()  
  -> 0.0050s  
-- create_table("groups", {:force=>:cascade})  
  -> 0.0013s  
-- create_table("students", {:force=>:cascade})  
  -> 0.0009s  
-- add_index("students", ["group_id"], {:name=>"index_students_on_group_id"})  
  -> 0.0009s  
-- create_table("user_files", {:force=>:cascade})  
  -> 0.0009s  
-- create_table("users", {:force=>:cascade})  
  -> 0.0008s  
-- add_index("users", ["email"], {:name=>"index_users_on_email", :unique=>true})  
  -> 0.0018s  
-- initialize_schema_migrations_table()  
  -> 0.0050s  
Matthews-MattBook-Pro:student_book-master-new matt$ █
```

Good work! If everything completed successfully.

You have now finished downloading and installing my application!

If any errors occur during this installation, please refer to the troubleshooting section in the documentation for help.

## 2: Instructions

### 2.1: Starting your Rails Application.

Once the Application has installed, you are ready to run the application.

In your terminal, access where you installed the rails application

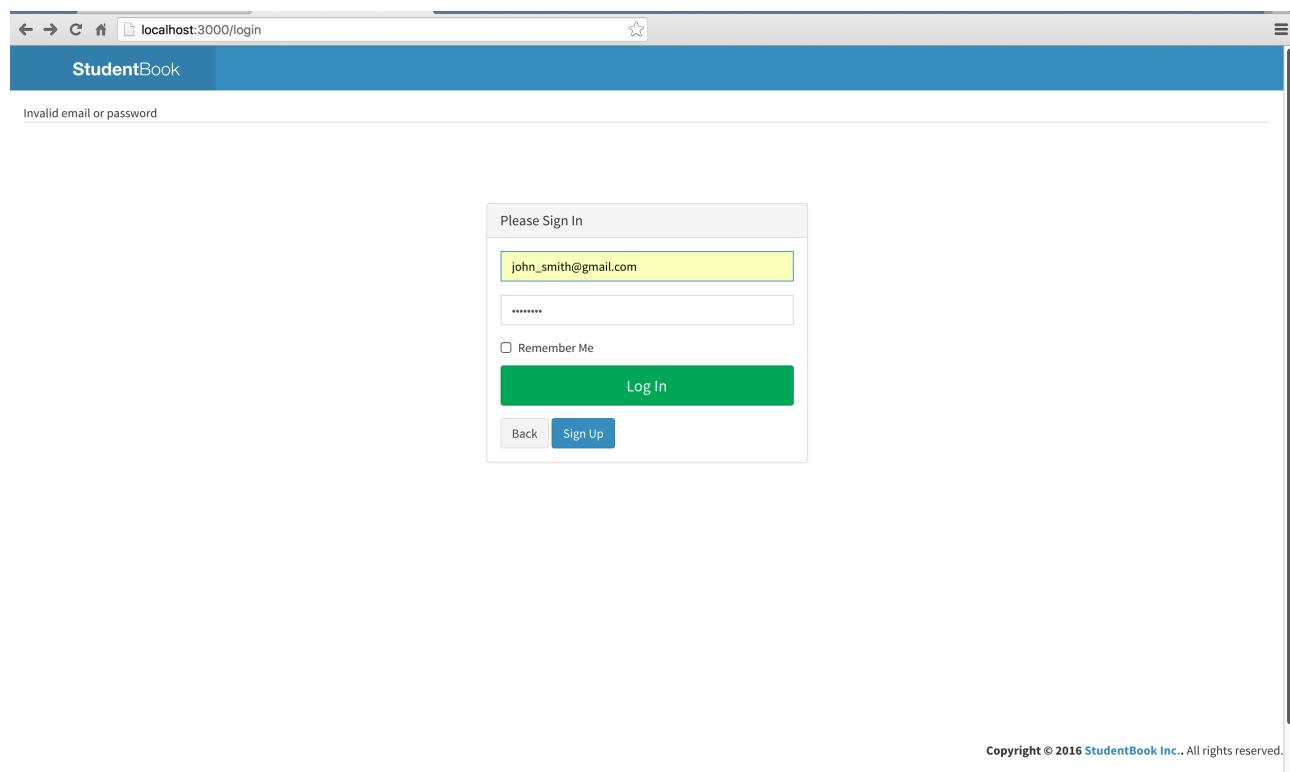
```
type  
rails server  
in chrome address go to  
localhost:3000
```

if you want to have your server run on a local network type

rails server -b '#YOURLOCALIP(E.g 10.20.0.195)'

And on the device address search for

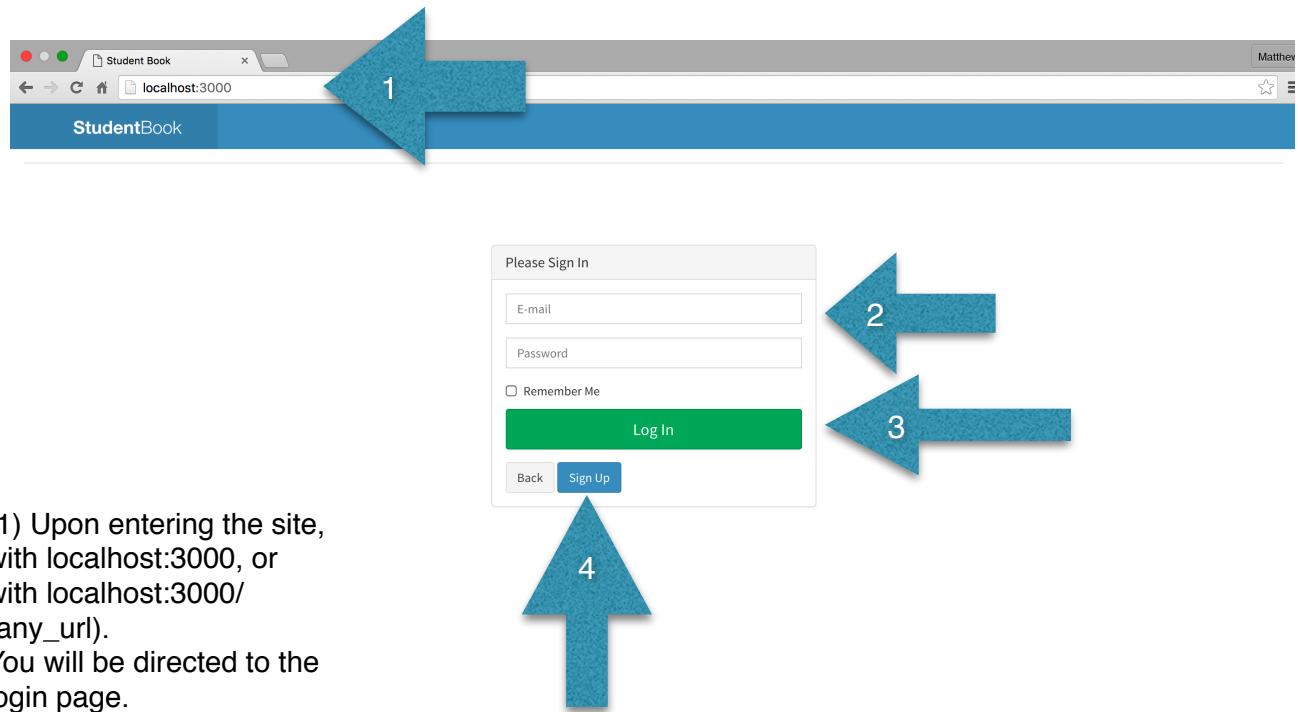
YOURLOCALIP:3000 where 3000 is the default port.



Great, you have run your Rails App. With a Fresh Database, for you to use.

## 2.2: Navigation

### 2.2.1: Landing Page



(1) Upon entering the site, with localhost:3000, or with localhost:3000/(any\_url). You will be directed to the login page.

(2) Features a sign in panel, where clicking on a field, will highlight around it, and allow you to type.

