IT/CS Workshop (ICW) Documentation

Shane Geller

December 5, 2019

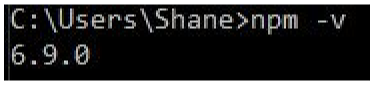
1. General Information

ICW is a Node.js web application with a React/Redux frontend and a MongoDB database. Its code base is split into two GitHub repositories icwNode (https://github.com/SMGeller/icwNode) and icwReact (https://github.com/SMGeller/icwReact). The icwNode repository contains the code for the backend Node.js application, and the icwReact repository contains the code for the React/Redux frontend. More detailed explanations about the code setups are located in the READMEs of each repository.

2. Local Installation Instructions

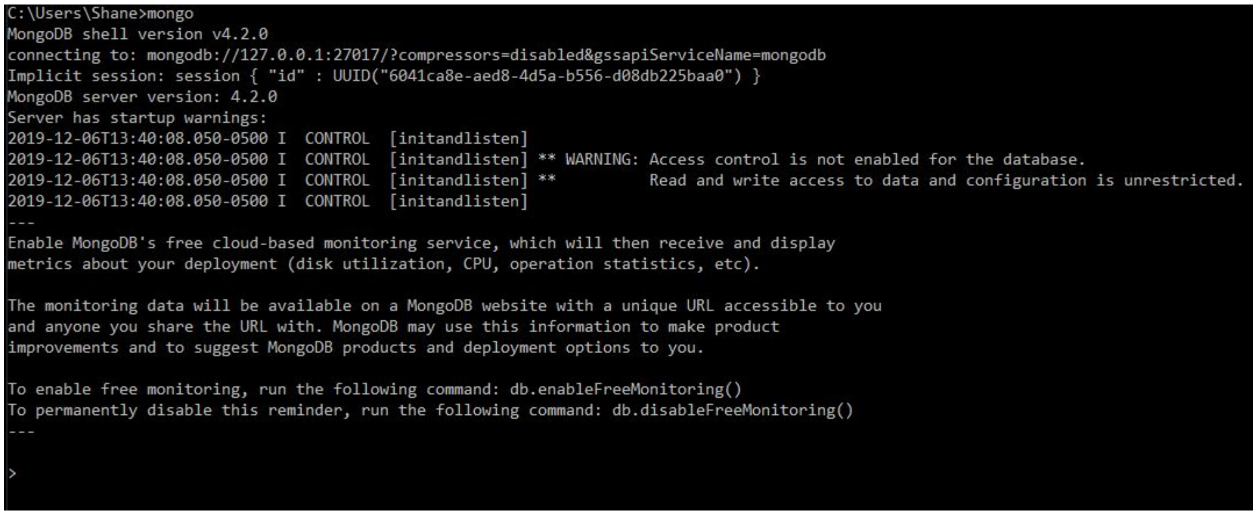
These instructions are for installing the application locally on a Windows 10 computer. Linux and macOS instructions will be similar but not exact.

First clone the code from the icwNode and icwReact repositories or copy it from the DVD that Dr. Schmalz has. Then download and install Node.js (https://nodejs.org/en/download/). Once the installation is complete type "*npm -v*" into the command console and you should see the version appear as shown below. If it doesn't, then check to make sure the Node.js directory is in the Path of your environment variables.



```
C:\Users\Shane>npm -v
6.9.0
```

Once Node is successfully installed, download and install MongoDB Community Server (https://www.mongodb.com/download-center/community). You don't have to install it as a service since the Node application will start the MongoDB daemon when it starts. Once it is installed make sure you can run the MongoDB Shell on the command line by typing "*mongo*". If you can't, then add the MongoDB bin directory (\Program Files\MongoDB\Server\4.2\bin) to your environment variables.

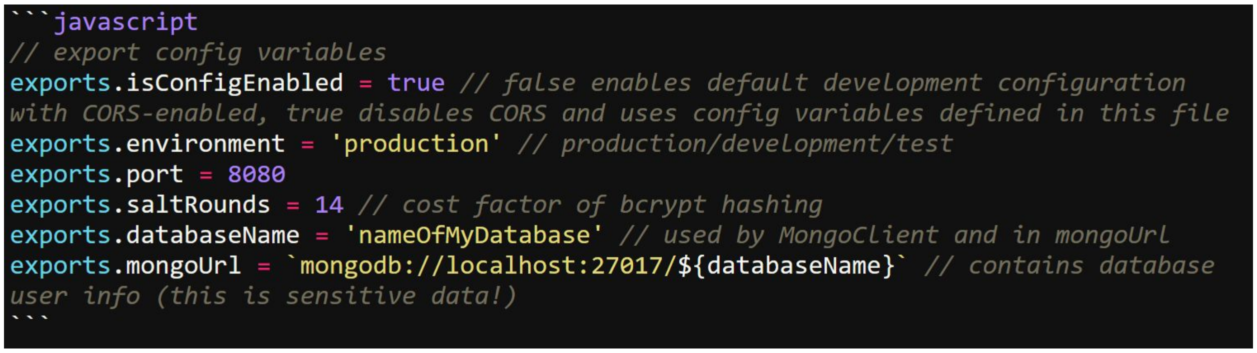


```
C:\Users\Shane>mongo
MongoDB shell version v4.2.0
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("6041ca8e-aed8-4d5a-b556-d08db225baa0") }
MongoDB server version: 4.2.0
Server has startup warnings:
2019-12-06T13:40:08.050-0500 I  CONTROL  [initandlisten]
2019-12-06T13:40:08.050-0500 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-12-06T13:40:08.050-0500 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-12-06T13:40:08.050-0500 I  CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

>
```

Type “exit” to quit the MongoDB Shell.

