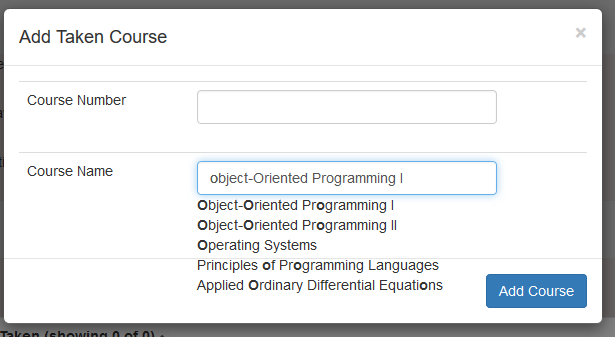
Open-Source Technology Used

**1 Twitter Typeahead**

We have implemented Twitter’s typeahead component to improve the user experience. It is a jQuery open-source text component that provides auto-completion suggestions as the user types (much like the google search bar). We use it to help the user add their needed and taken courses. The list of courses given in the auto-complete is retrieved from the server. This not only makes a better user experience, but makes it less likely for the user to input a course that doesn’t exist by accident.

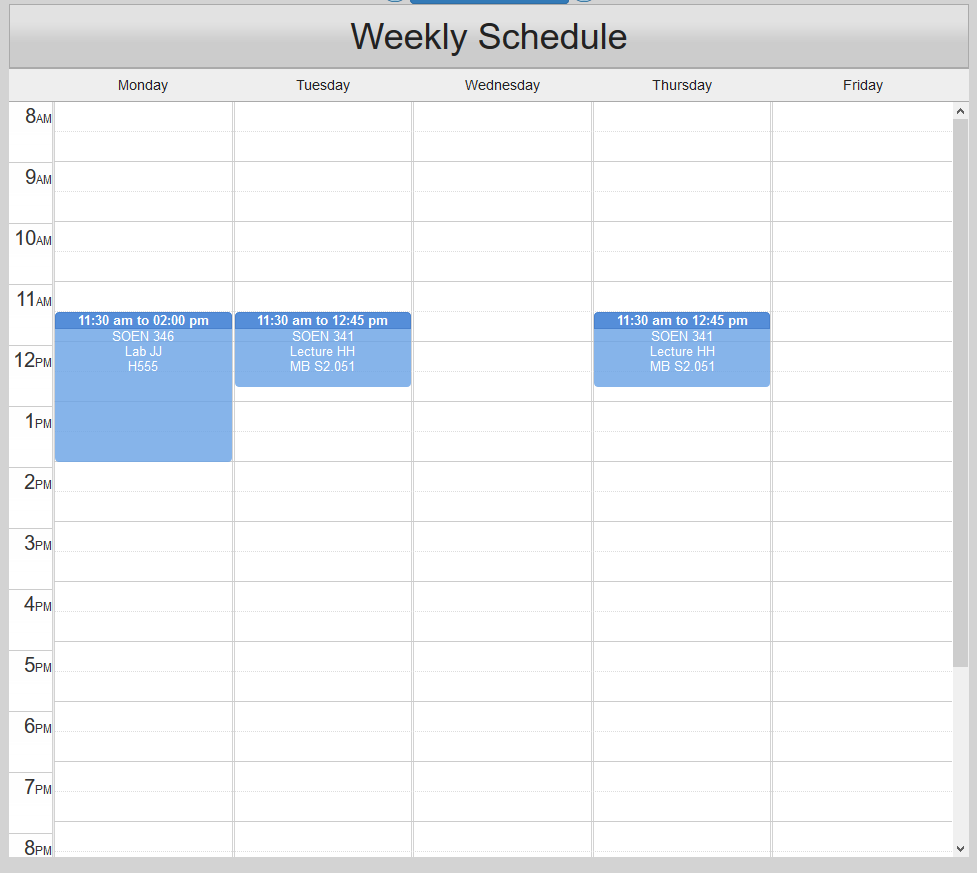


Twitter Typeahead

This component can be found at: <https://twitter.github.io/typeahead.js/>

**2 JQuery Week Calendar**

We use the jQuery Week Calendar to display the schedule once it has been generated. This majorly simplified this part of the UI. The component did need some customization for our uses, however, since originally it allows users to create, edit, and remove items on the schedule. It also displays items on a calendar with dates. None of these were things we wanted, so we had to customize the component to make it closer to what we wanted.



