

ALLENYOUNG.DEV

An Asian-American man doing software development

Allen Young's Stockmarket

**A full-stack web application for operating a private online stockmarket in the
cloud**

Allen Young

An Asian-American software developer

March 14, AD 2023

ALLENTOUNG.DEV

License

This version of *Asian-American Man Cloud Database Tester* software and associated documentation files are released to the public domain, for the benefit of the global public and software developers.

Software and its documentation in the public domain can be modified, distributed, or sold even without any attribution by anyone.

For more info on public-domain software, visit https://en.wikipedia.org/wiki/Public-domain_software.

Table of Contents

I. Overview.....	1
1. Software program type.....	1
2. Web App Design.....	1
3. HTTPS use, no nonce use, transmission of plain-text password to server via HTTPS.....	1
4. Password salt, and salt-hashed password storage in the backend database.....	1
5. Cookie authentication.....	1
6. Full-stack web app software development and deployment overall process.....	2
7. Single-source-of-truth settings files usage in both frontend and backend implementations..	2
8. Online payment processing.....	2
II. How to Use This Software.....	3
9. Demo trial at AllenYoung.dev/Stockmarket.....	3
10. How to deploy Allen Young's Stockmarket to an AWS LightSail multisite WordPress server.....	3
10.1. How to set up an AWS account for Allen Young's Stockmarket deployment.....	3
10.2. How to set up an AWS LightSail multisite WordPress instance for Allen Young's Stockmarket deployment.....	4
10.2.1. References.....	4
10.2.1.1. How to check for Linux distribution and version.....	4
10.2.1.2. How to install Git on Debian.....	4
10.2.1.3. How to clone Git repository.....	4
10.2.1.4. Linux how to delete directory recursively.....	5
10.2.1.5. Setting up a proxy server on Apache web server.....	5
10.2.1.5.1. Decision to not use Apache web server proxy.....	8
10.2.1.5.2. Ports and URL used.....	8
10.2.1.6. Is - allowed in URL.....	8
10.2.1.7. Does SSL certificate cover all the port numbers?.....	8
10.3. How to set up Jenkins on Ubuntu (22.04.2 LTS) Linux for Allen Young's Stockmarket deployment automation.....	9
10.3.1. NOTES.....	11
10.3.2. Deprecated.....	14
10.3.3. References.....	15
10.3.3.1. How to check Ubuntu version.....	15
10.3.3.2. How to enable root password on Ubuntu.....	15
10.3.3.3. How to boot into root user shell recovery mode in Ubuntu 22.04.2 LTS.....	15
10.3.3.4. How to install Selenium on Linux Ubuntu.....	15
10.3.3.5. Linux apt vs. apt-get.....	16
10.3.3.6. Linux how to start Google Chrome in terminal.....	16
10.3.3.7. Linux Google Chrome "Missing X server or \$DISPLAY" error.....	16
10.3.3.8. Linux how to run Google Chrome as jenkins user.....	16
10.3.3.9. Linux Ubuntu how to enable a user to use display.....	17
10.3.3.10. Linux Ubuntu how to install X server for jenkins user.....	18
10.3.3.11. Linux Ubuntu how to check terminal shell type.....	18

10.3.3.12. What is .profile for.....	18
10.3.3.13. When does .profile get executed?.....	18
10.4. How to auto-deploy Allen Young's Stockmarket to an AWS LightSail multisite WordPress instance using Jenkins.....	18
10.5. How to manually deploy Allen Young's Stockmarket to an AWS LightSail multisite WordPress instance.....	19
10.5.1. References.....	20
10.5.1.1. Linux file copy command.....	21
10.5.1.2. Linux command for replacing string in multiple files.....	21
10.5.1.3. React production library URLs.....	21
10.5.1.4. Bash shell command string substitution.....	22
10.5.1.5. Linux direct command output to null.....	22
10.5.1.6. Linux command on a separate process with output redirect.....	22
11. How to deploy Allen Young's Stockmarket to a running AWS Fargate instance.....	22
11.1. How to create a Docker image for deploying Allen Young's Stockmarket to a running AWS Fargate instance.....	22
11.1.1. Dockerfile content.....	22
11.2. How to deploy a Docker image as a new AWS Fargate instance for deploying Allen Young's Stockmarket to a running AWS Fargate instance (word this better, something more specific than just 'a Docker image', something tied to Allen Young's Stockmarket).....	23
11.3. How to deploy Allen Young's Stockmarket to a running AWS Fargate instance.....	23
12. How to deploy Allen Young's Stockmarket as an AWS Fargate blue/green deployment with Docker image auto-building by Jenkins.....	23
13. How to deploy Allen Young's Stockmarket with load-balancing multiple frontend AWS Fargate instances, load-balancing multiple user-facing or frontend-facing backend AWS Fargate instances, and one database backend AWS Fargate instance.....	23
14. How to deploy Allen Young's Stockmarket with load-balancing multiple frontend AWS Fargate instances, load-balancing multiple user-facing or frontend-facing backend AWS Fargate instances, and load-balancing multiple database backend AWS Fargate instances.....	23
15. How to deploy Allen Young's Stockmarket with load-balancing and auto-scaling multiple frontend AWS Fargate instances, load-balancing and auto-scaling multiple user-facing or frontend-facing backend AWS Fargate instances, and load-balancing and auto-scaling multiple database backend AWS Fargate instances.....	24
III. Web-App Frontend (UI) Design (with URIs for routing).....	24
16. [3/9/2023 4:38 PM CST] [4/14/2023 1:43 PM CST](stock stats update).....	24
16.1. Stockmarket home webpage.....	24
16.1.1. References.....	25
16.2. Buy checkout webpage.....	26
16.2.1. Purchase receipt page.....	26
16.2.2. Handling backward or forward web browser button to navigate back to the Checkout page.....	27
16.3. User registration webpage.....	29
16.3.1. Error notification message(s).....	29
16.3.2. Email address verification notification webpage.....	29
16.3.3. Email address verification email content.....	30
16.3.4. Email address verification success notification webpage.....	30

