Introduction

Team: Cross Country Coders

Organization Page: cross-country-coders.github.io

GitHub Repository: https://github.com/cross-country-coders/algo-trix

Wiki: https://github.com/cross-country-coders/algo-trix/wiki

Current Release Version: https://github.com/cross-country-coders/algo-trix/releases/tag/0.2

Team Members: Christan Jensen, Jerome Gallego, Jun Miao, Shinya Saito

Meeting Times:

- Monday: 2000 HST (Required)
- Wednesday: 2000 HST (Required)
- Friday: 2000 HST (Optional)
- Sat: 2000 HST (Required)
- Sun : 2000 HST (Optional)

Application Title:

- **AlgoTrix (Algorithm + Matrix)**

Application Type

- Web Based Application

Description:

With ICS 311 Algorithm being the gatekeeper course in the ICS community, many students are found stressed and mentally pressured to pass. The application provides those college students with additional practice/resources for Algorithms outside of the textbook where they can save their individual progress and add notes.

Functional Requirements

- User Login (Sensitive)
- Read through some topic notes
- Get practice with sample problems
- List Recommended videos
- Modify notes or problems (admin only)
- Get/Mody Registration Activity (admin only)

Development Tools:

- Javascript

- HTML
- CSS
- React
- Meteor
- Semantic UI
- Formantic UI

Requirements

1. **Security and Privacy Requirements**

   - Ensure that the user login information such as email and passwords are stored safely and can be only accessed by the admin related accounts.

   - Unless the user is admin, a normal user should not be able to see other user's information.

   - Our group uses the GitHub project page with the *kanban* style, therefore, all the issues or tasks that need to be worked on can be color coded or organized based on its immediate actions.

2. **Quality Gates (or Bug Bars)**

   - Consent Agreement before registering an account; to be able to use their information

Privacy

a. Critical ("Release may create legal or regulatory liability for the organization.")

b. Important ("Release may create high risk of negative reaction by privacy advocates or damage the organization's image.")

i. Lack of notice and consent

1. Program may record user information such as progress and past site activity without consent or notice

ii. Lack of user controls

1. Program will record user progress and past progress, user does not have the ability to opt out of data collection

c. Moderate ("Some user concerns may be raised, some privacy advocates may question, but repercussion will be limited.")

d. Low ("May cause some user queries. Scrutiny by privacy advocates unlikely.")

Security

a. Critical: *Security vulnerability that would be rated as having significant potential for damage*

- elevation of privilege
-

b. Important: *A security vulnerability that would be rated as having significant potential for damage, but less than Critical*

- Information disclosure (targeted)
- Spoofing
- Security Features

c. Moderate:*A security vulnerability that would be rated as having moderate potential for damage, but less than Important*
- Denial of Service
- Spoofing
- Security Assurances

d. Low: *A security vulnerability that would be rated as having low potential for damage*
- information disclosure
- tampering
- encryption

3. Risk Assessment Plan for Security and Privacy

a) How

- We shall use plugin like Meteor MiniMongo to check whether or not there is data leak within the application
- If a feature is going to be added to the application, the group will discuss what are the consequences (good or/and bad)

b) Parts that Require Threat Modeling

- Login
- Database and user interaction

C. Design

1. Design Requirements

A. Password Protection

1) Do not store the password in a text file
2) Encrypt the password
3) Make it so that only Admins have control over other accounts and make sure that no one can see Admin's password

B. User Data Information

1) Only a single user can see their own progress. not even the admin has access to that information
2) Data not stored in a text file

2. Attack Surface Analysis and Reduction

A. Privilege Levels

i. Regular Users

A. Can read data such as taking a look at the practice problems

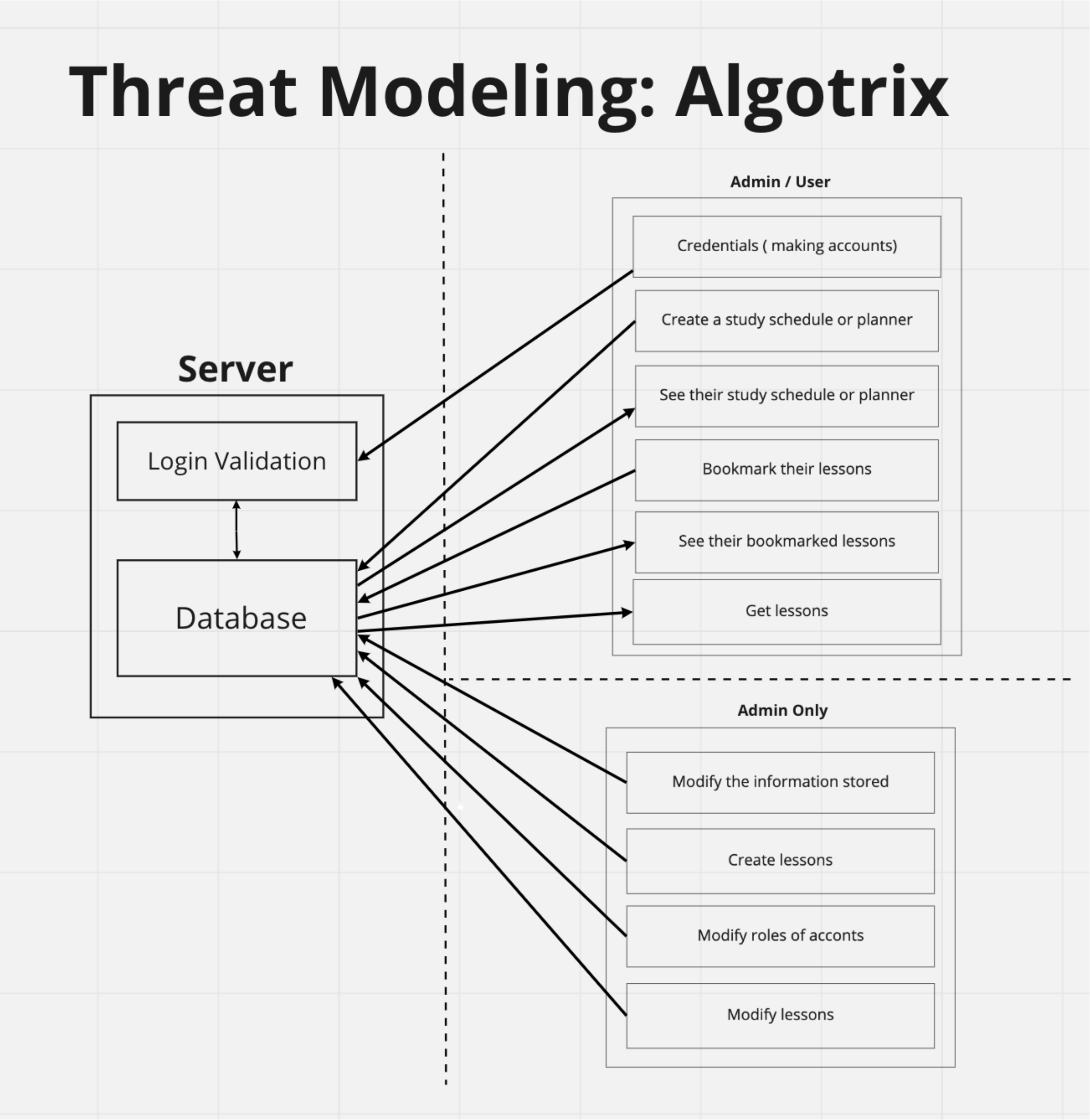
B. Can send data such as through the contact admin page