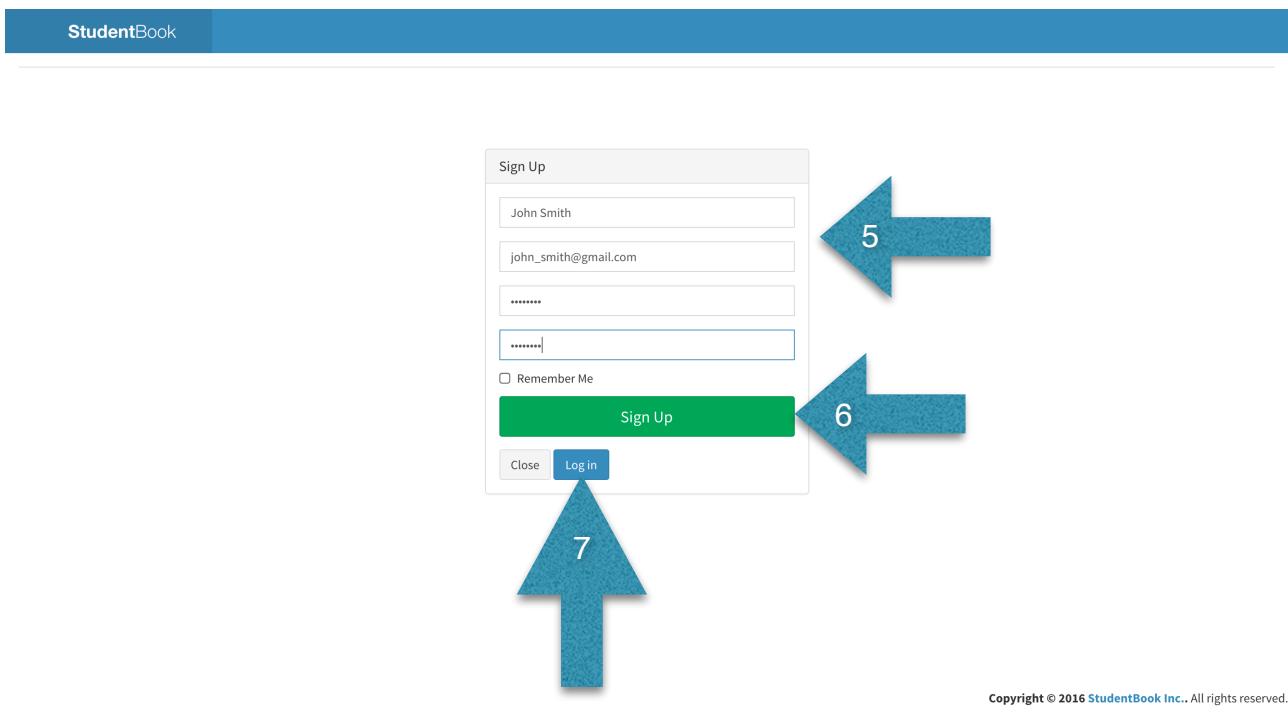
Email field requires a Valid Email address used to sign up, to the website.

Password must also be the password used to sign up to the website.

(3) Clicking on Login, will submit the data located in the form fields, and allow you to log in, if the data entered is correct.

(4) Clicking on sign up, will direct you to the sign up page, click this if you do not already have an account on the website.

Copyright © 2016 StudentBook Inc. All rights reserved.



Copyright © 2016 StudentBook Inc.. All rights reserved.

(5) Features a Sign Up panel, clicking on a field will highlight blue around it, and allow you to enter your details in it.

No fields can be left empty,  
Email must be a valid email address.  
Password must be more than 7  
characters, and must match with the  
password confirmation.

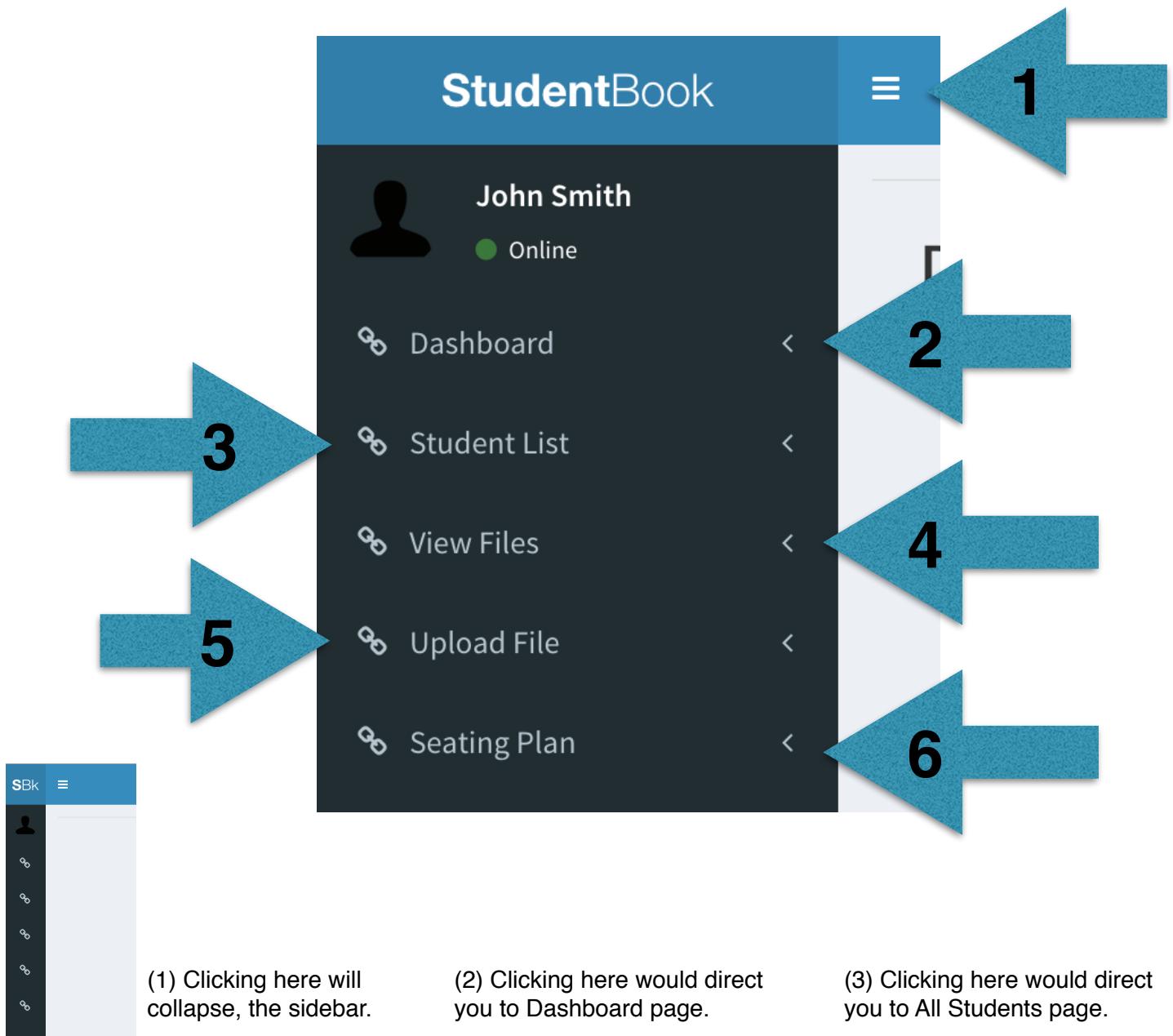
(6) Clicking on Sign Up, will submit the data located in the form fields, and allow you to create an account on the website if the data entered is correct.