16.3.5. Password reset email content.....	31
16.3.6. Password reset page.....	31
16.3.6.1. HTTPS URI encryption.....	31
16.3.7. Password reset success notification webpage.....	31
16.3.8. (References).....	32
16.4. Login webpage.....	32
16.4.1. Error notification message(s).....	33
16.5. User Account Webpage.....	33
16.6. Admin Webpage.....	34
16.7. Login required notification webpage (for backend URIs only).....	35
17. DEPRECATED [3/9/2023 4:38 PM CST].....	35
17.1. Stock Buy and Sell Webpage.....	35
17.1.1. Notes.....	36
17.2. Buy page.....	36
17.3. Buy checkout page.....	37
17.4. Purchase receipt page.....	37
17.5. Sell page.....	38
17.6. User registration webpage.....	39
17.7. Email address verification notification webpage.....	39
17.8. User Account Webpage.....	39
17.9. Admin Webpages admin account management webpage(s).....	40
18. DEPRECATED [3/9/2023 2:08 PM CST].....	41
19. DEPRECATED [3/8/2023 6:37 PM EST].....	41
19.1. Buy tab.....	42
19.2. Sell tab.....	42
20. Reference.....	42
20.1. UI/UX design guides.....	42
20.2. Color code reference.....	43
20.3. HTML grid.....	43
IV. Web-App Backend Design.....	43
21. Stock Database.....	43
22. User Database.....	43
23. Payment Processing Via PayPal.....	43
24. RESTful API.....	43
25. GraphQL API.....	44
V. Different Implementation Editions and Software Frameworks Used.....	44
26. Frontend.....	44
27. Backend.....	44
28. Full-stack.....	45
VI. Backend I/O.....	45
29. Database schema.....	45
29.1. Stock stats table columns.....	45
29.2. Stock reservation table columns.....	46
29.3. Users table columns.....	46
29.4. Transactions table columns.....	46
30. Backend routes and I/Os.....	47

30.1. Stockmarket home page routes.....	47
30.2. Login webpage routes.....	47
30.3. Buy checkout page routes.....	48
30.4. User registration webpage routes.....	49
30.4.1. References.....	50
30.5. User Account Webpage routes.....	50
30.6. Admin Webpage routes.....	51
30.7. (References).....	52
30.7.1. HTTP request types.....	52
30.7.2. HTTP request type for login.....	52
31. Cookies.....	52
32. OpenAPI document specification strategies and references.....	52
VII. Backend I/O Processes Spec.....	52
33. Stockmarket home webpage route I/O processes spec.....	53
33.1. /CheckAccessTokenValidity route I/O process spec.....	53
33.2. /GetCompanyStockData route I/O process spec.....	53
33.3. /GetNumberOfCompanySharesOwnedByUser route I/O process spec.....	53
34. Login webpage route I/O processes spec.....	54
34.1. /Login route I/O process spec.....	54
34.1.1. Login cybersecurity measures.....	54
34.1.2. Reference.....	55
34.2. /Logout route I/O process spec.....	56
34.2.1. Reference.....	56
35. Checkout webpage route I/O processes spec.....	57
35.1. /Checkout/:numberOfSharesToBuy route I/O process spec.....	57
35.2. /Checkout/api/orders route I/O process spec.....	58
35.2.1. Double web-browser tab checkout case handling.....	59
35.2.2. Notes.....	59
35.3. /Checkout/api/orders/:orderID/capture route I/O process spec.....	60
35.3.1. PayPal API order expiration.....	61
35.4. /Receipt/:orderID route I/O process spec.....	61
35.5. Design case considerations.....	61
35.5.1. User logout and login during transaction and access token change.....	61
35.5.2. Transaction initiation success, payment completion failure.....	62
35.6. References.....	62
36. User registration webpage route I/O processes spec.....	62
36.1. /CheckEmailAddress route I/O process spec.....	62
36.2. /RegisterUser route I/O process spec.....	62
36.3. /SendEmailAddressVerificationEmail/:accessToken/:emailAddress route I/O process spec.....	63
36.4. /GetEmailAddressVerificationSuccessPage/:accessToken route I/O process spec.....	63
36.5. /SendResetPasswordEmail/:emailAddress route I/O process spec.....	63
36.6. /ResetPassword route I/O process spec.....	64
37. User account webpage route I/O processes spec.....	64
37.1. /GetUserAccountSettings route I/O process spec.....	64
37.2. /UpdateUserAccountSettings route I/O process spec.....	64

