BCS Degree Navigator

A project by Danny, Cody & Kathryn

Why did we create this?

BCS Program Requirements Worksheet **Name of Student:** ______________________________

Contents Updated on These Dates (Advisor: Provide the Date and Your Initials):

1.			3.			5.	
2.			4.			6.	

21 courses are required for the BCS degree (usually 63-70 credits, depending on the number of 4-credit courses). The maximum number of credits allowed is 90. In the "Completed?" column, use a check-mark for completed courses, or write "CIP" (course in progress), "P1" (planned for term 1), "P2", etc. For electives and exemption replacements, a "course" means 3 credits. So, a 6-credit course like PSYC 304 counts as 2 courses while a 1-credit course like CPSC 189 only counts if grouped with 2 more credits.

Course	**Completed?**	**Exempt?**	**If Exempt, Replaced with what?**	**Comments or Explanations (e.g., Reason for Exemption)**
ENGL 1xx				Approved ENGL course required prior to admission.
ENGL 1xx				Another first-year ENGL course. ENGL 140 does not fulfill this requirement
CPSC 110				Or 85% or more in CPSC 111, or a pass for both CPSC 111 and 211
CPSC 121				
MATH 180				Or MATH 184, 100, 102, …
STAT 203				Or STAT 200 or STAT 241
Communication Requirement				ENGL 301, SCIE 300, or approved alternative. If exempted, replacement must be upper-level (300+)
CPSC 210				Or CPSC 211
CPSC 213				
CPSC 221				

Goals Achieved

Which goals were achieved?

- **Minimal Goals**
  - Accounts
  - Enter and store courses
  - Display completed requirements
  - Display remaining requirements
- **Standard Goals**
  - Access the app on any device
  - Calculate GPA
- **Stretch Goals**
  - Include session planning

Goals Changed

How did goals change over time?

- **Standard Goals**
  - Search
  - Validation
  - Suggestions

Demo

Why would a student use the BCS Degree Navigator?

User #1

Almost finished!

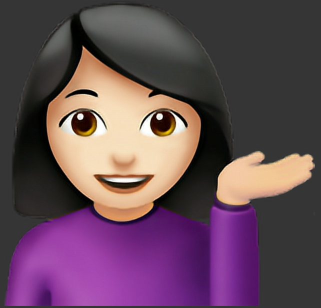
A long-time user who is almost finished the BCS degree wants to plan his or her remaining semesters. Then the user wants to email Steve to confirm that his or her plan will satisfy the requirements

Todo

Double check that I've completed the courses that I'm supposed to.

Thankfully, it looks like I'm nearly done! Phew!

User #2 New student!

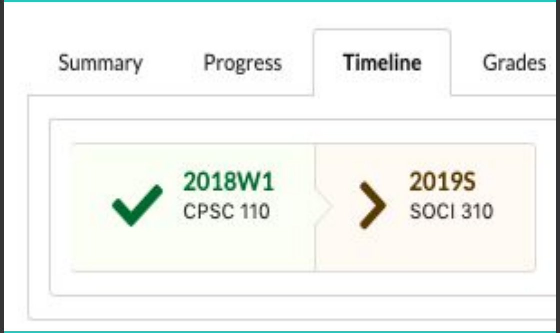


A new user wants to create an account and visualize how much of the BCS degree they have completed to date.

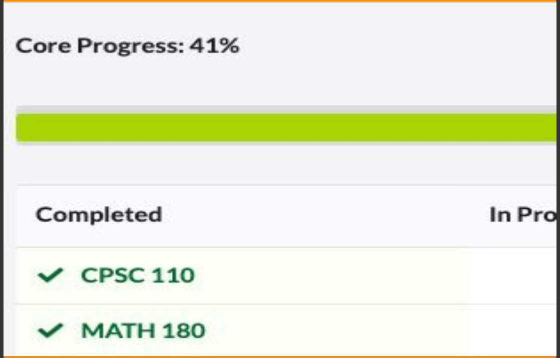
To do

How long is this degree gonna take me, anyways?!

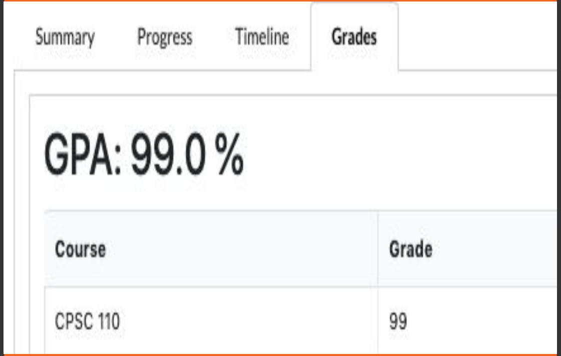
Super cool stuff here 😎



✓ 2018W1 CPSC 110	2019S SOCI 310


Core Progress: 41%

Completed	In Pro
✓ CPSC 110	
✓ MATH 180	



GPA: 99.0 %

Course	Grade
CPSC 110	99

Some bumps in the road...