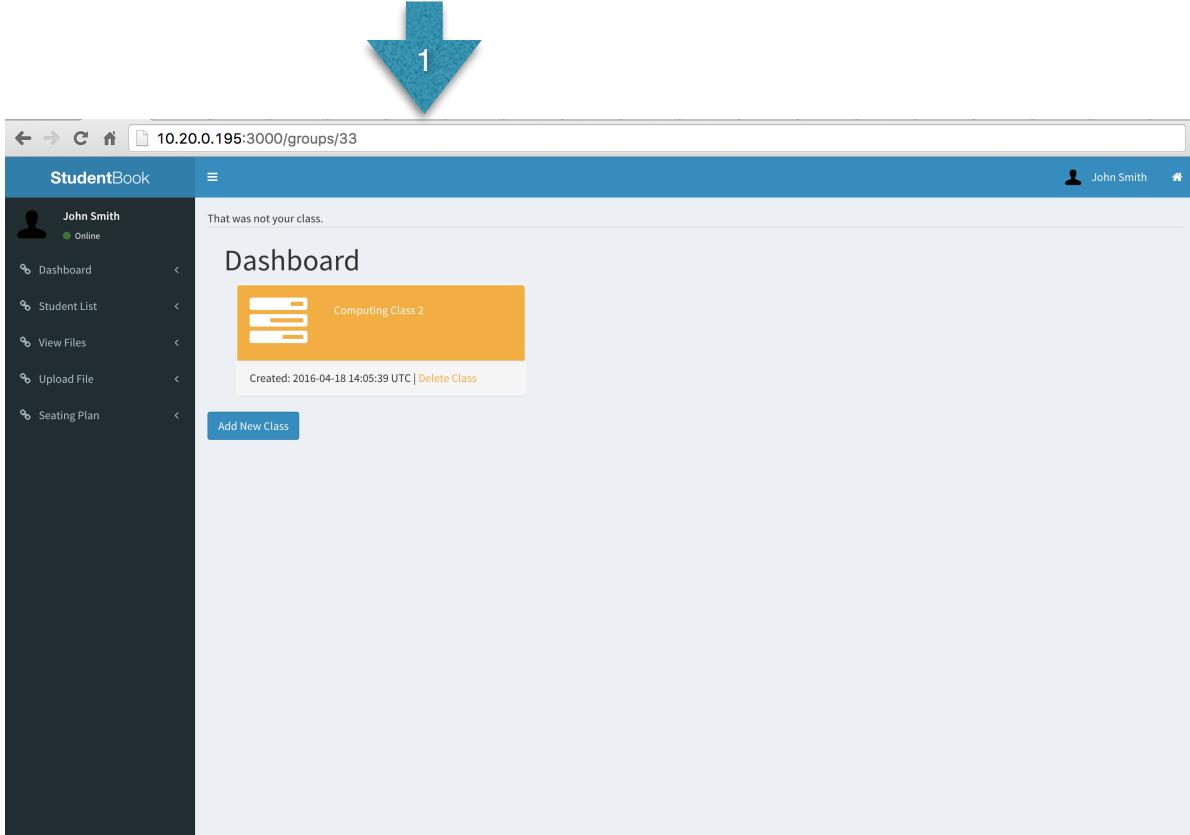
After an Account is made, you will be directed to the login page, in order to log in with your just made log in details

An account must be created in order to access any parts of the website.

(7) Clicking on the Log In button, will direct you to the login main page, in order to log in , if you already have an account made on the website.

## 2.2.2: General Navigation



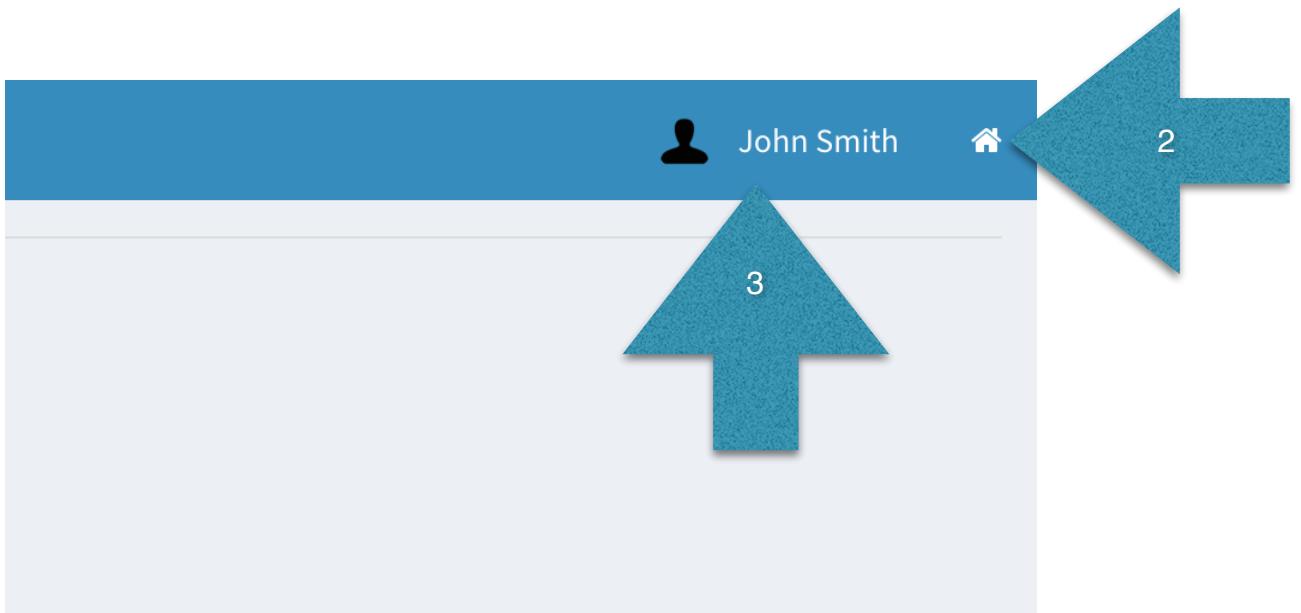


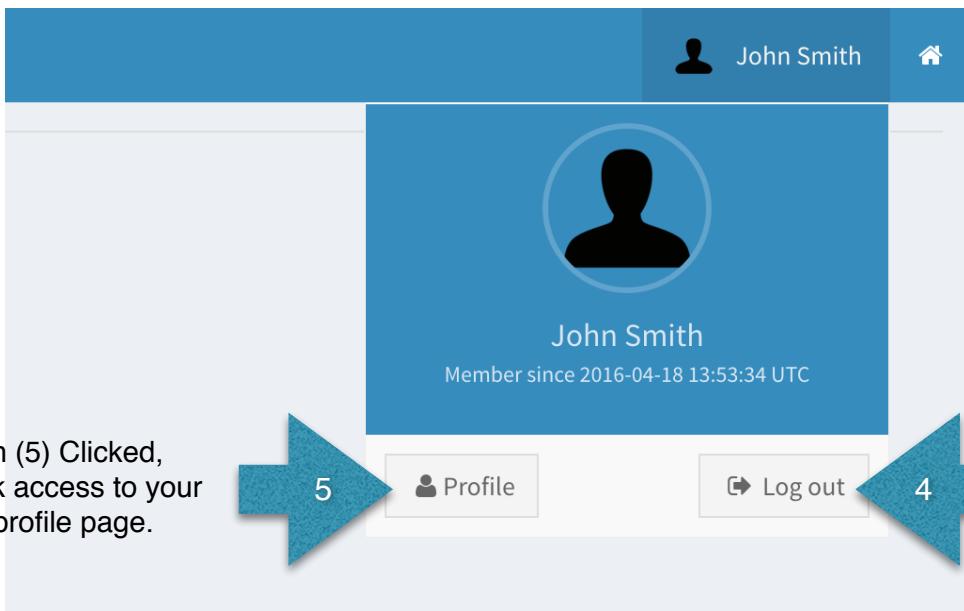
Attempting to access a page in which, was not created by you, or a page you are not permitted to view, will lead to a redirect to the dashboard page.

(1)

In this case `/groups/33` was attempted to access, however, that class was created by another user on the website.

Other Examples, may be trying to access another user's profile page.