37.3. /GetNumberOfUserStockTransactions route I/O process spec.....	64
37.4. /GetUserStockTransactionHistory/:pageNumber route I/O process spec.....	65
38. Admin webpage route I/O processes spec.....	65
38.1. /GetTotalNumberOfUsers route I/O process spec.....	65
38.2. /GetUserList/:pageNumber route I/O process spec.....	65
38.3. /GetTotalNumberOfTransactions route I/O process spec.....	65
38.4. /GetTransactionHistory/:pageNumber route I/O process spec.....	66
39. References.....	66
39.1. Difference between HTTP GET and POST.....	66
39.2. HTTP error types.....	66
VIII. Implementation Methods.....	66
40. Project directory structure.....	66
41. Multiple database management systems support method.....	67
IX. (move this to Udemy product document) Backend Implementation Methods.....	68
42. Nest.js implementation methods.....	68
42.1. Nest.js backend database access method.....	68
43. Next.js implementation methods.....	68
43.1. Next.js backend database access method.....	68
44. Python Flask implementation methods.....	68
44.1. Python Flask backend database access method.....	68
44.1.1. Python SQLite3 library.....	68
45. Golang Gin implementation methods.....	68
45.1. Golang-Gin backend database access method.....	69
45.1.1. Golang SQLite3 library.....	69
46. PHP Laravel implementation methods.....	69
46.1. PHP Laravel backend database access method.....	69
46.1.1. PHP SQLite3 library.....	69
47. Ruby Roda implementation methods.....	69
47.1. Ruby Roda backend database access method.....	69
47.1.1. Ruby SQLite3 library.....	69
48. C# ASP.NET Core controller-based API backend implementation methods.....	69
48.1. C# backend database access method.....	70
48.1.1. C# SQLite3 library.....	70
49. C# ASP.NET Core minimal API backend implementation methods.....	70
50. Java Spring Boot implementation methods.....	70
51. Kotlin Spring Boot implementation methods.....	70
52. Kotlin Ktor implementation methods.....	70
53. Online payment processing implementation.....	70
53.1. Paypal payment processing implementation.....	70
X. (move this to Udemy product document) Frontend Implementation Methods.....	70
54. React Native implementation methods.....	70
54.1. React Native implementation main strategy.....	70
55. Angular implementation methods.....	70
55.1. Angular implementation main strategy.....	71
55.2. Angular project directory structure.....	71
55.3. Angular project files.....	71