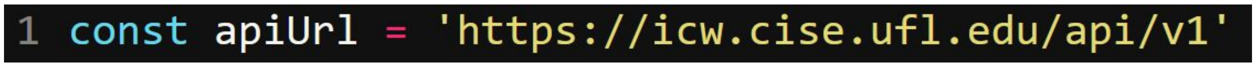
Install the npm-check module (https://www.npmjs.com/package/npm-check) using “*npm install -g npm-check*” which installs the module globally onto your machine so it can be used by any Node application. This is a tool to check and update all the Node modules located in the package.json files of each repository. Npm-check-updates (https://www.npmjs.com/package/npm-check-updates) is another module that does the same thing but isn't as feature rich. Update the module versions for both icwNode and icwReact.

Create a *.config.js* file, like the one shown below, that points to your local MongoDB installation and place it in the root directory of the icwNode application.

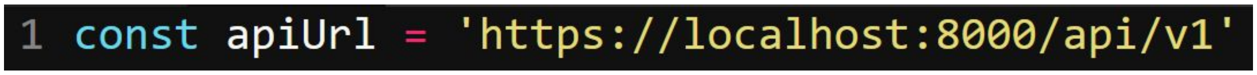


~~~
```javascript
// export config variables
exports.isConfigEnabled = true // false enables default development configuration
with CORS-enabled, true disables CORS and uses config variables defined in this file
exports.environment = 'production' // production/development/test
exports.port = 8080
exports.saltRounds = 14 // cost factor of bcrypt hashing
exports.databaseName = 'nameOfMyDatabase' // used by MongoClient and in mongoUrl
exports.mongoUrl = `mongodb://localhost:27017/${databaseName}` // contains database
user info (this is sensitive data!)
```
~~~

Change the apiUrl variable of the actions.js file located in the /src/ folder of the icwReact repository. It is currently *https://icw.cise.ufl.edu/api/v1* which is for the production environment and needs to be changed to *https://localhost:<port>/api/v1* for the development environment. The default port is 8000, but can be changed in the *.config.js* file.



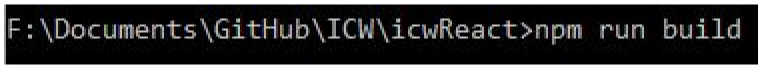
```
const apiUrl = 'https://icw.cise.ufl.edu/api/v1'
```



```
const apiUrl = 'https://localhost:8000/api/v1'
```

Install create-react-app (https://create-react-app.dev/) using npm "*npm install -g create-react-app*". If it is already installed then uninstall and reinstall it to get the latest version. This module builds the React/Redux code and minifies it into static files to be served by the Node application.

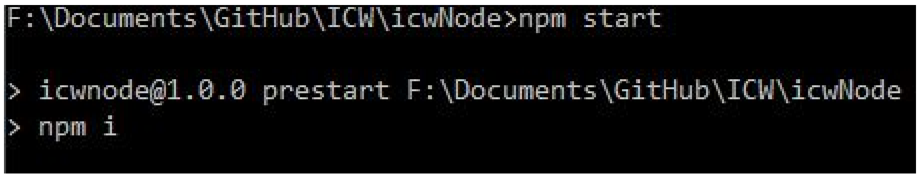
Open the command line and navigate to the root directory of the icwReact application and run the following command to build the static files.



```
F:\Documents\GitHub\ICW\icwReact>npm run build
```

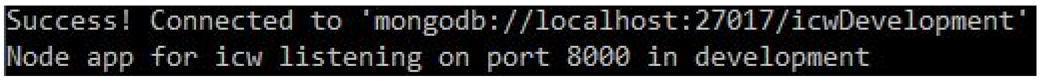
Once the command is complete, copy the newly created /build/ folder located in the root icwReact directory into the root directory of the icwNode application.

Navigate to the icwNode directory on the command line and run "*npm start*" which will install all the Node modules that ICW needs and start a locally running version of the application server.