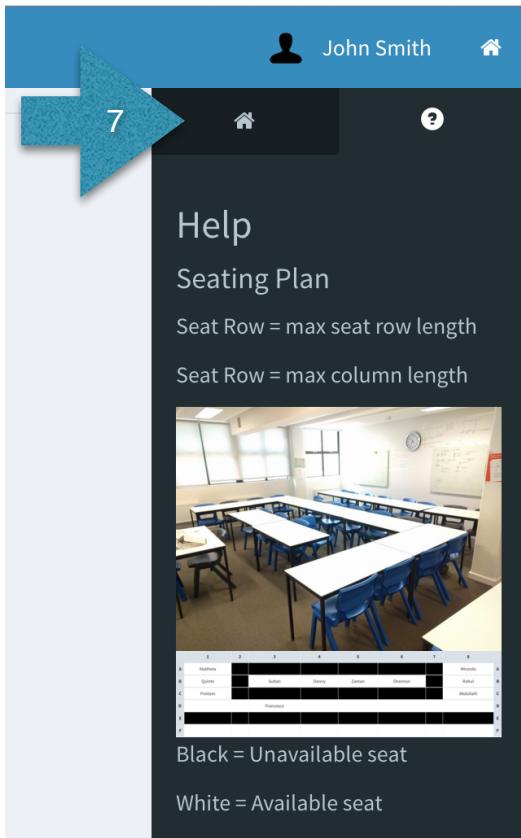
When (5) Clicked,  
Quick access to your  
own profile page.

5

Log out

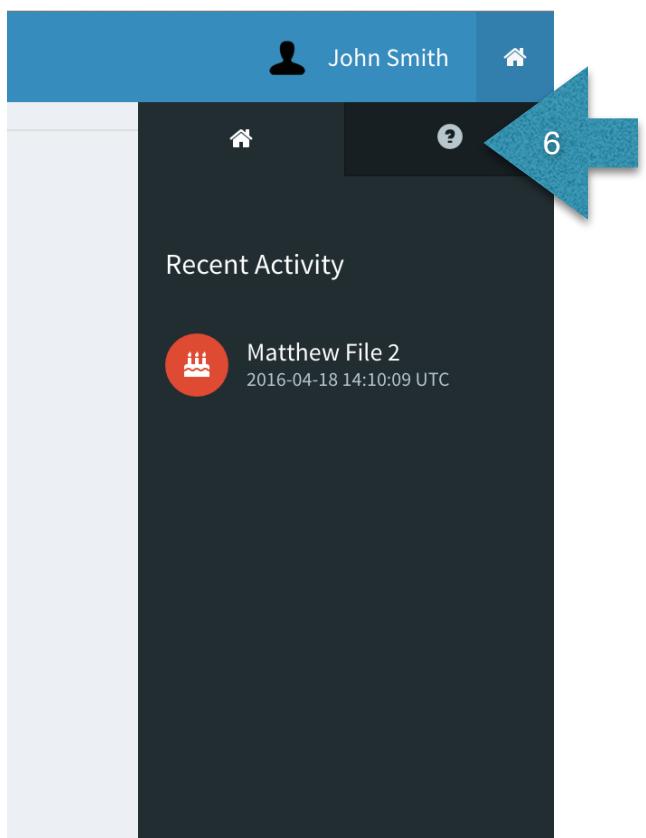
4

When (4) Clicked,  
Logs out. Redirects to  
the Landing page.



When (6) Clicked,  
Opens the Help tab part of the panel, with icon  
indicating it is for help.  
Features, an example of a seat layout created  
using the Seat Generator, located on the  
website. And a legend key, for the table.

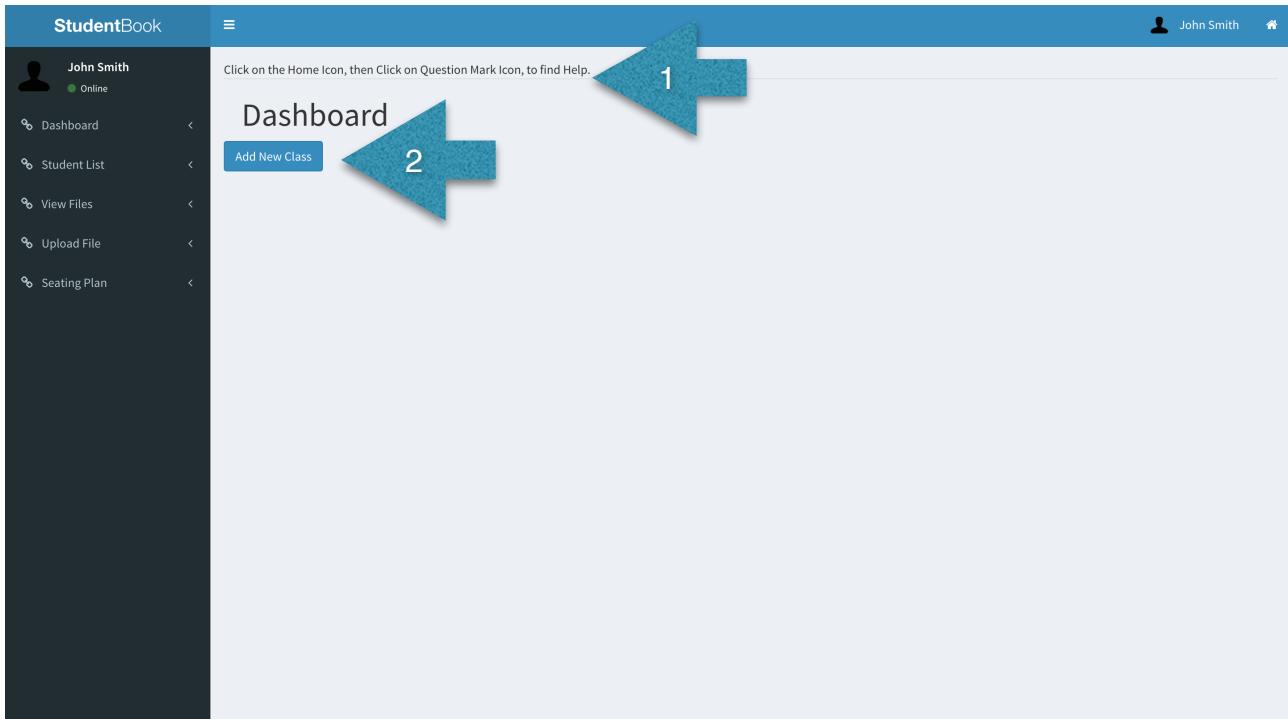
7



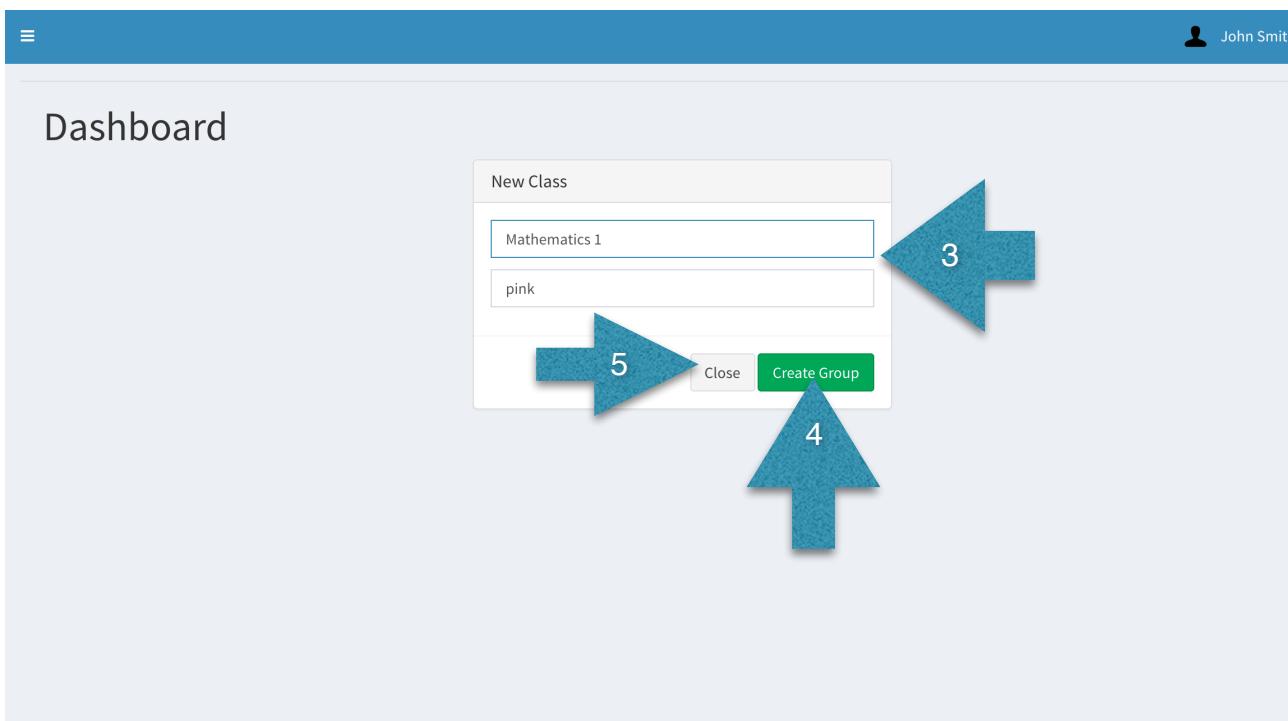
When (2) or (7) Clicked,  
Opens Side bar from the right of the page.  