ii. Admin

A. Reads the data sent through the forms from the users of the application

B. Change or make updates to information on the application

3. Threat Modeling


Threat Modeling: Algotrix
Server
Login Validation
Database
Admin / User
Credentials (making accounts)
Create a study schedule or planner
See their study schedule or planner
Bookmark their lessons
See their bookmarked lessons
Get lessons
Admin Only
Modify the information stored
Create lessons
Modify roles of acconts
Modify lessons

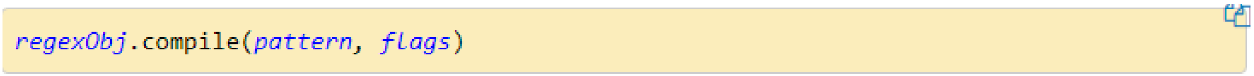
Implementation

1. Approved Tools

Tools/Frameworks	Version
IntelliJ Idea	2021.1.1
Semantic UI React	2.0.3
HTML	5
CSS	2.1
Node.JS	14.17.0
Markdown	GitHub Flavored Markdown
Meteor	2.2
GitHub Desktop	2.8.3

2. Deprecated/Unsafe Function

Method Name	What and Why
compile()	a feature not supported in some browsers and as an alternative should use a RegExp constructor



```
regexObj.compile(pattern, flags)
```

Other Alternative:

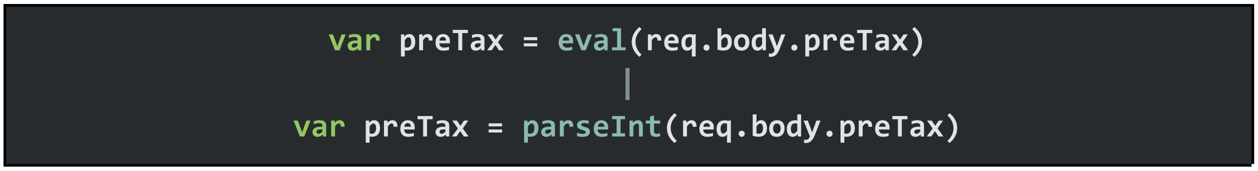
Instead of using compile(), creators should use the RegExp.

componentWillMount	Deprecated lifecycle methods that encourage unsafe coding practices, and has been replaced by **getDerivedStateFromProp**s and **getSnapshotBeforeUpdate** static methods.
componentWillReceiveProps	
componentWillUpdate	

eval()	A dangerous function because if not used properly it may result in attackers to get full access to the full production environment.

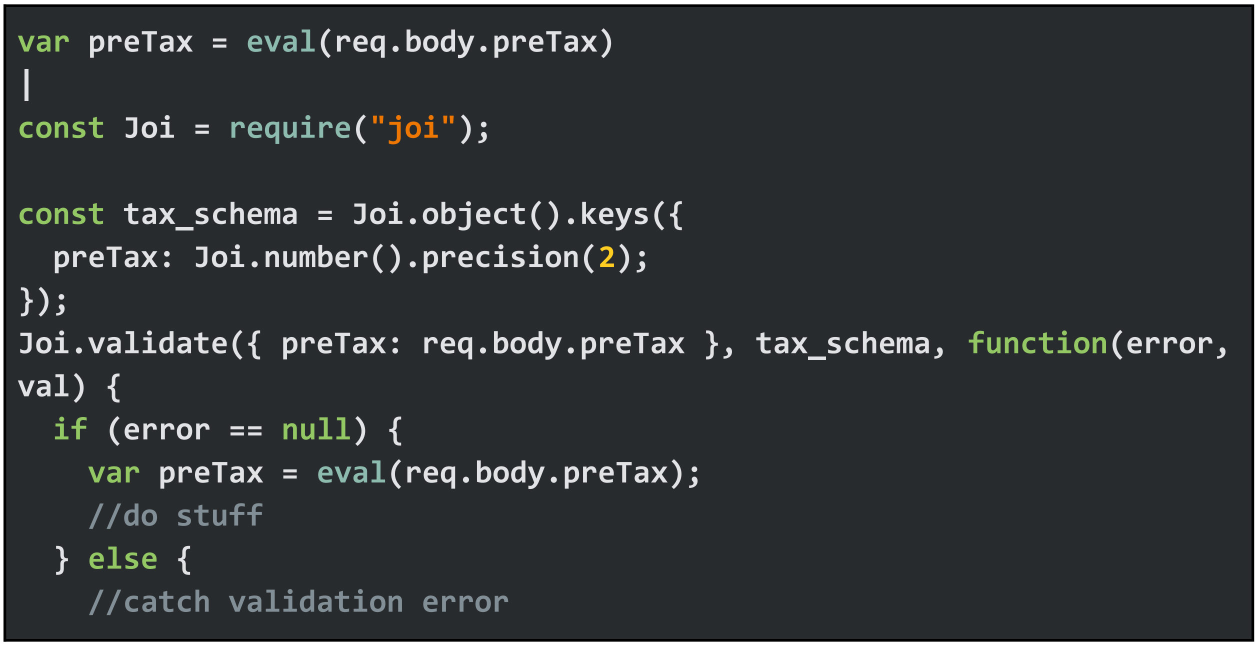
Other alternatives to eval() function:

- replacing the usage of eval() with a different function such as the following



```
var preTax = eval(req.body.preTax)
                 |
var preTax = parseInt(req.body.preTax)
```

- validate the input with the Joi package



```
var preTax = eval(req.body.preTax)
|
const Joi = require("joi");

const tax_schema = Joi.object().keys({
  preTax: Joi.number().precision(2);
});
Joi.validate({ preTax: req.body.preTax }, tax_schema, function(error,
val) {
  if (error == null) {
    var preTax = eval(req.body.preTax);
    //do stuff
  } else {
    //catch validation error
  }
});
```

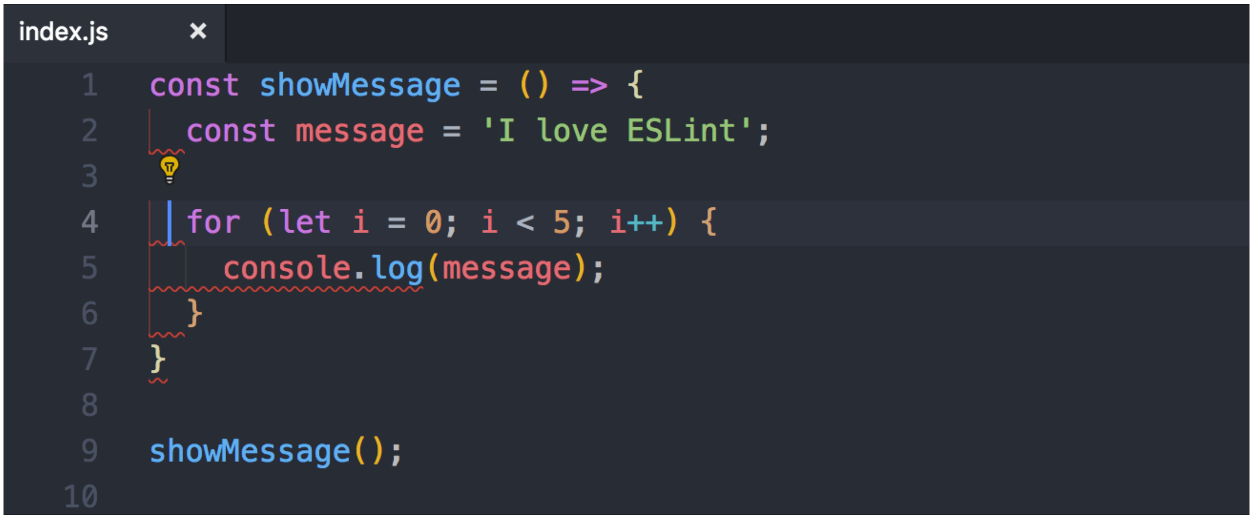
new Function()	A code that allows dynamically to define a function based on string literals but brings in potential security concerns.

Alternative to new Function()

`let func = function() { alert(value); };`

3. Static Analysis

Tool	**Version**
eslint	7.20.0



```
const showMessage = () => {
  const message = 'I love ESLint';

  for (let i = 0; i < 5; i++) {
    console.log(message);
  }
}

showMessage();
```

Figure 1: An example screenshot of ESLint being active.

We use eslint as a tool for static code analysis. It detects any problems found in our javascript code. Eslint also highlights any issues it identifies with yellow or red, yellow being a minor issue and red being a bit more serious, or if it fails our set code guidelines. Messages are displayed in the problems console in IntelliJ. There, eslint explains the error and might even suggest possible fixes. The "ics-se-code-style.xml" is the code style preference that we all use to conform to current code style conventions used by ICS 314, and 414.