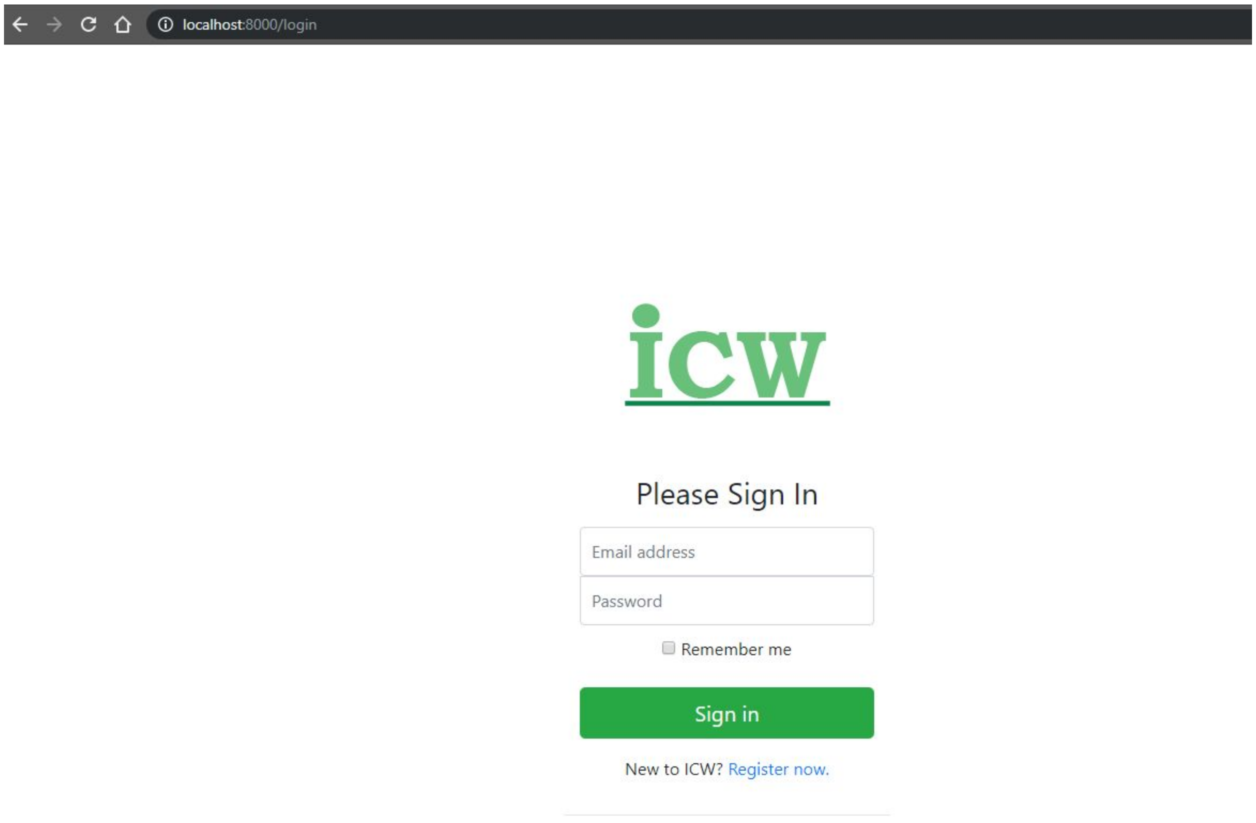


```
F:\Documents\GitHub\ICW\icwNode>npm start

> icwnode@1.0.0 prestart F:\Documents\GitHub\ICW\icwNode
> npm i
```

When you see the success message, try to connect to application at *localhost:<port>*.



```
Success! Connected to 'mongodb://localhost:27017/icwDevelopment'
Node app for icw listening on port 8000 in development
```

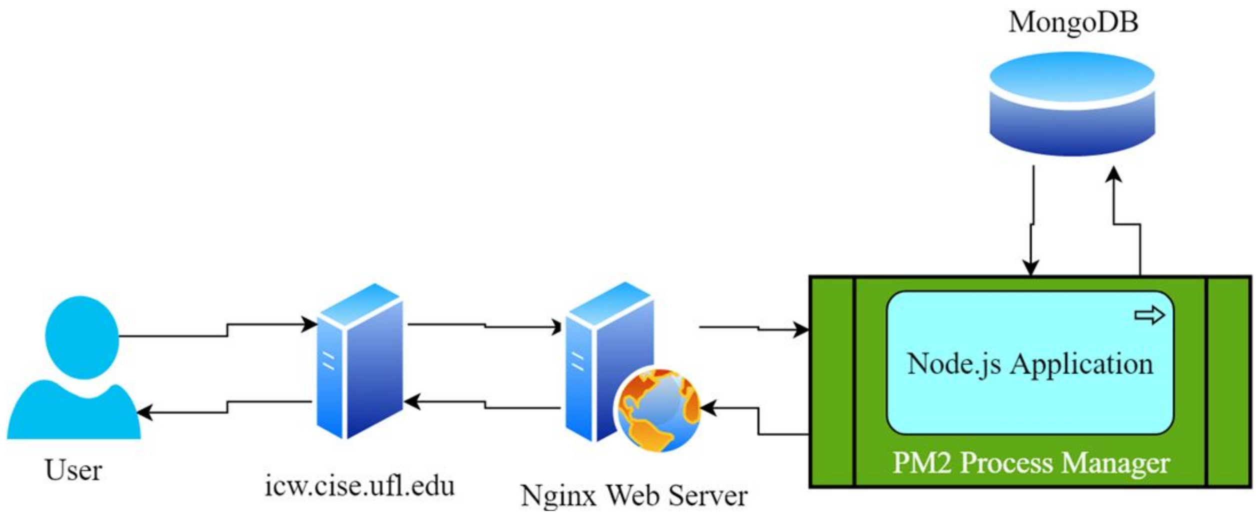


The login screen should appear, and ,if you are able to register a new account, then the installation was successful. If you can't register, then check the *action.js* file as mentioned above to make sure the apiUrl variable is correct. You can also open your browser's console to see if it is using the correct Url. Sometimes your browser is still using a cached version of the static files so try refreshing a few times if you are sure the apiUrl is correct.