The Customized jQuery Week Calendar

This component can be found at: <http://www.jqueryrain.com/2012/04/jquery-week-calendar-with-demo/>

**3 React Bootstrap**

Bootstrap is a package of component that can be used for any web UI. It includes buttons, form objects, tables, dialogs, and many more commonly used UI components. They were made to have a list of components of the same style that behave the same across all major browsers. React Bootstrap is a package of the Bootstrap components that were made to work with React. They helped make our pages look consistent and professional as well as work across all major browsers without too much extra work.

React Bootstrap can be found at: <https://react-bootstrap.github.io/>

Encryption and Security

Bcrypt is used to encrypted passwords. It is a key derivation function that has a salt to protect from table attacks. As well bcrypt can be changed to a slower encryption which allows for stronger encryption. Bcrypt will be using crypt blowfish hashing that has a salt followed by two digit cost parameter and 22 characters from the alphabet.

In this encryption, username will be used as salt. This will prevent rainbow attack and will make it difficult to break it.

A salt is a random data that is used as an additional input to a hashed password. A new salt is randomly generated for each password.

The number 14 before the $ sign is the cost parameter, it decides how intensive the hashing is. According to php documentation, the range is from 04 to 31.

Server Calls

The following is the fully documented list of server calls used in the product. It lists the call, the type of call, the expected data to be sent with the call, and the expected format of the returned data.