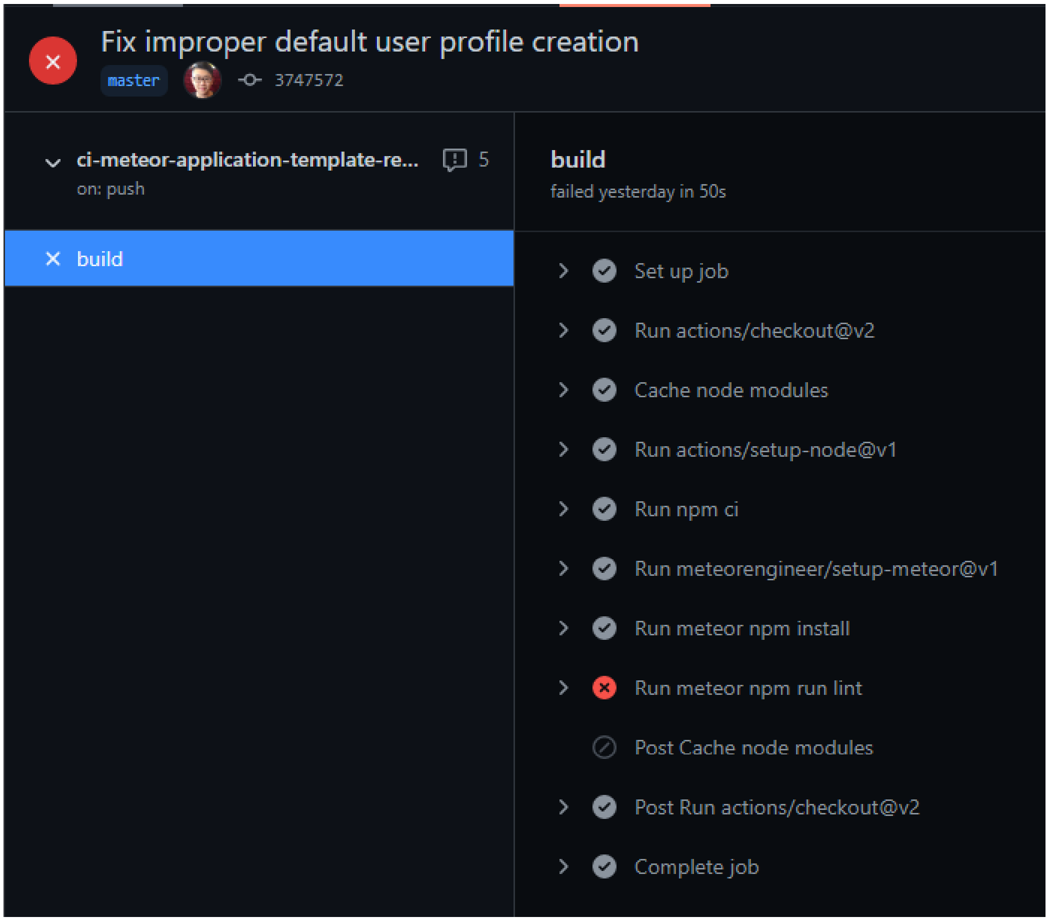
Dynamic Analysis

- GitHub (Note: Find the actual name before submission)

From our previous experience in ICS 314 and ICS 414, we have found out that using GitHub, at least for JavaScript, is efficient and useful.

Description/ How to use it?

On GitHub, every time someone works on an issue on the project, each person makes a branch labeled issue XX, where XX is the number. For example, if the issue number is 14 then the branch would be called "issue-14". After finishing their issue, the user would deploy on GitHub and then merge it into the "master" branch. Once the merge is successful, GitHub will provide an analysis based on a pass or error scale. Error indicates that the recent merge or updates had some ESLint errors or questionable choices.



✕ Fix improper default user profile creation

master · 3747572

ci-meteor-application-template-re... 5

on: push

✕ build

build

failed yesterday in 50s

- ✓ Set up job
- ✓ Run actions/checkout@v2
- ✓ Cache node modules
- ✓ Run actions/setup-node@v1
- ✓ Run npm ci
- ✓ Run meteorengineer/setup-meteor@v1
- ✓ Run meteor npm install
- ✕ Run meteor npm run lint
- ⊘ Post Cache node modules
- ✓ Post Run actions/checkout@v2
- ✓ Complete job

Figure 2: A sample rundown of the GitHub

Once the fixes are made then GitHub does another test on it to determine the outcome.

Attack Surface Review

Information based on June 9, 2021.
As of the date listed, there have not been updates made since the submission of Assignment 2. However, the factor that we have to consider is that the given time constraint of this project is short making updates not as common.

Tool	**Current Version**	**Changes made from the previous version**	**Vulnerabilities mentioned**	**Other comments**
IntelliJ Idea	2021.1.1	No change	N/A	N/A
Semantic UI React	2.0.3	No change	N/A	N/A
HTML	5	No change	N/A	N/A
CSS	2.1	No change	N/A	N/A
Node.JS	14.17.0	No change	N/A	N/A
Markdown	GitHub Flavored Markdown	No change	N/A	N/A
Meteor	2.2	No change	N/A	N/A
GitHub Desktop	2.8.3	No change	N/A	N/A