Cache full when running npm install?

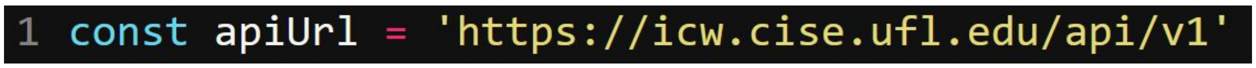
1. Delete npm cache "*npm cache clean --force*"
2. Delete node_modules folder "*del /f node_modules*"
3. Run npm-check
4. Start Node app with "*npm start"*

3. Server Setup and Deployment Instructions



The icw.cise.ufl.edu server currently runs Ubuntu 18.04 LTS. Much of the configuration was setup using DigitalOcean documentation regarding node/express/mongo configuration on Ubuntu:
https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-18-04.
[PM2](http://pm2.keymetrics.io/) runs the node app on port 8000 for the [Nginx](https://www.nginx.com/) reverse proxy to https.icw.cise.ufl.edu. (SSL certificate self-generated and managed using [Certbot](https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04)). MongoDB runs as a daemon as configured per DigitalOcean documentation:
https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-18-04.

Only the icwNode repository (https://github.com/SMGeller/icwNode) is located on the server at **/srv/icw/icwNode/** and it contains the static build folder from icwReact. To deploy changes to the server, make sure the apiUrl of the React code is set properly for production.



```
const apiUrl = 'https://icw.cise.ufl.edu/api/v1'
```

Build the React code and copy the \build\ folder over to the Node application as described in the local installation instructions and push the changes to the remote repository.

Pull the icwNode changes to the server using git.

```
git pull
```

Tell PM2 to restart the “server” process to apply the changes.

```
pm2 restart server
```

Restart the Nginx web server. (Only really necessary for Nginx configuration changes).