Features, a recent activity tab, where it shows  
the most recently added file.

(2) can be Clicked again, when the  
sidebar panel is open , in order to close it.

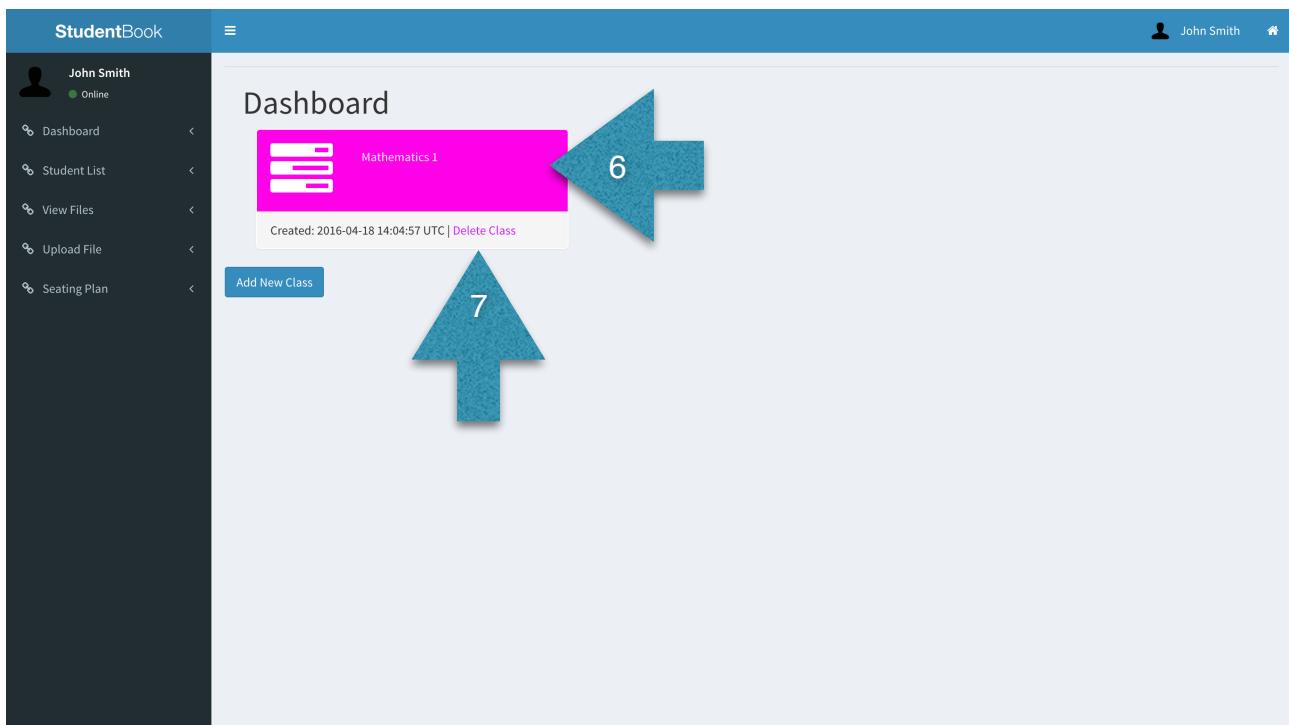
## 2.2.3: Dashboard



- (1) Upon first log in to the website, a notice will be added to the top of the page, informing you how to access, the help menu.
- (2) Clicking on the Add new class button will reveal a New Class form as shown in (3)

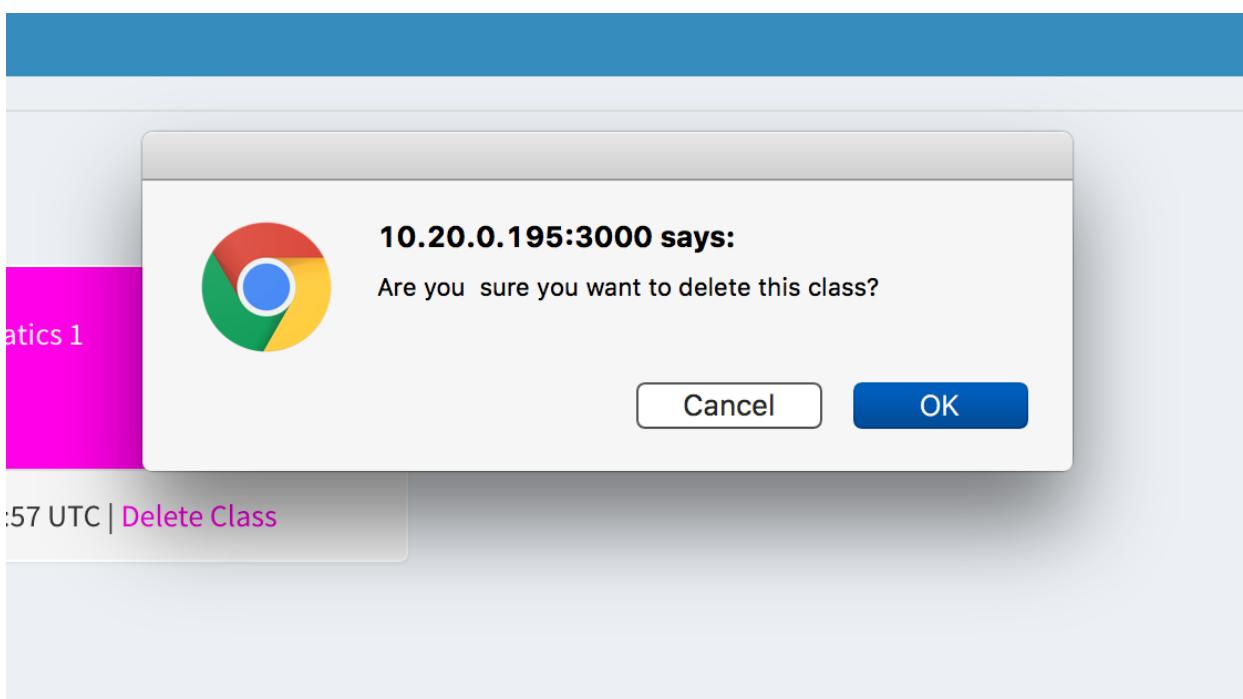


- (3) Revealed Form, that allows you to enter class name , and class color. Class Name is name, you will give for the new class. E.G Mathematics 1, and in the color field, you can able to enter a color for the class to be, e.g.. pink.
- (4) Create Group Submit button, when clicked on. Will process information from fields in order to generate a Class.
- (5) Clicking on Close button, will hide the form, and allow for the add new class button to reappear.



(7) Clicking on the Delete Class Button, allows you to delete the Class, in case you changed minds about it, or it is no longer relevant.

Once clicked, an alert menu will pop up, asking you to confirm your selection to delete the class. Press OK, if you still want to delete, or press cancel to no longer delete the class.



The dashboard shows the following class cards:

- Computing Class 1** (Pink card): Created: 2016-04-18 06:40:35 UTC | Delete Class
- dt** (Purple card): Created: 2016-04-18 11:34:26 UTC | Delete Class
- Computing Class 2** (Green card): Created: 2016-04-18 11:35:37 UTC | Delete Class
- Physics** (Red card): Created: 2016-04-19 20:08:16 UTC | Delete Class
- Further Maths** (Blue card): Created: 2016-04-19 20:08:32 UTC | Delete Class
- Chemistry 1** (Orange card): Created: 2016-04-19 20:08:44 UTC | Delete Class
- Dance** (Yellow card): Created: 2016-04-19 20:08:55 UTC | Delete Class

**Add New Class**

Here is an example of the range of colours available for you to pick from. In respective order. pink, purple, green, red, blue, yellow, orange.