Working with Semantic UI has some limitations.

- The table component is designed to handle records in rows
- Our progress table organizes courses into columns
- We had to write functions to restructure our data so that it could be mapped row by row

Some bumps in the road...

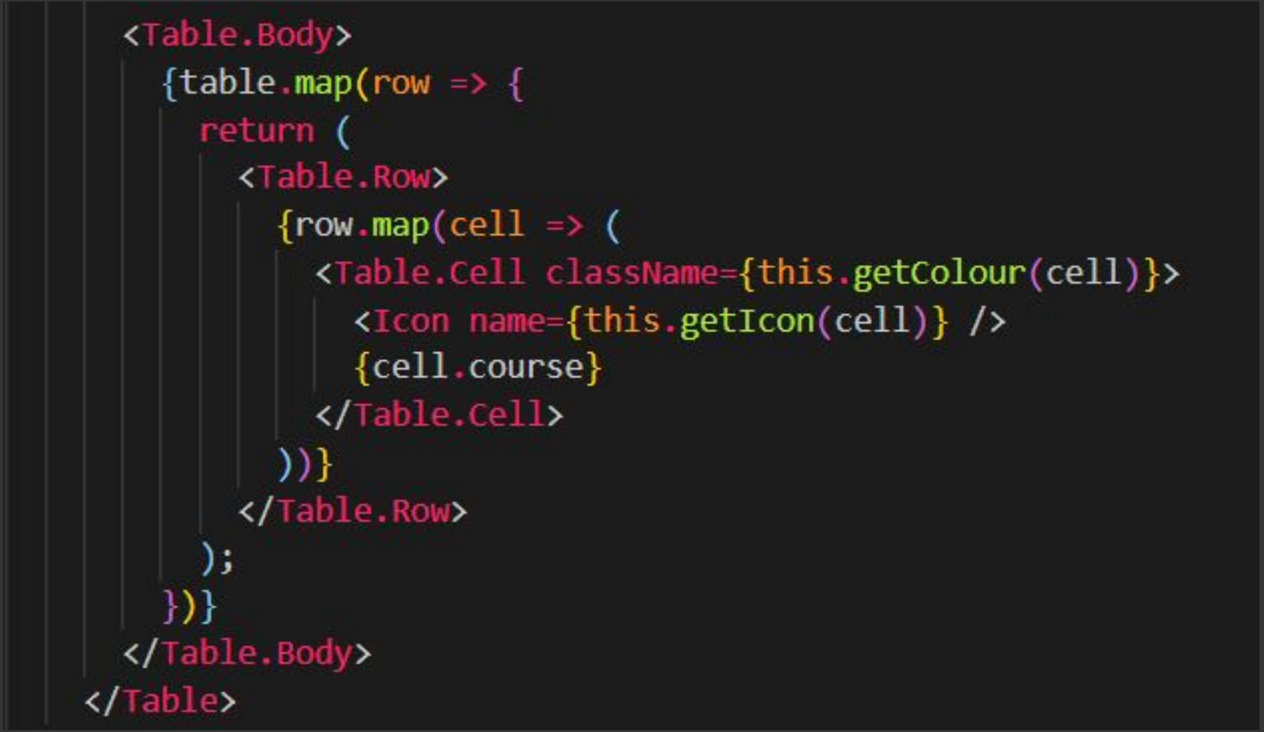
Designing degree requirement verification:

- First attempt: two arrays, $O(n^2)$
- Third attempt: one array + one JSON Object, $O(n)$
- Recompute from fresh slate on render

The 5 Units

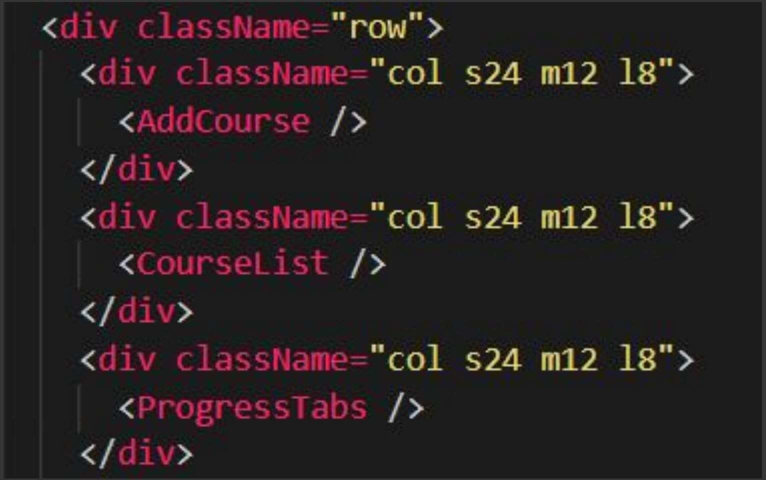
1) HTML, CSS, JS
2) React, Meteor, and the “Front End”
3) NodeJS, Meteor, and other framework “Back Ends”
4) NoSQL and MongoDB
5) Release Engineering