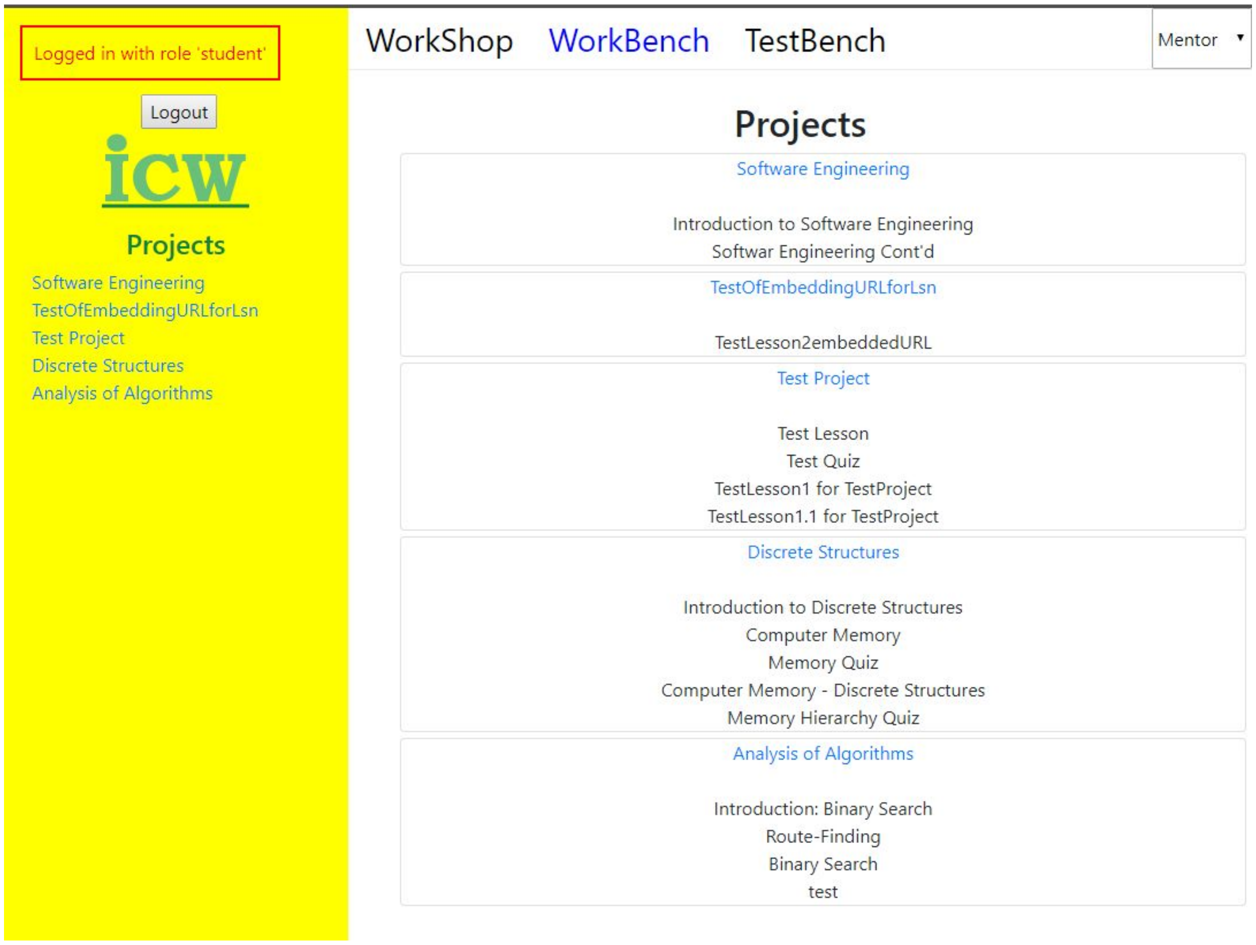
```
systemctl restart nginx
```

You might have to be connected to the UF network (on campus or through VPN) in order to access the website.

Connect ICW at https://icw.cise.ufl.edu and test your changes. You may need to refresh the page a few times so that your browser uses any UI changes you made instead of its cached version.

4. Screenshots

A few screenshots of the current application.



Logged in with role 'student'

Logout

icw

Projects

Software Engineering
TestOfEmbeddingURLforLsn
Test Project
Discrete Structures
Analysis of Algorithms

WorkShop WorkBench TestBench Mentor

Projects

Software Engineering

Introduction to Software Engineering
Softwar Engineering Cont'd

TestOfEmbeddingURLforLsn

TestLesson2embeddedURL

Test Project

Test Lesson
Test Quiz
TestLesson1 for TestProject
TestLesson1.1 for TestProject

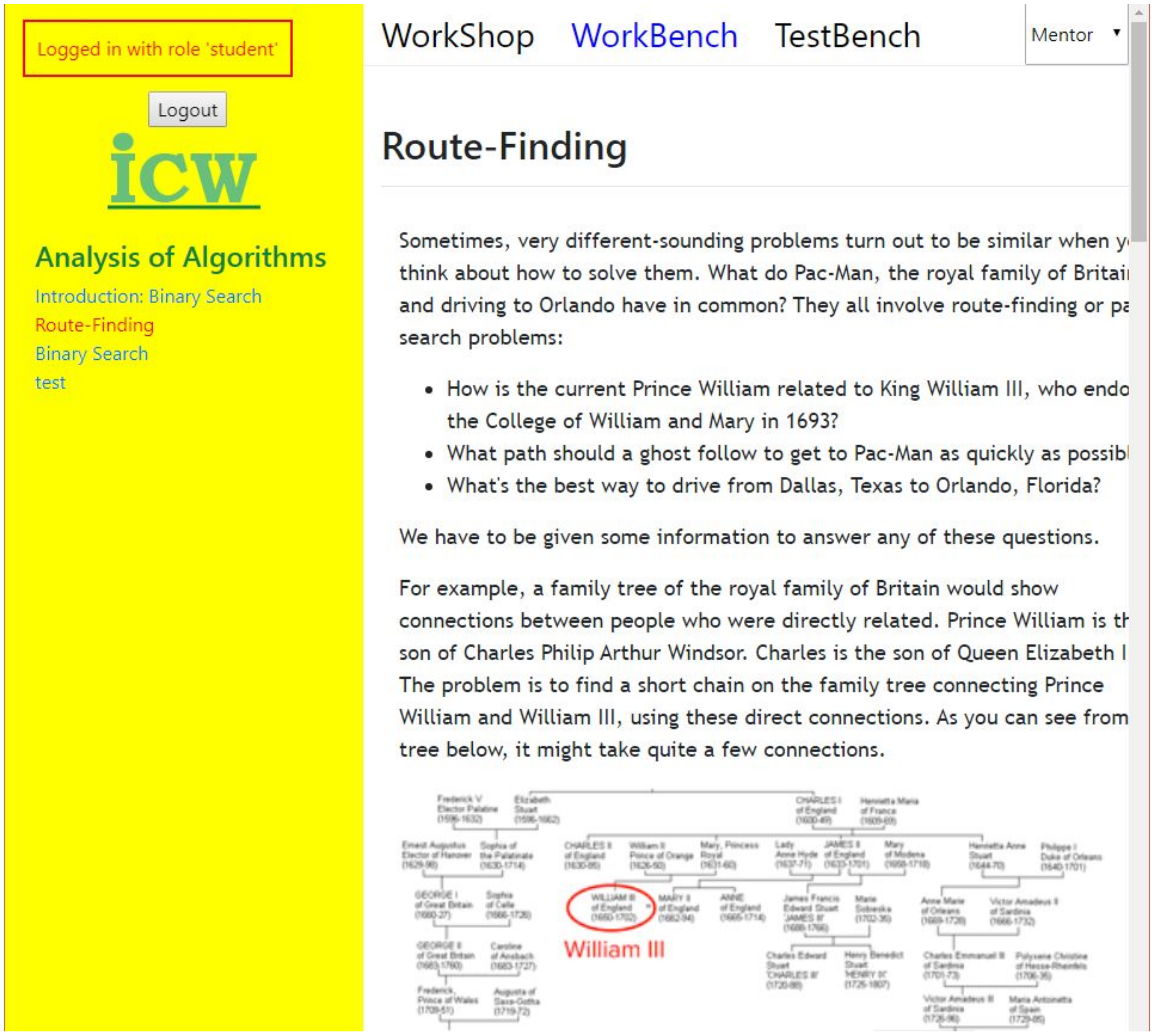
Discrete Structures

Introduction to Discrete Structures
Computer Memory
Memory Quiz
Computer Memory - Discrete Structures
Memory Hierarchy Quiz