These contracts were important to make sure the front and back end teams could work independently. Once these server calls were decided on, each team could work independently, programming their side to work according to these calls. The front-end team was able to fake these server calls using some hardcoded values until the back-end team had finished programming them. This allowed the front-end team to make everything work and then they just had to replace the server calls with the real calls when they were finished.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URL** | **Purpose** | **Type** | **Input** | **Returns** |
| /index | Login/Home Page | GET | None | The login/home page |
| /preferences | Preferences Page | GET | None | The preferences page |
| /schedule | Schedule Page | GET | None | The schedule page |
| /sequence | Course Sequence Page | GET | None | The course sequence page |
| /account | Account Page | GET | None | The account page |
| /admin | Admin Courses Page | GET | None | The admin courses page |
| /admincourse | Admin Edit Courses Page | GET | None | The admin edit courses page |
| /courses | Get all courses in DB | POST | None | [{"courseId":"1","courseCode":"COMP 248","semester":"Fall","description":"Introduction to programming. Basic data types, variables, expressions, assignments, control flow. Classes, objects, methods.","name":"Object-Oriented Programming I","credits":"3}]  //goes on forever |
| /sections | Get all sections for all courses in DB | POST | None | ^ Similar output |
| /sectioncourse | Get all section in DB for the specified course | POST | $courseId=$\_POST('courseId'); | ^ Similar output |
| /userprefs | Get preferences for a user | POST | $username=$\_POST['username']; | {"success":"true","username":"SprinkKing","courseload":"4","dayoff":"Monday","preferredTime":"Afternoons"} |
| /needed | Get needed courses for a user | POST | $username=$\_POST['username']; | String that will be parsed  {"List":[{"name":"oop", "number":"248"},{"name":"oop2", "number":"249"}]} |
| /taken | Get taken courses for a user | POST | $username=$\_POST['username']; | String that will be parsed  {"List":[{"name":"oop", "number":"248"},{"name":"oop2", "number":"249"}]} |
| /scheduler | Generate schedule | POST | $username=$\_POST['username']; |  |
| /addadmincourse | Add a course to the DB | POST | $json=$\_POST['json'];  //$json='{"section":"UI-X","sectionId":"1","classroom":"H-4","semester":"Winter","type":"Lab","dayOffered":"5","beginTime":"13:35:00","endTime":"14:35:00","courseId":"2","courseCode":"COMP 248","sectionNum":"1"}'; | False: if course with that id already exists  True: otherwise  {"success":"true","courseID":"142","Course":"{\"courseId\":\"null\",\"courseCode\":\"COMP 2489895\",\"semester\":\"Summer\",\"description\":\"Salt level critical, abort !\",\"name\":\"Object-Oriented Programming I\",\"credits\":\"3\"}"} |
| /addadminsection | Add a section to the DB | POST | $json=$\_POST['json'];  //$json='{"section":"UI-X","sectionId":"1","classroom":"H-4","semester":"Winter","type":"Lab","dayOffered":"5","beginTime":"13:35:00","endTime":"14:35:00","courseId":"2","courseCode":"COMP 248","sectionNum":"1"}'; | {"success":"true","SectionID":"352","Section":"{\"section\":\"UI-GR\",\"sectionId\":\"null\",\"classroom\":\"H-4\",\"semester\":\"Winter\",\"type\":\"Lab\",\"dayOffered\":\"5\",\"beginTime\":\"13:35:00\",\"endTime\":\"14:35:00\",\"courseId\":\"2\",\"courseCode\":\"COMP 248\",\"sectionNum\":\"1\"}"} |
| /removeadmincourse | Remove a course from the DB | POST | //$courseId=$\_POST['courseId']; //course id | N/A |
| /removeadminsection | Remove a section from the DB | POST | //$sectionId=$\_POST['sectionId']; //section id | N/A |
| /editcourse | Edit course information | POST | $json=$\_POST['json'];  //$json='{"courseId":"1","courseCode":"COMP 248","semester":"Fall","description":" Critical amount of salt atteined","name":"Object-Oriented Programming I","credits":"3"}'; | {"success":"true","Course":"{\"courseId\":\"1\",\"courseCode\":\"COMP 248\",\"semester\":\"Fall\",\"description\":\"Introduction to programming. Critical amount of salt atteined\",\"name\":\"Object-Oriented Programming I\",\"credits\":\"3\"}"} |
| /editsection | Edit section information | POST | $json=$\_POST['json'];  //$json='{"section":"UI-X","sectionId":"1","classroom":"H-905","semester":"Winter","type":"Lab","dayOffered":"5","beginTime":"13:35:00","endTime":"14:35:00","courseId":"2","courseCode":"COMP 248","sectionNum":"1"}'; | {"success":"true","Section":"{\"section\":\"UI-GR\",\"sectionId\":\"1\",\"classroom\":\"H-420\",\"semester\":\"Winter\",\"type\":\"Lab\",\"dayOffered\":\"5\",\"beginTime\":\"13:35:00\",\"endTime\":\"14:35:00\",\"courseId\":\"2\",\"courseCode\":\"COMP 248\",\"sectionNum\":\"1\"}"} |
| /editpreferences | Edit user’s preferences | POST | $username=$\_POST['username'];//username  $cload=$\_POST['cload'];//courseload  $dayoff=$\_POST['dayoff'];//time off  $preftime=$\_POST['preftime'];//pref time | {"success":"true","username":"JasonB","courseload":"5","dayoff":"Monday","preftime":"Mornings"} |
| /editneededcourses | Edit user’s needed courses | POST | $old=$\_POST['username'];  $json=$\_POST['json'];  // $json='{"List":[{"name":"oop", "number":"248"},{"name":"oop2", "number":"249"}]}'; | {"success":"true","username":"user17","List":"{\"List\":[{\"name\":\"Operating Systems\",\"number\":\"COMP 346\"},{\"name\":\"Principles of Electrical Engineering\",\"number\":\"ELEC 275\"},{\"name\":\"Probability and Statistics in Engineering\",\"number\":\"ENG….. }]}"} |
| /edittakencourses | Edit user’s taken courses | POST | {username:’’, json: '{"List":[{"name":"oop", "number":"248"},{"name":"oop2", "number":"249"}]}'} | {"success":"true","username":"user17","List":"{\"List\":[{\"name\":\"Mathematics for Computer Science\",\"number\":\"COMP 232\"},{\"name\":\"Object-Oriented Programming I\",\"number\":\"COMP 248\"},…….{name\":\"Sustainable Development and Environmental Stewardship\",\"number\":\"ENGR 202\"}]}"} |
| /editusername | Change username for user | POST | $old=$\_POST['old']; // old username  $new=$\_POST['new'];  //new username | False: new username exists already  True: otherwise  result"=>"true","username"=>"$new") |
| /editemail | Change e-mail for user | POST | $old=$\_POST['old'];//username  $new=$\_POST['new'];//new email | ("success"=>"true","username"=>"$old","email"=>"$new")  False: username not found |
| /editpassword | Change password for user | POST | $username=$\_POST['old']; //username  $newPassword=$\_POST['new']; //new password | "success"=>"true","password"=>"$new")); |
| /email | Get user’s email | POST | $username=$\_POST['username'];//username | E-mail as a String  "result":"good","username":"JasonB","email":"newEnail@email.com"} |
| /login | Login verification | POST | $username=$\_POST['username']; $password=$\_POST['password']; | {success:bool, isAdmin:bool}  {"success":"true","username":"JASONB","isAdmin":"true"} |
| /register | Register user | POST | $username=$\_POST['username'];  $password=$\_POST['password'];  $email=$\_POST['email']; | False: username exists already  True: otherwise  {"success":"false","username":"SprinkKing","error":"usernametakenalready"} |