The screenshot shows the 'Enter Student Details' dialog box open. The dialog contains the following fields:

- Name: Matthew
- Surname: Castrillon-Madrigal
- Grade: A

Buttons at the bottom of the dialog are 'Close' and 'Create Student'.

When (6) is clicked, you will be redirected to the Classes show page, where you can enter the current students of the class that you clicked on to enter this page.

(8) is a Student Details form, for you to enter student's details.

An example shown, where Students name is entered as Matthew, and surname as Castrillon-Madrigal, a predicted grade of an 'A'. However, you are able to enter any student's details.

(9) When clicked, submits the student's details, this must be done in order the save the student's details into the class.

(10) When clicked, closes similarly to that of the one located in the Dashboard.

The screenshot shows the StudentBook application interface. On the left is a dark sidebar with navigation links: Dashboard, Student List, View Files, Upload File, and Seating Plan. The main content area has a blue header bar with the title "Computing Class 2" and a user profile for "John Smith". Below the header is a section titled "Current Students:" containing fields for Firstname: Matthew, Surname: Castrillon-Madrigal, and Predicted Grade: A. At the bottom of this section is a timestamp: Created: 2016-04-18 14:06:11 UTC | Delete student. A modal window titled "Enter Student Details" is open in the center, containing input fields for Forename, Surname, and Enter Predicted Grade, along with a "Create Student" button.

This screenshot shows the same StudentBook interface as the first one, but with a different list of students. The "Current Students:" section now displays: Firstname: Junad, Surname: Ahmed, Predicted Grade: B; and Firstname: Matthew, Surname: Castrillon-Madrigal, Predicted Grade: A. A large blue arrow points to the "Delete student" link next to the second student's entry. The timestamp at the bottom of this section is: Created: 2016-04-18 16:42:06 UTC | Delete student. Below this is another student entry: Firstname: James, Surname: Smith, Predicted Grade: B, with a timestamp: Created: 2016-04-18 16:51:44 UTC | Delete student.

Student's are sorted in Alphabetical Order by surname. This is an example of when a few Students are added to the class.

(11) When clicked, allows for the Individual student of the class to be deleted. This might come hand, if you accidentally misspell a student's name, or a student is no longer enlisted in your class. Similarly, to that of the Class deletion, deleting a student will first prompt you with a confirmation alert.

## 2.2.3: All Students Page

StudentBook

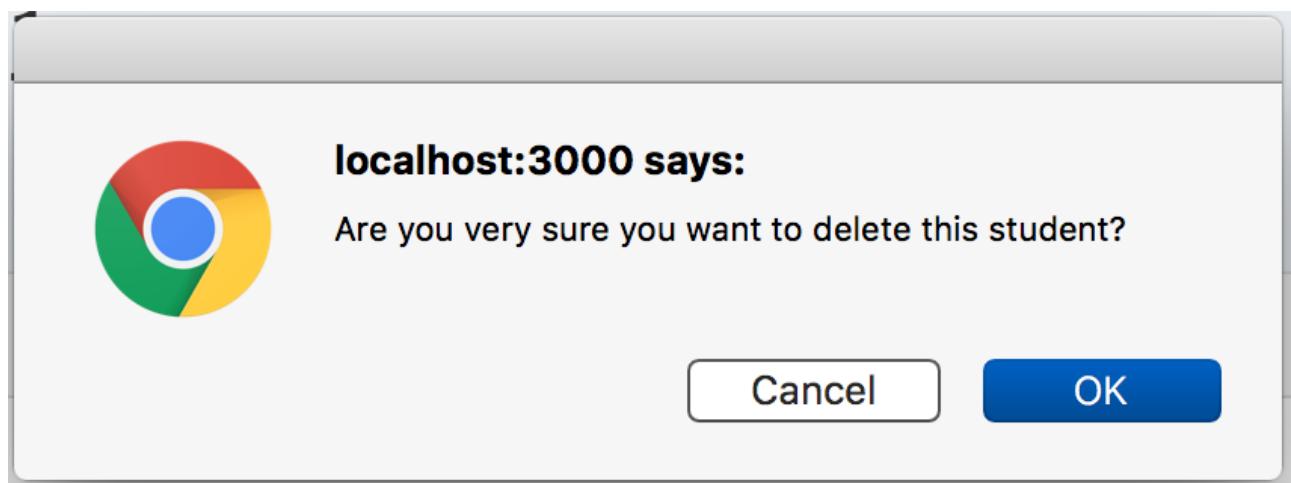
John Smith Online

- Dashboard
- Student List
- View Files
- Upload File
- Seating Plan

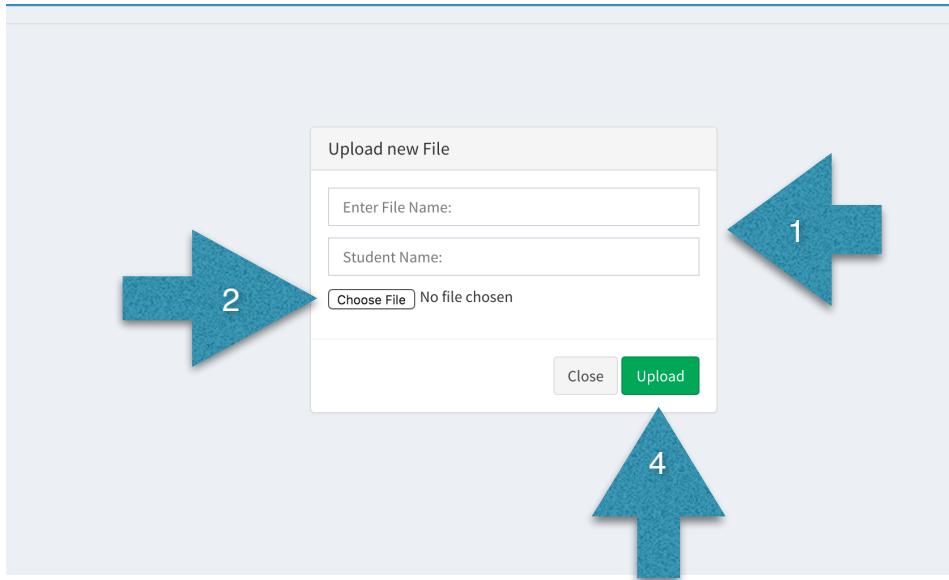
### All Students

Class: Computing Class 2
Firstname: Junad
Surname: Ahmed
Predicted Grade: B
Created: 2016-04-18 16:42:06 UTC   <a href="#">Delete student</a>
Class: Computing Class 2
Firstname: Matthew
Surname: Castrillon-Madrigal
Predicted Grade: A
Created: 2016-04-18 14:06:11 UTC   <a href="#">Delete student</a>
Class: Computing Class 2
Firstname: James
Surname: Smith
Predicted Grade: B

The all students page, gives you an overview of ALL students enlisted, in all of your classes. You can delete individual students from here by clicking (1) And Selecting OK to confirm your selection, however if you change your mind, you can always press cancel.