Verification

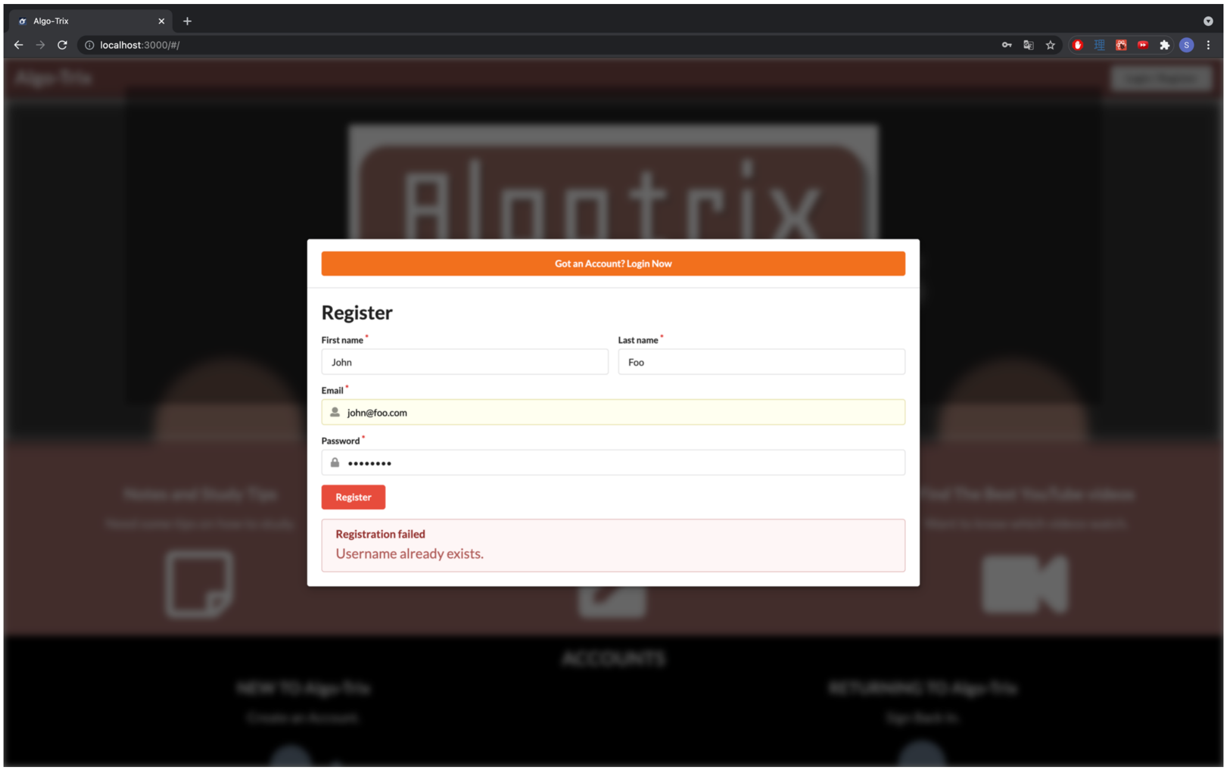
Fuzz Testing

Logging in with the incorrect password multiple times

- If a user should type in the incorrect password a few times the application should indicate that the password is incorrect the first time. However, if the user should insist on placing the wrong password the application indicates to consider taking a break.

Attempting to create an account with the same email address

- Another way we can test is to try to register an account with the same email address. For instance, if I were to register an account with the email "john@foo.com" the expected result is that it should return a message that there exists an account associated with that specific email address.



Register

First name * John

Last name * Foo

Email * john@foo.com

Password * ••••••••

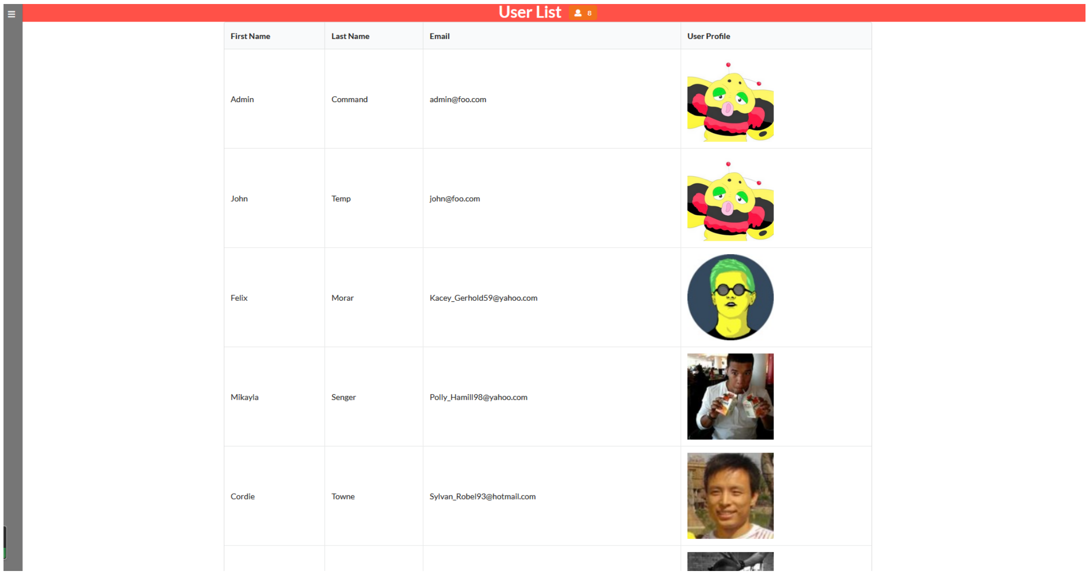
Registration failed
Username already exists.

Figure 4-1: The attempt to registering with "john@foo.com"

As Figure 4-1 shows when a user registers with "john@foo.com" as an email it returns a message that the registration failed due to the username already existing.

Attempting to access pages that are either logged in/admin access only

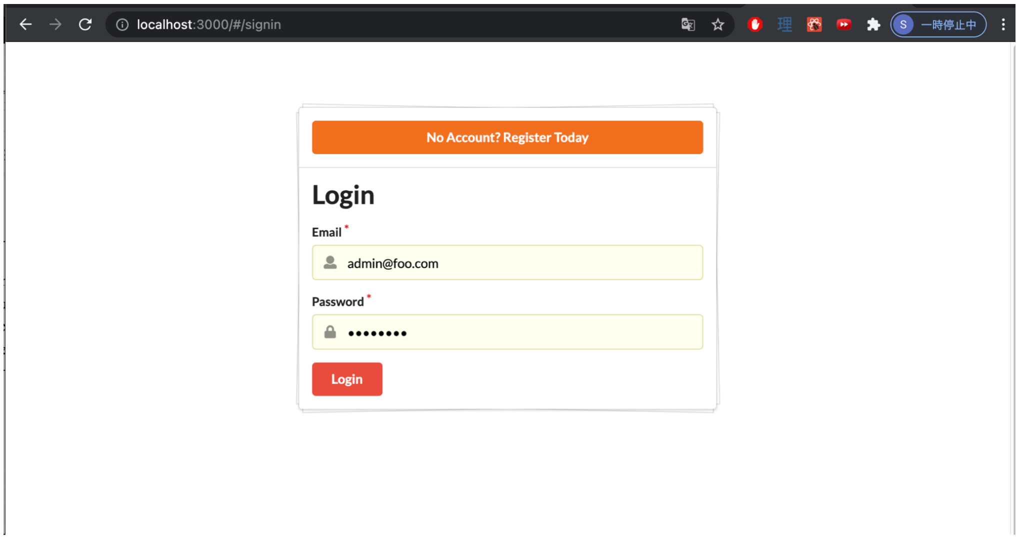
- Another way is that we can see whether directly copying the URL for restricted access pages such as the pages only can be used by admin is truly only accessible by admin. To test this we logged in to the application, copied the URL to the page and then logged out. After signing out, we directly paste the URL.
  However, the URL does not take us to the page directly, but asks the user to log in **again to the application.**