Analysis of Algorithms

Introduction: Binary Search
Route-Finding
Binary Search
test

Workbench View



Analysis of Algorithms

Introduction: Binary Search
Route-Finding
Binary Search
test

Route-Finding

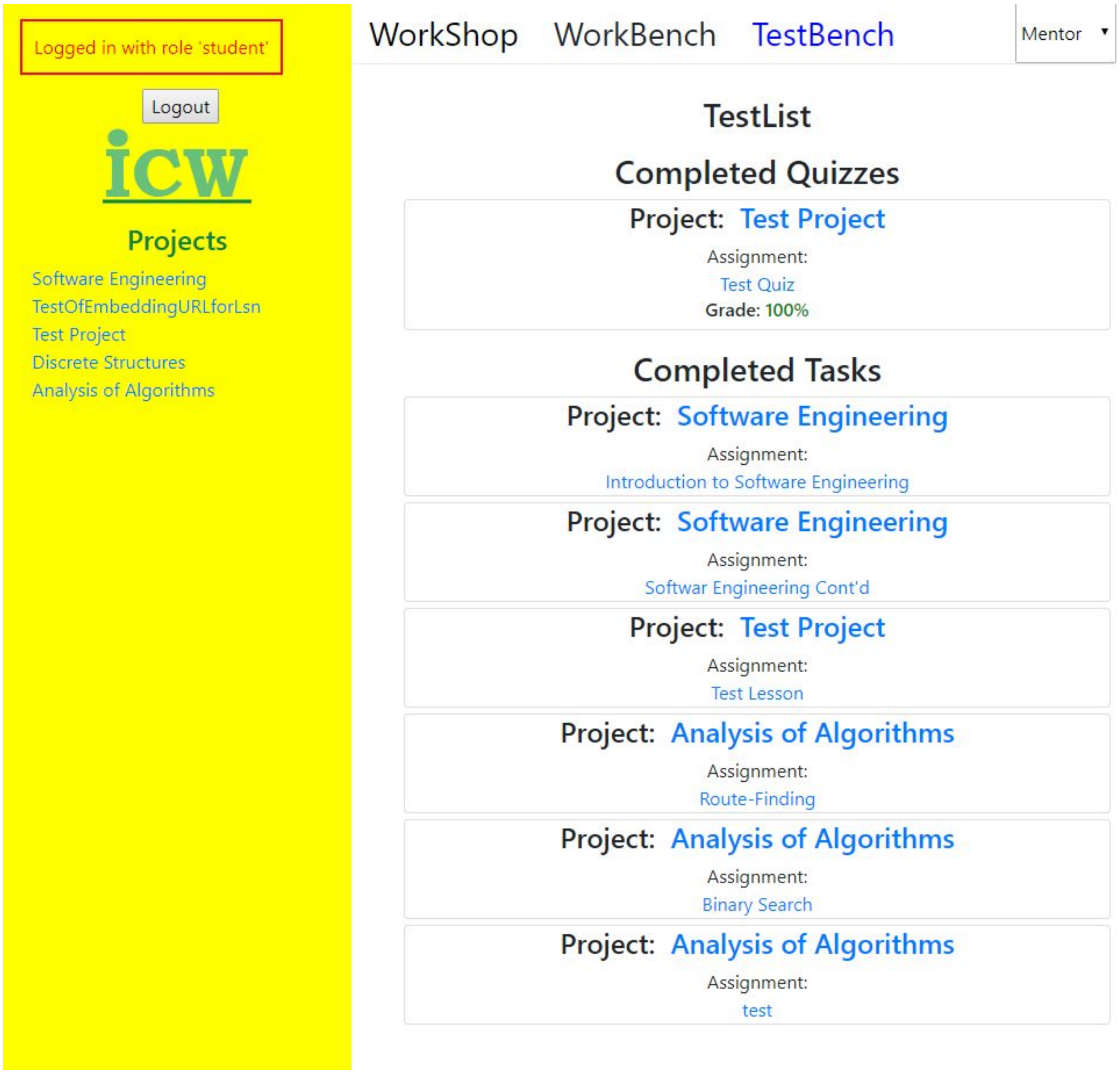
Sometimes, very different-sounding problems turn out to be similar when y think about how to solve them. What do Pac-Man, the royal family of Britai and driving to Orlando have in common? They all involve route-finding or pa search problems:

- How is the current Prince William related to King William III, who endo the College of William and Mary in 1693?
- What path should a ghost follow to get to Pac-Man as quickly as possibl
- What's the best way to drive from Dallas, Texas to Orlando, Florida?