## 2.2.4: Upload Files Page

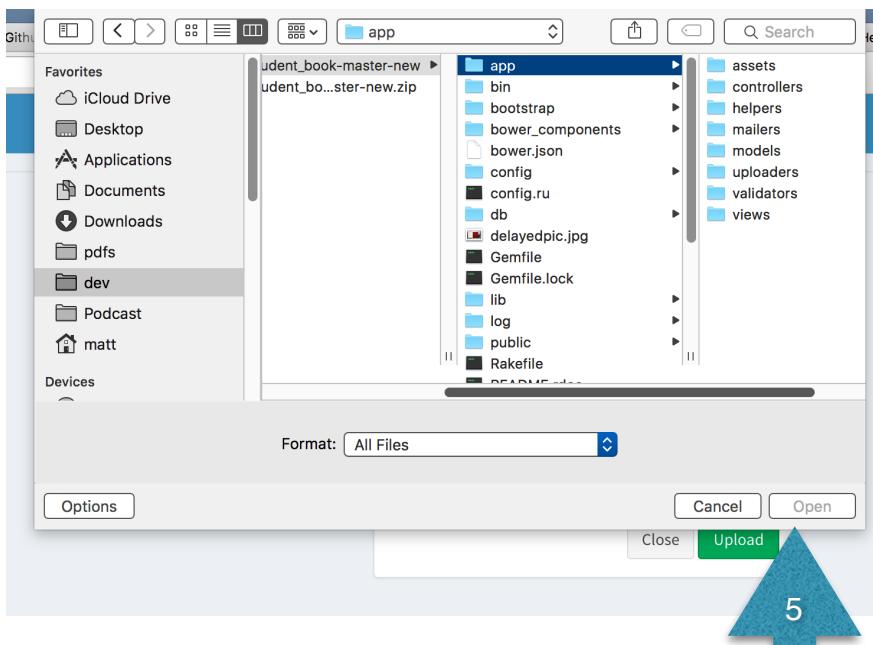


(1) is a Form for Uploading a new file. File name , would be the name which you find is related to the file. For example, A Student's coursework Doc. Could be named "Coursework Submission 1"

Student's Name field, would be the

Clicking (4) Will submit the user file data, and even save your file into the website.

Keep in mind, some larger files, may take longer to upload, depending on your device.



when (2) is clicked, you are able to select a file from your system, Any File type is ok.

Files can even be selected and uploaded from (most) mobile devices, such as android smart phones.

(5) After a file is selected, click Open, to be able to upload that file.  
However, pressing ENTER after you select a file, would do the same.

## 2.2.5 View Files

Task Name	Student Name	Download Link	Image Preview	Delete
Matthew File 1	Matthew Castrillon-Madrigal	<a href="#">Download File</a>		<a href="#">Delete</a>
Matthew File 2	Matthew Castrillon-Madrigal	<a href="#">Download File</a>		<a href="#">Delete</a>

(1) A table of current files you have added are presented, upon clicking on the View Files page. Information presented , is the Name of each file/task, the Student Name associated with the file, A download link for the file, and a delete button.

(2) A download link, that when clicked on, allows the user to download straight from the page, however if the media is an Image, or a PDF, It will allow you to view that file, straight from your chrome browser.

localhost:3000/uploads/user\_file/attachment/17/M1ans.pdf

## Chapter 2 Answers

(3) If the Media is an Image, the View Files table will allow you to preview the image from the table, straight from this page.

(4) Clicking on Delete, will allow you to delete the file, after deleting, the page should quickly refresh with a notice message, confirming that the file has been deleted.

(5) Clicking on Upload New File, will allow you to upload a file straight from this page, by making a form panel appear. Similar to that of the one in the Upload New File Page.

Task Name	Student Name	Download Link	Image Preview	Delete
First File	Matthew Castrillon-Madrigal	<a href="#">Download File</a>		<a href="#">Delete</a>
Matthew File 2	Matthew Castrillon-Madrigal	<a href="#">Download File</a>		<a href="#">Delete</a>

Upload new File

Enter file name:

Student Name:

No file chosen

[Close](#) [Upload](#)

## 2.2.6 Seating Plan Generator

The screenshot shows the 'Seating Plan' page from the StudentBook application. At the top right, it says 'John Smith'. On the left is a sidebar with 'StudentBook' at the top, followed by 'Dashboard', 'Student List', 'View Files', 'Upload File', and 'Seating Plan'. Below the sidebar is a large white area containing a form. The form has three input fields: 'Seat Row:', 'Seat Column:', and 'Available Seats:'. Below the fields is a blue 'Submit' button. To the left of the form, there are three large blue arrows: one pointing up labeled '1', one pointing down labeled '2', and one pointing left labeled '3'.

- (1) Upon entering the seating plan page, you will be encountered with a form, asking for details of Seat Row, Seat Column, and Available Seats. Seat Row would be the amount of rows down the layout of your class goes.



(2) Clicking the submit button, will allow for you to submit the data entered into the fields. However the page cannot process this information, if data entered is not valid.  
I.E, Available Seats must not be greater than the Seat Row X the Seat Columns, as even with a completely maxed out class layout, this is not physically possible.

(3) Features a back button, as well as a forward button, Acts the same as if you would press back, and forward button in your browser. You can use this to go back to a previously submitted form, to edit the details.

(1)  
Using the example of the Sixth Form Class to the left.

There are 6 rows of seats, Not Including the teacher's desk, And including the Gap in-between the 4th Row and the 6th Row. There are 8 columns of seats, from the furthest left side, and the furthest right side.

And there are 26 Available seats for students to sit.

The Class Row size is the Maximum, table width's you could fit from the first Students' Desk, to the last Students' Desk.

The Class Column Size is the Maximum table Width's you could fit from the Desk on the furthest left side, to the table on the furthest right side. Or Vice Versa.

Available Seats, are how many seats, are available for students to sit at.

The screenshot shows the StudentBook application's Seating Plan feature. On the left is a sidebar with navigation links: Dashboard, Student List, View Files, Upload File, and Seating Plan. The main area is titled "Seating Plan" and displays a grid. The grid has 10 rows labeled A through J and 10 columns labeled 1 through 10. All cells in the grid are black, indicating they are currently unavailable. Below the grid are three input fields: "Seat 1:", "Seat 2:", and "Seat 3:", each with a "Seat Position:" dropdown below it. A large blue arrow labeled "6" points to the right edge of the grid. A second blue arrow labeled "5" points to the bottom-right corner of the grid. A third blue arrow labeled "4" points to the top-right corner of the grid.

(4) Upon Clicking submit, from the previous form (3), the Grid matching your values entered will be presented. In this example, 10 Rows, and 10 columns, were submitted. Columns are numbered from 1-10, however will go up to whatever value you entered, as seat column size. And Rows are indexed from A-J, however can go up to as however much you entered. Entering 26 as row size, would go to A-Z, And entering 28 as row size will go from A-AB.

Grid elements are all coloured black indicating that there are no current available seats for students to sit in, as they have not yet been created.

(5) In addition to the grid, a form for you to enter the seat positions are shown. The number of forms to be filled, depends on the amount of seats that you have submitted from the previous form. In this case 3 available seats was the data entered.

A valid seat position, has to be in the format LETTER(s)NUMBER(s), for example G6 , or J10. Entering a value outside the grid range, for example, AA1, or B12, is not valid.

(6) Clicking Submit, will move you on to the next stage of the seating plan generator, if the values entered are all valid.

The screenshot shows the Seating Plan section of the StudentBook application. On the left is a sidebar with navigation links: Dashboard, Student List, View Files, Upload File, and Seating Plan. The main content area is titled "Seating Plan". It features a 10x10 grid of seats, labeled from A1 to J10. Seats A1, A3, and A5 are highlighted in white, while the rest are blacked out. A large blue arrow labeled "7" points to seat A5. Below the grid is a form with fields for "Class Name:" and "Number Of Students:", and a "Submit" button. A large blue arrow labeled "9" points to the "Submit" button. Another large blue arrow labeled "8" points to the "Number Of Students:" field.

(7) The Grid is now rebuilt, using the values of the seat locations to create a grid showing where the available seats are located. In this example the seats A1, A3, A5 were created.

(8) A new form is shown, where you will be able to enter the Class name, that the seating plan will be for. In addition, to the number of students enlisted in that class. Valid number of Students, would be where the number of students is less than , or equal to the number of seats available in the class, otherwise, where would the students sit?.

(9) Pressing Submit, will submit the data that you have entered into the form. Assuming that the data entered is valid.

The screenshot shows a user interface for a seating plan. On the left is a sidebar with a user profile for 'John Smith' (Online) and links for Dashboard, Student List, View Files, Upload File, and Seating Plan. The main area has a title 'Computing Seating Plan'. It features a large grid divided into 10 rows (A-J) and 10 columns (1-10). Below the grid are two form sections: 'Student 1:' and 'Student 2:', each with fields for Student Name, Student Gender m/f, Priority Learning level 1-3, and Student to not pair. A blue arrow labeled '10' points to the top of the grid. Two blue arrows labeled '11' and '12' point to the 'Submit' button at the bottom of the page.