User List 8

First Name	Last Name	Email	User Profile
Admin	Command	admin@foo.com	
John	Temp	john@foo.com	
Felix	Morar	Kacey_Gerhold59@yahoo.com	
Mikayla	Senger	Polly_Hamill98@yahoo.com	
Cordie	Towne	Sylvan_Robel93@hotmail.com	

Figure 4-2: Result UserList page on Admin account



localhost:3000/#/signin

No Account? Register Today

Login

Email *

admin@foo.com

Password *

••••••••

Login

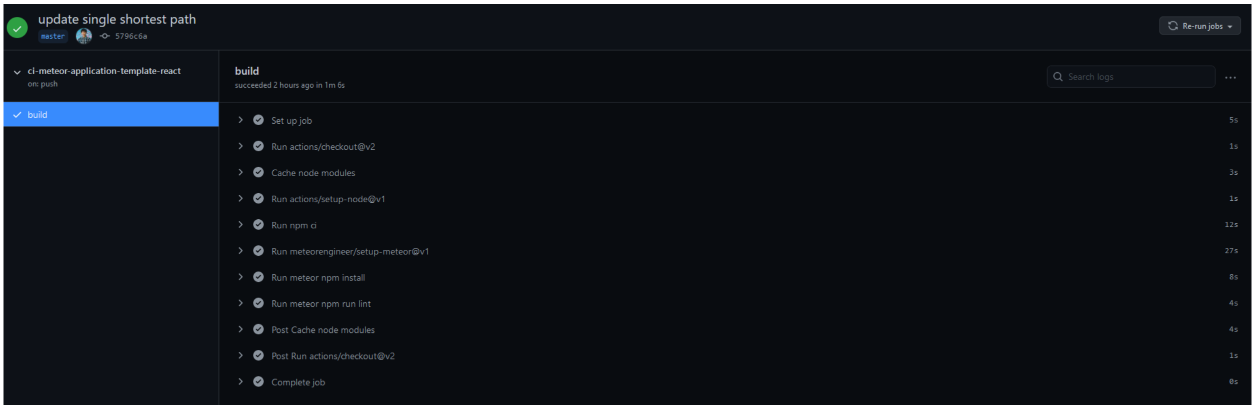
Figure 4-3: Result when just simply copying paste the URL

Static Analysis Review

We ran ESlint on our current code and there don’t seem to be any errors so far. We will fix the error that comes up throughout the development process.

Dynamic Review

The analysis result of our latest Github commit is green. If there is any issue or error, we will fix it as we see fit.



```
update single shortest path
master  5796c6a

ci-meteor-application-template-react
on: push

build

build
succeeded 2 hours ago in 1m 6s

Set up job                              5s
Run actions/checkout@v2                 1s
Cache node modules                      3s
Run actions/setup-node@v1               1s
Run npm ci                             12s
Run meteorengineer/setup-meteor@v1     27s
Run meteor npm install                  8s
Run meteor npm run lint                 4s
Post Cache node modules                 4s
Post Run actions/checkout@v2            1s
Complete job                            0s
```

Final Security Review, Incident Response Plan, Archive Report, User Guide

Incident Response Plan

Privacy Escalation Team

- **Shinya Saito (**Escalation Manager)
    - divides up the to do and the issues between the teammates
    - create the list of issues and organizes which requires immediate action or not
    - will supervise the team and the progress
    - create a plan or schedule to dealing with the issues
    - create a list of what the public relation representative could say during a press conference based on current status.

- **Jerome Gallego (**Legal Representative)
    - Responsible for the legality of the created application
    - Represents our organization if we are faced with a lawsuit
    - Ensures that all official documents are in order in case of some form of legal action

- **Christian Jensen (**Public Relations Representative)
    - Listens to users complaints and comments
    - Organization representative, boost brand reputation
    - Warn users of any potential threats or issues with application and provide assurance that we are on route to get it fixed ASAP.
    - Promote organization/app/brand with media and ads.
    - Find creative ways to help people discover organization/application through ads.

  - Media awareness to counter bad press.
  - Present application features and uses appropriately.

- **Jun Miao** (Security Engineer)
  - Identify and locate security vulnerabilities within the application
  - Provide optimal solutions to security vulnerabilities
  - Ensure each rollouts are up to the set security standards

Email Contact

As of right now our group is planning to create three email accounts for separate purposes.

- email address A (algotrixQA@hawaii.edu)
  Our group has created a contact admin page in which users can inquire the organization about questions or place comments. This email address will be the in box for the contact page and will be checked on a basis of every two hours during rush hour and once a day during weekends or holidays. This email would be used for general inquiry listed in the organization page.
- email address B (algortix_relation@hawaii.edu)
  This email address can be used for public relations such as for mass media to make inquiries about the application.
- email address C (algotrix@hawaii.edu)
  This email will be used for legal relations and issues.