Code 1) HTML, CSS, Semantic UI, JS

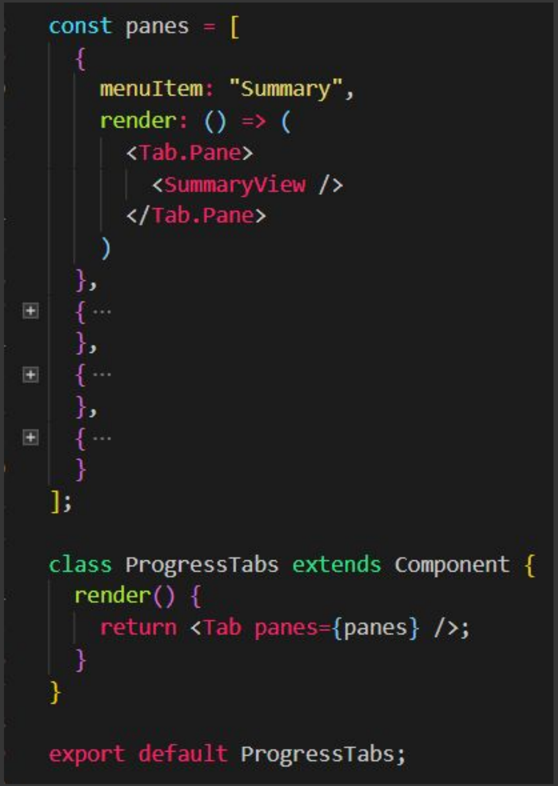


```
      <Table.Body>
        {table.map(row => {
          return (
            <Table.Row>
              {row.map(cell => (
                <Table.Cell className={this.getColour(cell)}>
                  <Icon name={this.getIcon(cell)} />
                  {cell.course}
                </Table.Cell>
              ))}
            </Table.Row>
          );
        })}
      </Table.Body>
    </Table>
```

Code 2) React, Meteor, & “Front End”

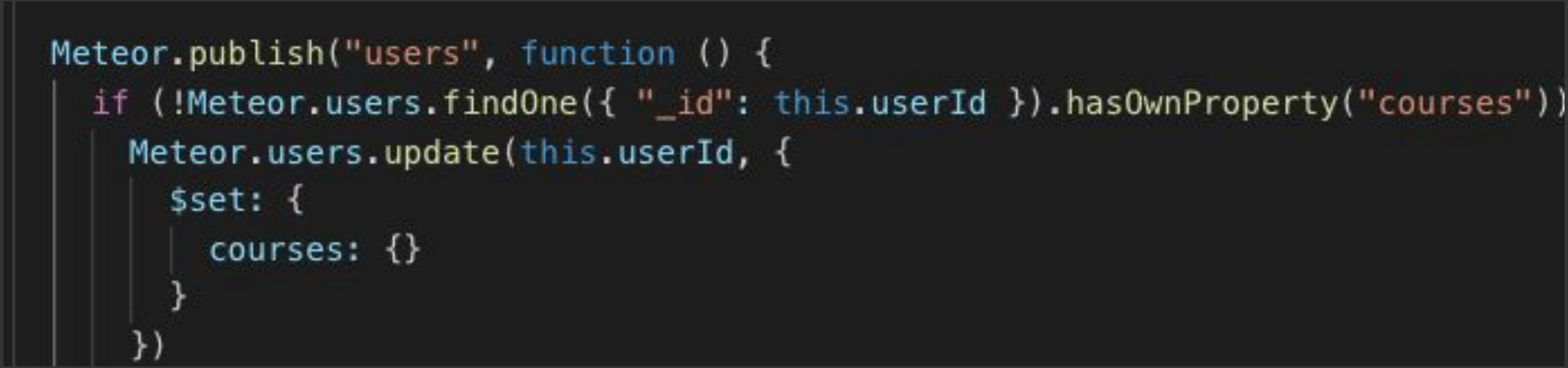


```
<div className="row">
  <div className="col s24 m12 l8">
    <AddCourse />
  </div>
  <div className="col s24 m12 l8">
    <CourseList />
  </div>
  <div className="col s24 m12 l8">
    <ProgressTabs />
  </div>
```