We have to be given some information to answer any of these questions.

For example, a family tree of the royal family of Britain would show connections between people who were directly related. Prince William is th son of Charles Philip Arthur Windsor. Charles is the son of Queen Elizabeth I The problem is to find a short chain on the family tree connecting Prince William and William III, using these direct connections. As you can see from tree below, it might take quite a few connections.

Example Course


Logged in with role 'student'

Logout

icw

Projects

Software Engineering
TestOfEmbeddingURLforLsn
Test Project
Discrete Structures
Analysis of Algorithms

WorkShop WorkBench TestBench Mentor

TestList

Completed Quizzes

Project: Test Project
Assignment:
Test Quiz
Grade: 100%

Completed Tasks

Project: Software Engineering
Assignment:
Introduction to Software Engineering

Project: Software Engineering
Assignment:
Softwar Engineering Cont'd

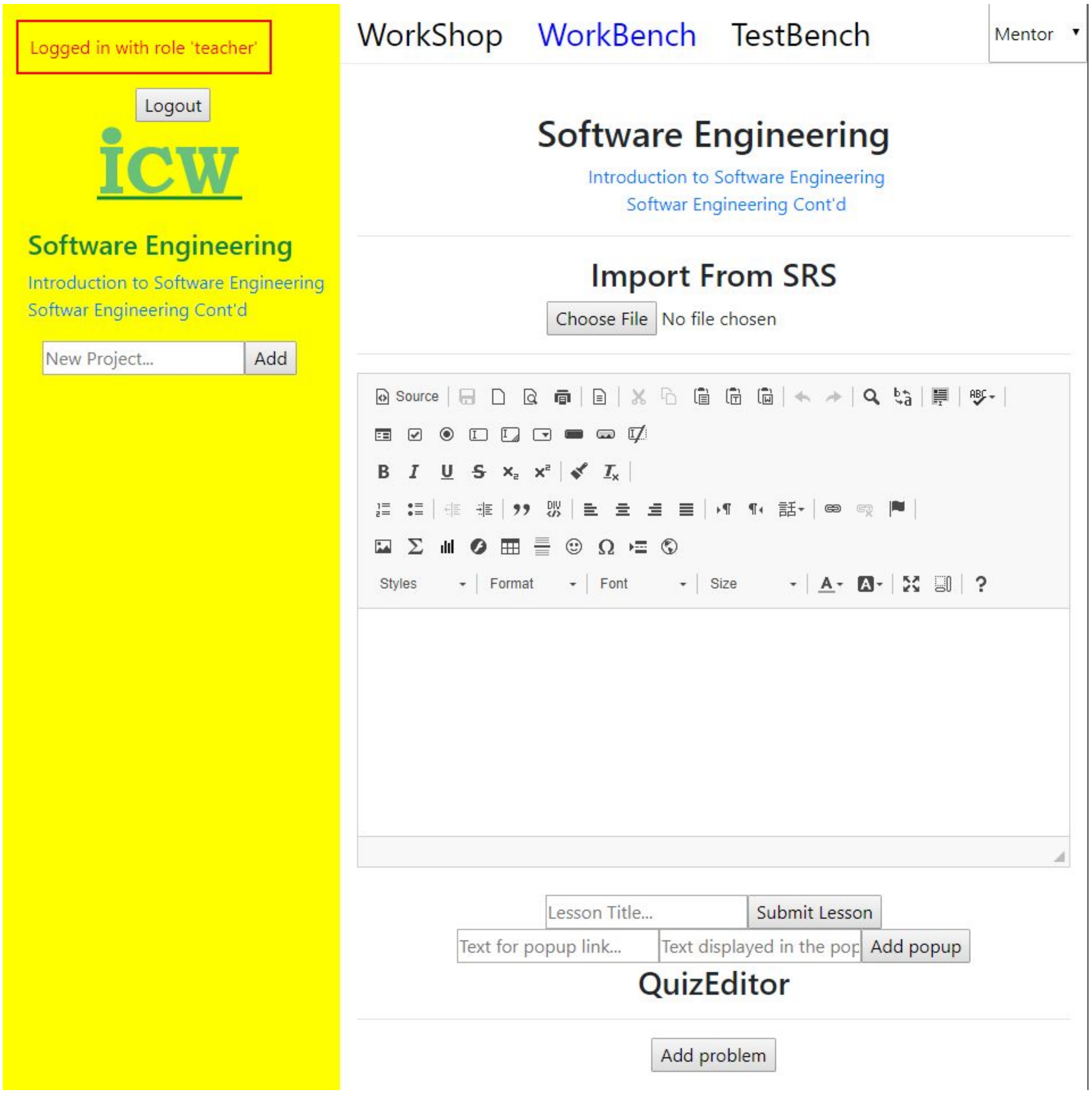
Project: Test Project
Assignment:
Test Lesson