(10) The Class Name Entered from the previous form, will change the title of the seating plan.

(11) Features a new form page, with student data to be filled in.

Student Name, would be the student's First name, however, if there are multiple students with the same first name you can use student's surname, or even a mix, as long as the student, as well as you are able to identify the student with the name typed.

Student Gender M/F, Enter m or M for Male, f or F for Female.

Student Learning Level, Enter the level at which the student may need to be further at the front of the class, if the student needs to be near the front, type 1, if the student does not need to be far at the front, type 3.

Student to not pair, Enter the Student's name that should not be paired with the student currently being filled in. If Student's name is entered in this field, then the program will do it's best to not put those 2 students together, however if are no more available seats for students. Then the program will have no choice but to place those students next to each other. However, this is rare to occur.

(12) Clicking on submit, will submit all the data entered, if entered correctly, and then move on to the next stage in the program.

The screenshot shows the 'Computing Seating Plan' page. A large blue arrow labeled '13' points to a red message at the top left that says 'Must Select Option'. Below the message is a seating grid for 10 rows (A-J) and 10 columns (1-10). Row A has seats 1, 3, 5, and 7 filled. Row B has seat 2 filled. Rows C through J are entirely empty. At the bottom, there are two radio buttons: 'Sort By Gender' (unchecked) and 'Do Not Sort' (checked), followed by a 'Submit' button.

(13) Shows an example of a notice that data entered is not valid, in this case the submit button was pressed(15) when there were was not a selection made.

(14) Features a radio button, which only allows for 1 value to be entered, clicking on Sort by gender, with select the sort by gender option, clicking on do not, will select the do not sort by gender option. And only 1 selection can be made at a time, since it is a radio button. Clicking on sort by gender, will allow for the data entered to be processed in a BOY-GIRL-BOY-GIRL-BOY-GIRL format. However, this option is completely optional.

(15) Clicking Submit will process this data and move onto the next form.

The screenshot shows the same 'Computing Seating Plan' page after a valid seating arrangement has been entered. A large blue arrow labeled '16' points to the 'Automatic Mode' radio button, which is now checked. The seating grid now shows a valid pattern: Row A has seats 1, 3, 5, and 7 filled. Row B has seats 2 and 4 filled. Rows C through J are entirely empty. The 'Submit' button is visible at the bottom.

- (16) Clicking on 16 with highlight the radio button, with the respective option that you have chosen. Clicking on Manual with highlight the left button, automatic with highlight the right button. Manual Mode allows you to add the students individually by yourself, however automatic, will use a program algorithm to auto fill the seating plan, taking into account multiple conditions.
- (17) Will Submit the data, from the Radio Selection, and then proceed to the next page.

Computing Seating Plan

Row	1	2	3	4	5	6	7	8	9	10	Row
A											A
B											B
C											C
D											D
E											E
F											F
G											G
H											H
I											I
J											J

1: enter student location for ishak      2: enter student location for oliver

Seat Position:      Seat Position:

Submit

- (18) If decided Manual Mode, from (16). The Page should show a form in order to enter each student Individually. The Notation for entering the positions are the same as entering the seat position previously. E.G A valid position would be A1, A3, or A

Computing Seating Plan

Row	1	2	3	4	5	6	7	8	9	10	Row
A	Ishak			Oliver							A
B											B
C											C
D											D
E											E
F											F
G											G
H											H
I											I
J											J

5.

- (20) Upon Clicking the Submit button (19), the Grid would update with all the positions for the students in the class, Using the data from the previous form on the previous page.(18)  
 Congratulations, you have now got a Completed Seating Plan.  
 If you want to Auto Generate the same seating plan, you can press the back button (3), and change the manual mode selection, to auto mode selection.

StudentBook

John Smith Online

Dashboard

Student List

View Files

Upload File

Seating Plan

Go Back Go Forward

Computing Seating Plan

Row	1	2	3	4	5	6	7	8	9	10	Row
A	Ishak				Oliver						A
B											B
C											C
D											D
E											E
F											F
G											G
H											H
I											I
J											J

(20) If decided Automatic, from (16) the Seating Plan, with automatically generate, and will show you the Final updated grid, No more forms are shown as the process has finished.

This is the Final seating plan, of your class room, and with the allocated students, to be used in your class.

If you want to Manually enter the positions for each student, you can press the back button (3), and change the Automatic mode selection, to Manual mode selection.

Either (20) or (21)

Now that you have a Completed Seating Plan, Either Auto Generated, or Manually Made, you can now be able to use this seating plan in your class.

Present this seating plan, at the start of class, and inform your students of the seating plan, and they should be able to work out where they are in the class, by looking at the grid, and searching for their name.

## 3: Troubleshooting

### Download and Installation Issues:

Issue: OS-X, Rails: “Failed to build gem native extension”

Solution: You must install Xcode Command line tools first. Continue to xCode Command Line tools Installation in Documentation.

Issue: Migrations are pending.

Solution: In Terminal type  
rake db:schema:load

Issue: An error occurred while installing pg (x.xx.x), and Bundler cannot continue

Solution: Install Homebrew and type  
brew install postgres.

### Running Server Issues:

Issue: Performance of Mac/PC slow when running application.

Solution: If the issue is really bad, either run the server on a more capable machine, or consider purchasing faster ram for your machine.

---

## xCode Command Line Tools Installation.

These instructions can be used for Mac OS X 10.9 Mavericks or 10.10 Yosemite. However, upgrading to OS X El Capitan, is still a good decision.

### Upgrade Mac OS X to 10.11

Mac OS X El Capitan was released on September 30, 2015. Make sure you have the latest version of Mac OS X. Under the Apple menu, check “About This Mac.” It should show “Version 10.11.1” or newer.

Search for the Mac OS X Terminal application by pressing the Command-Spacebar combination (which Apple calls "Spotlight Search") and searching for “Terminal.” Or look in the Applications/ Utilities/ folder for the Terminal application.

Check if you have previously installed the full Xcode package:

```
$ xcode-select -p
```

If you see:

xcode-select: error: unable to get active developer directory...

The Xcode package is not installed. Jump to the next section and install only the Xcode Command Line Tools.

If you see:

/Applications/Xcode.app/Contents/Developer

or /Library/Developer/CommandLineTools

The full Xcode package is already installed.

### Install Xcode Command Line Tools

Before installing Ruby, you'll need to prepare your computer by installing Apple's Xcode Command Line Tools.

```
$ xcode-select --install
```

Click “Install” to download and install Xcode Command Line Tools.

The instructions in the alert box may be confusing.

You don't need to “Get Xcode” from the App Store. Just click “Install” for the Xcode Command Line Tools.

Get Xcode, installs the full Xcode app, from the App Store, and may take quite a long time, depending on your internet connection.

After installing Xcode, restart terminal, and perform gem install rails, bundle install once again.

## 4: Glossary

**Cursor:**

A movable indicator on a computer screen identifying the point that will be affected by input from the user.

**Click:**

Typically when you left click, after placing cursor over an item.

**Panel:**

A block, containing information, or is different from other sections of the program.

**Form:**

Used so that data can be submitted and used by the website application. Uses the data from the fields of the form.

**Field:**

Part of a form, where data is entered into.

**Database:**

File structure in which data is saved to, typically in a rails application.

**Seat:**

A place where a student is able to sit, Each seat is typically half a desk length, or the size of the table width.

**Manual:**

Controlled or manipulated by a human operator (not automatically, such as by a computer)

**Automatic:**

A Process working by itself with little or no direct human control.

**File**

A collection of data or information that has a name, called the filename. Almost all information stored in a computer must be in a file.

**File Type**

Different type of files store different types of information, for example program files store programs, text files store text, etc.