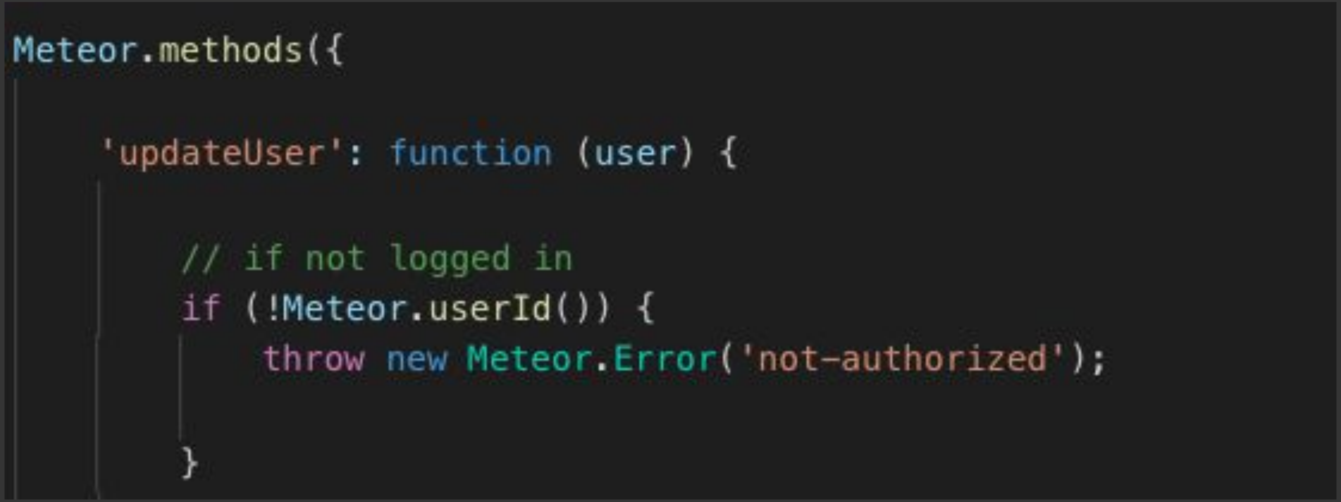
```
const panes = [
  {
    menuItem: "Summary",
    render: () => (
      <Tab.Pane>
        <SummaryView />
      </Tab.Pane>
    )
  },
  {…
  },
  {…
  },
  {…
  }
];

class ProgressTabs extends Component {
  render() {
    return <Tab panes={panes} />;
  }
}

export default ProgressTabs;
```

Code 3) NodeJS, Meteor, & “Back End”



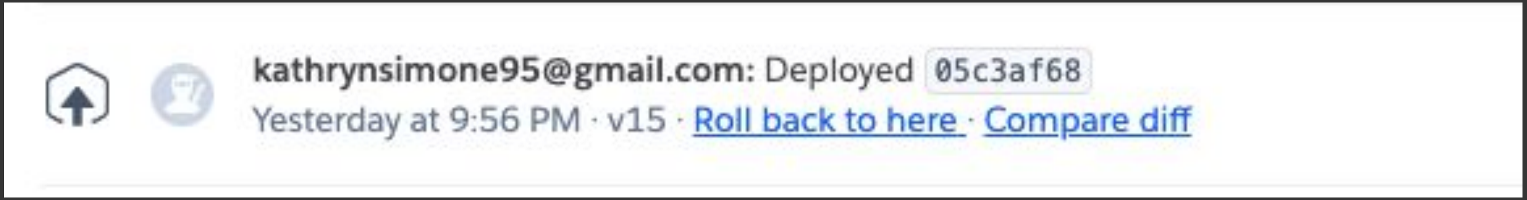
```
Meteor.publish("users", function () {
  if (!Meteor.users.findOne({ "_id": this.userId }).hasOwnProperty("courses"))
    Meteor.users.update(this.userId, {
      $set: {
        courses: {}
      }
    })
```

Code 4) NoSQL and MongoDB



```
Meteor.methods({

    'updateUser': function (user) {

        // if not logged in
        if (!Meteor.userId()) {
            throw new Meteor.Error('not-authorized');

        }
```

Code 5) Release Engineering


kathrynsimone95@gmail.com: Deployed `05c3af68`
Yesterday at 9:56 PM · v15 · Roll back to here · Compare diff

Thank you!

Questions are welcomed.