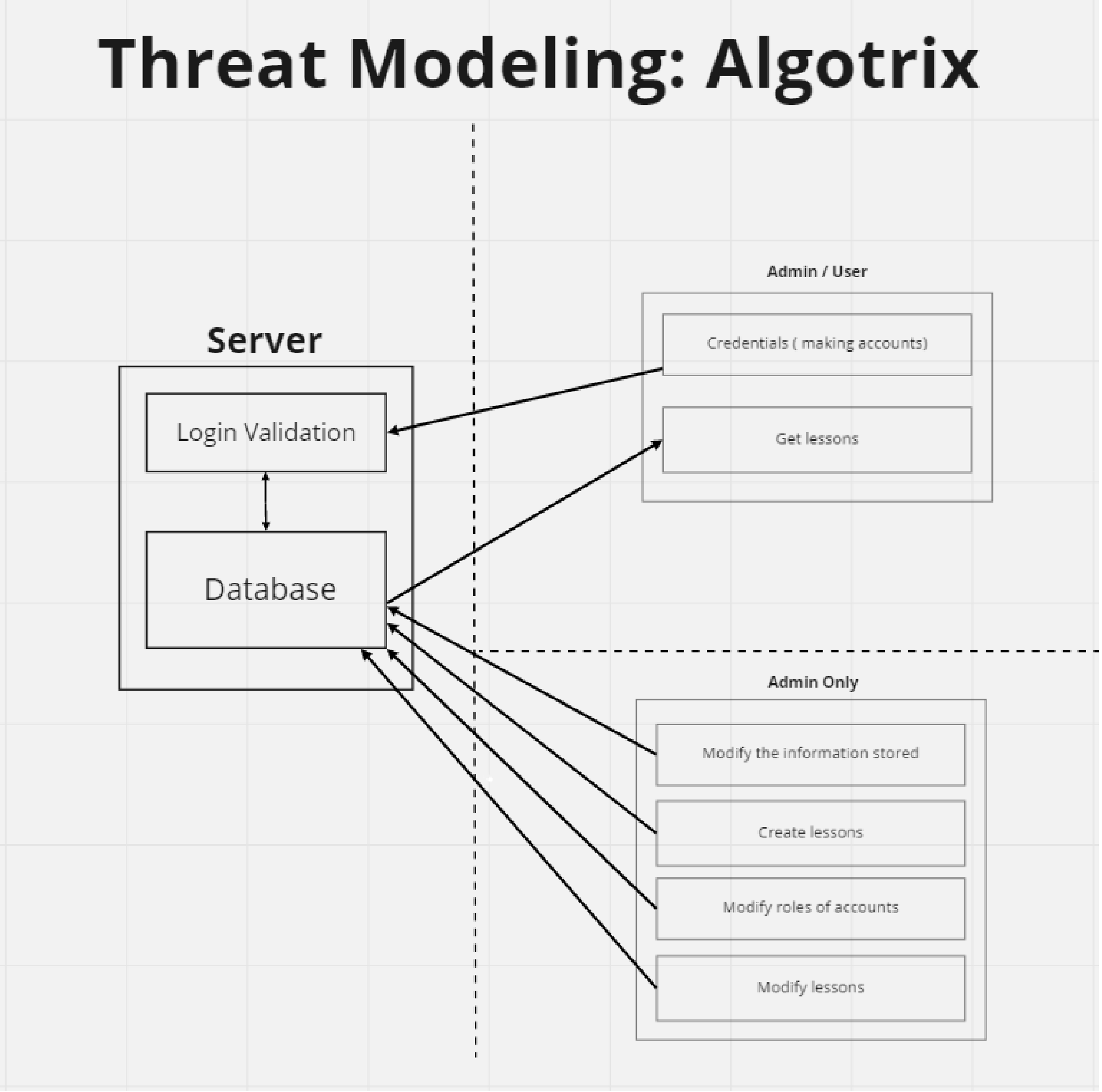
Procedure

Below here are the procedures that the organization will take when an issue happens.

- Once the issue is either discovered the
- The escalation manager will devise a plan and a schedule based on the issue. However, if there are two issues the escalation manager will compare which requires much more immediate action and then will devise a plan.

- Once the plan is created the public relation officer will notify the public or users and release the paraphrased down plan on how the issue will be dealt with.
- The escalation manager will divide up the work associated with the issue to the team members.

Final Security Review



Our threat model has changed because the functionality of the application has changed slightly from initial planning due to the six week time constraint. However, besides that nothing has vastly changed.

Performing static analysis on the code, we see that there are no new erros and all the codes follow the rules. We have also played around with the application by logging in as a student user and seeing if we could see the admin page or going to a page that needs a logged in account without logging in.

Certified Release & Archive Report

Here is the link to the release version of Algo-Trix which is version 0.01:

https://github.com/cross-country-coders/algo-trix/releases/tag/0.01

Summary of the Current Features

- secure login system
- secure admin and user accounts
- take a look at the different notes, videos, and problems covering the topics in an Introduction to Algorithm course

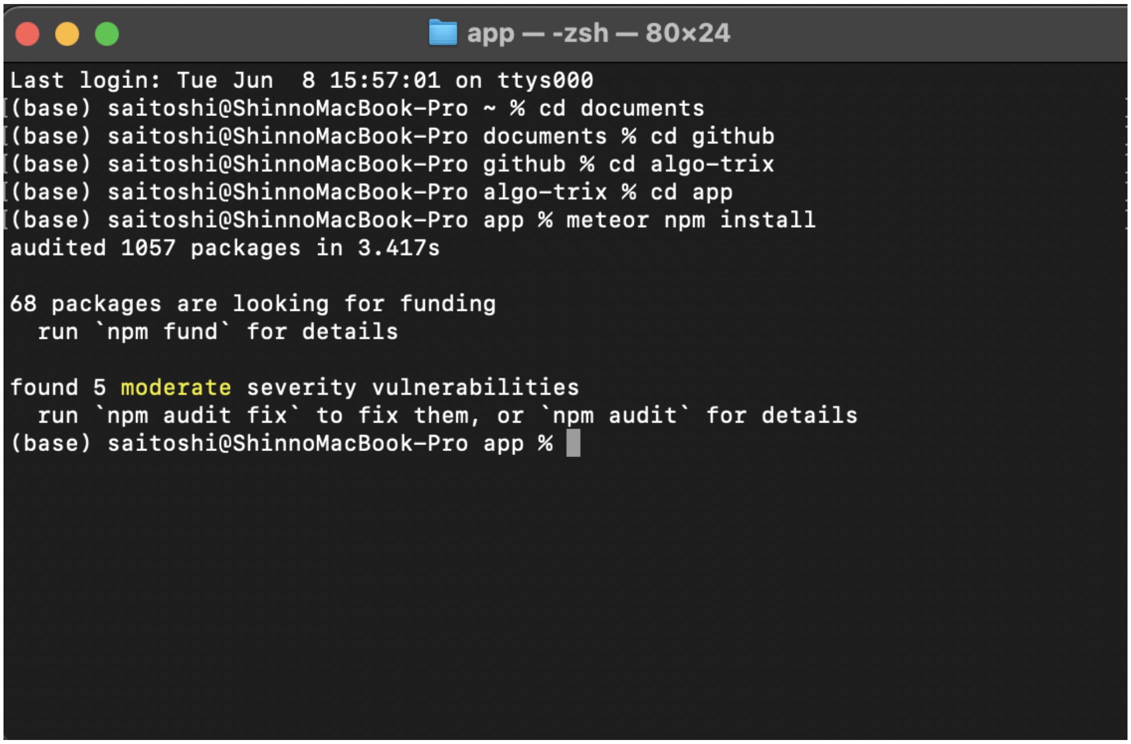
Future Developments Plans

- create a bookmark system where users can bookmark their current lesson they were working on
- create a level system that every time a user finishes a lesson they get experience points and then level up