Project: Analysis of Algorithms
Assignment:
Route-Finding

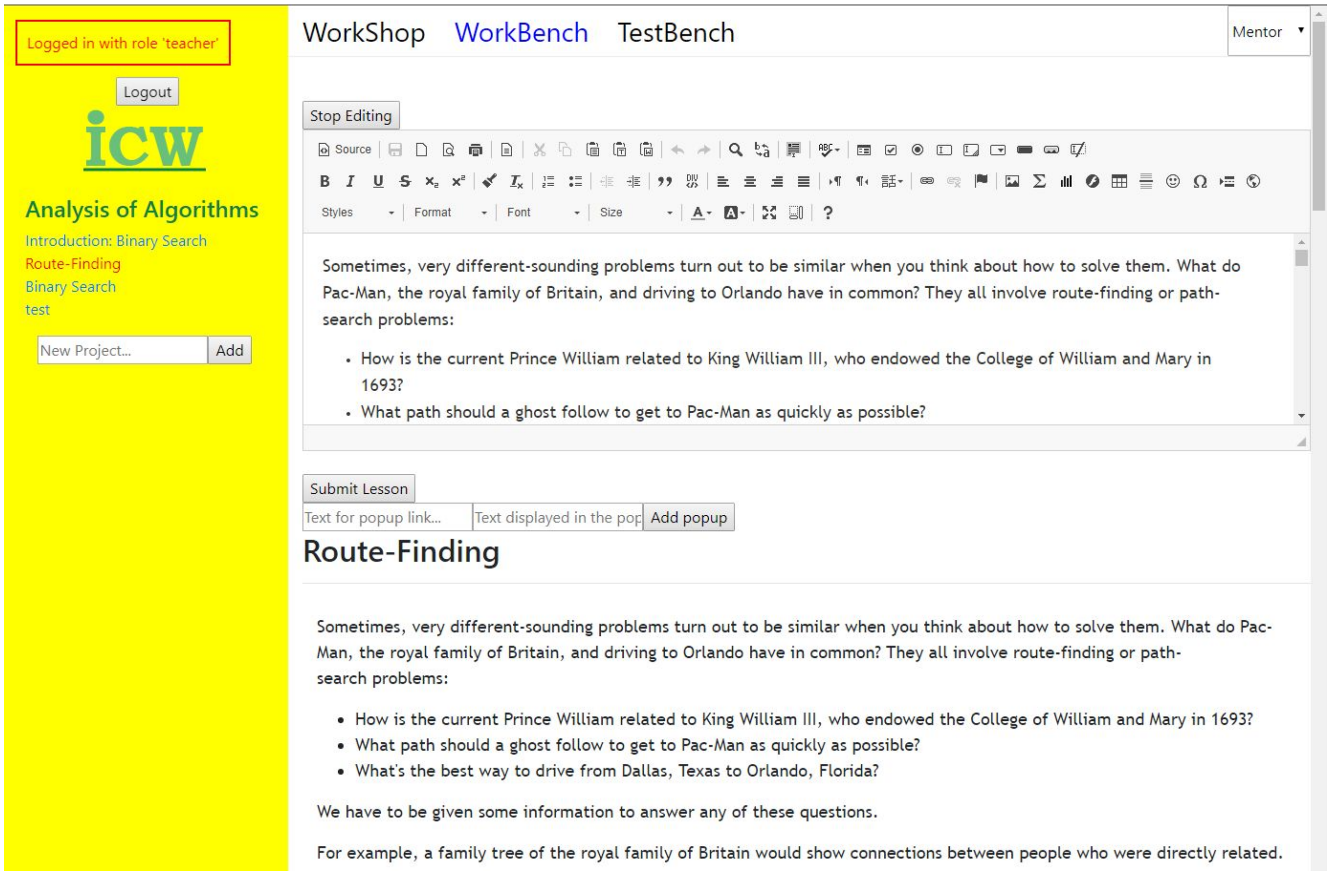
Project: Analysis of Algorithms
Assignment:
Binary Search

Project: Analysis of Algorithms
Assignment:
test

Student TestBench View



Teacher Project Editor



Sometimes, very different-sounding problems turn out to be similar when you think about how to solve them. What do Pac-Man, the royal family of Britain, and driving to Orlando have in common? They all involve route-finding or path-search problems:

- How is the current Prince William related to King William III, who endowed the College of William and Mary in 1693?
- What path should a ghost follow to get to Pac-Man as quickly as possible?

Route-Finding

Sometimes, very different-sounding problems turn out to be similar when you think about how to solve them. What do Pac-Man, the royal family of Britain, and driving to Orlando have in common? They all involve route-finding or path-search problems:

- How is the current Prince William related to King William III, who endowed the College of William and Mary in 1693?
- What path should a ghost follow to get to Pac-Man as quickly as possible?
- What's the best way to drive from Dallas, Texas to Orlando, Florida?

We have to be given some information to answer any of these questions.

For example, a family tree of the royal family of Britain would show connections between people who were directly related.

Teacher Course Editor

5. Known Issues

The CKEditor that is used to edit and create courses sometimes fails to initialize properly which is fixed by switching to a different application tab, refreshing the page, then returning to the course/project you want to edit. The application also currently does not display charts created in the CKEditor. The SRS import feature is currently just a GUI that reads files (.srsnote) and prints them to console because the backend for it still needs to be implemented.