Installation

In order to install Algo-Trix follow the steps written below

1. Go to the repository page <https://github.com/cross-country-coders/algo-trix> and then download the package of AlgoTrix from GitHub.
2. Then using the terminal in your computer go to where the downloaded package is saved. Then, install meteor in the app folder using this following command: meteor npm install.



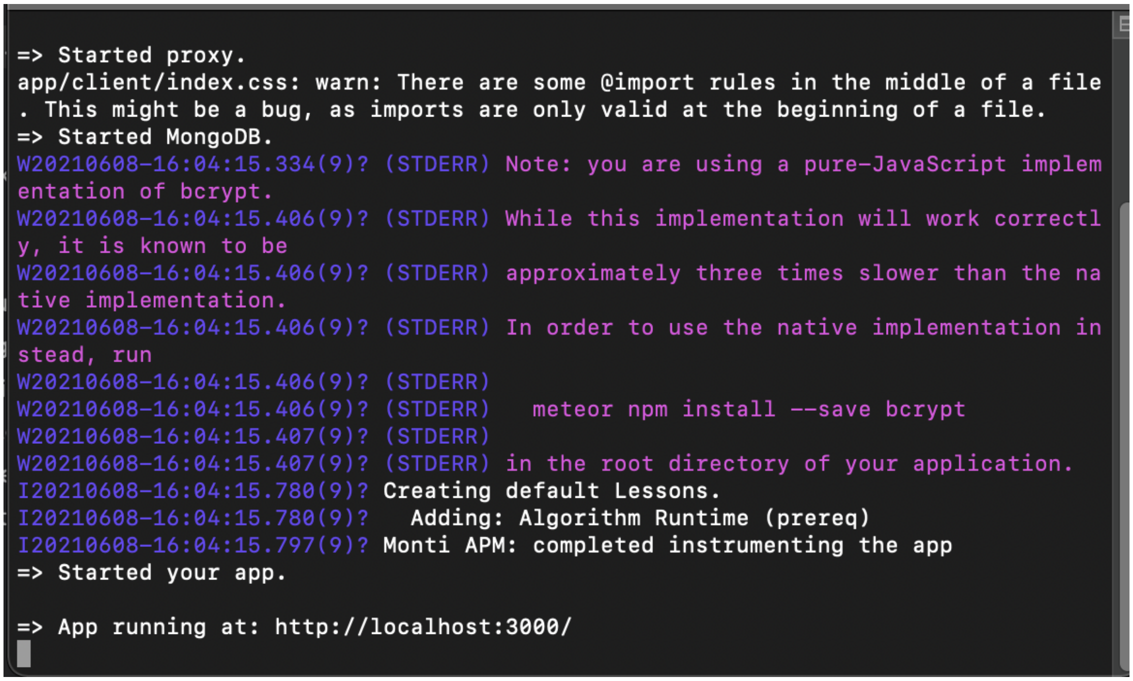
```
Last login: Tue Jun  8 15:57:01 on ttys000
(base) saitoshi@ShinnoMacBook-Pro ~ % cd documents
(base) saitoshi@ShinnoMacBook-Pro documents % cd github
(base) saitoshi@ShinnoMacBook-Pro github % cd algo-trix
(base) saitoshi@ShinnoMacBook-Pro algo-trix % cd app
(base) saitoshi@ShinnoMacBook-Pro app % meteor npm install
audited 1057 packages in 3.417s

68 packages are looking for funding
  run `npm fund` for details

found 5 moderate severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details
(base) saitoshi@ShinnoMacBook-Pro app % 
```

Figure 5-1: A sample screenshot of the terminal

3. Then type in meteor npm run start then go to the following link http://localhost:3000 to see the application running locally on your computer.



```
=> Started proxy.
app/client/index.css: warn: There are some @import rules in the middle of a file
. This might be a bug, as imports are only valid at the beginning of a file.
=> Started MongoDB.
W20210608-16:04:15.334(9)? (STDERR) Note: you are using a pure-JavaScript implem
entation of bcrypt.
W20210608-16:04:15.406(9)? (STDERR) While this implementation will work correctl
y, it is known to be
W20210608-16:04:15.406(9)? (STDERR) approximately three times slower than the na
tive implementation.
W20210608-16:04:15.406(9)? (STDERR) In order to use the native implementation in
stead, run
W20210608-16:04:15.406(9)? (STDERR)
W20210608-16:04:15.406(9)? (STDERR)   meteor npm install --save bcrypt
W20210608-16:04:15.407(9)? (STDERR)
W20210608-16:04:15.407(9)? (STDERR) in the root directory of your application.
I20210608-16:04:15.780(9)? Creating default Lessons.
I20210608-16:04:15.780(9)?   Adding: Algorithm Runtime (prereq)
I20210608-16:04:15.797(9)? Monti APM: completed instrumenting the app
=> Started your app.

=> App running at: http://localhost:3000/
```

Figure 5-2: A sample screenshot of the terminal

At this point you should have Algo-Trix running. However, if you update the database associated with the application make sure to run meteor reset. Which can be done in the terminal by typing in meteor reset.

Uninstalling Algo-Trix

If you wish to uninstall Algo-Trix from your local computer, delete the algotrix folder.