56. Vue implementation methods.....	71
57. jQuery implementation methods.....	71
58. C# ASP.NET Core Razor Pages frontend implementation methods.....	71
59. C# ASP.NET Core MVC frontend implementation methods.....	71
60. C# .NET Blazor Server frontend implementation methods.....	72
61. C# .NET Blazor WebAssembly frontend implementation methods.....	72
XI. (move this to Udemy product document) Full-stack Implementation Methods.....	72
62. Python Django full-stack implementation methods.....	72
63. PHP Laravel full-stack implementation methods.....	72
64. Ruby on Rails full-stack implementation methods.....	72
65. C# and ASP.NET Core full-stack implementation methods.....	72
XII. Testing Methods.....	72
66. Frontend testing methods.....	72
66.1. Jest, Selenium, and Postman frontend-testing sequence.....	72
66.2. Jest testing source code files.....	73
66.2.1. Snapshot testing source code file.....	73
66.2.2. DOM testing source code file.....	73
67. Postman CI/CD backend testing automation.....	73
67.1. Postman backend API test implementation strategy.....	73
67.2. Postman backend API test request sequence.....	73
67.3. References.....	76
67.3.1. Cookie HTTP header.....	77
68. Jest React frontend testing automation.....	77
68.1. Jest snapshot testing error troubleshooting.....	77
68.1.1. CLI error.....	77
68.1.2. Jest testing environments.....	78
68.1.3. CLI error 2.....	78
68.1.4. jest-environment-jsdom installation.....	79
68.1.5. CLI error 3.....	79
68.1.6. Importing React library in Jest snapshot test file.....	80
68.1.7. CLI error 4.....	80
68.1.8. React test renderer.....	81
68.1.9. Downloading and using production React library files.....	81
68.1.10. CLI error 5.....	81
68.1.11. Installing React and ReactDOM npm modules in React web app project directory.....	83
68.1.12. CLI error 6.....	83
68.1.13. CLI error 7.....	85
68.1.14. CLI error 7 troubleshooting.....	85
68.1.15. JavaScript how to import only when module is imported.....	87
68.1.15.1. Try 1 (fail).....	88
68.1.15.2. Try 2 (fail).....	89
68.1.15.3. Try 3 (fail).....	90
68.1.15.4. Try 4 (fail).....	91
68.1.15.5. Try 5 (fail).....	92
68.1.15.6. Try 6 (success!).....	93

68.1.15.7. Try 7 (fail).....	94
68.1.15.8. Try 6 (success) (use this. the most proper [synchronous] code.).....	94
68.1.16. Can JavaScript module detect when it is being imported?.....	95
68.1.17. JavaScript how to check if a variable is defined?.....	95
68.1.18. JavaScript how to create global variable that can be accessed in all modules....	95
68.1.19. JavaScript global object.....	95
68.1.20. JavaScript import to global variable.....	96
68.1.21. JavaScript import to data variable.....	96
68.1.22. JavaScript what is top level body of module.....	96
68.1.23. How to call exported JavaScript function in HTML.....	96
68.2. React frontend Jest snapshot and DOM unit test cases.....	96
68.2.1. index.js Jest snapshot and DOM unit test cases.....	96
68.2.2. registration.js Jest snapshot and DOM unit test cases.....	97
68.2.3. EmailAddressVerificationNotice.js Jest snapshot and DOM unit test cases.....	97
68.2.4. login.js Jest snapshot and DOM unit test cases.....	97
68.2.5. PasswordReset.js Jest snapshot and DOM unit test cases.....	98
68.2.6. account.js Jest snapshot and DOM unit test cases.....	98
68.2.7. admin.js Jest snapshot and DOM unit test cases.....	99
69. Selenium end-to-end testing automation.....	99
69.1. Selenium end-to-end test process spec.....	99
69.1.1. Original notes.....	100
69.2. Python dependencies to install.....	100
69.3. CLI commands to execute for Selenium end-to-end testing.....	100
69.4. Email-address verification status update database Python commands.....	101
69.5. Python commands to execute for launching backend and frontend servers in the Selenium end-to-end test Python program.....	101
69.5.1. What doesn't work.....	102
69.5.1.1. [5/15/2023 4:53 PM CST].....	102
69.6. Frontend and backend servers start failure troubleshooting.....	102
69.6.1. Node.js execution file location in Windows.....	102
69.6.2. Python subprocess run env example.....	103
69.6.3. Python multiprocessing subprocess run failure.....	103
69.6.4. PermissionError: [WinError 5] Access is denied.....	103
69.6.5. EOFError: Ran out of input.....	113
69.7. Frontend server termination failure troubleshooting.....	113
69.8. Python how to exit a program with an exit code.....	117
69.9. Checkout page rendering failure troubleshooting.....	117
69.10. Troubleshooting the frontend JavaScript rendered Paypal buttons and card input fields not being found by Selenium API.....	119
69.11. References.....	119
69.11.1. Selenium Python documentation.....	119
69.11.2. Python how to check for OS type.....	120
69.11.3. Selenium Python get text.....	120
69.11.4. Selenium Python how to check if element is disabled.....	120
69.11.5. Selenium Python scroll to element.....	120
69.11.6. How to fix Python was not found on Windows.....	120

69.11.7. How to remove an installed Python version on Windows.....	120
XIII. Test Code Design.....	122
70. Unit Test Code Design.....	122
70.1. Full-Stack Web App Unit Test Code Design.....	122
70.1.1. Python/Django Full-Stack Web App Unit Test Code Design.....	122
70.1.2. PHP/Laravel Full-Stack Web App Unit Test Code Design.....	122
70.1.3. Ruby on Rails Full-Stack Web App Unit Test Code Design.....	122
70.2. Frontend Web App Unit Test Code Design.....	122
70.2.1. Angular Frontend Web App Unit Test Code Design.....	122
70.2.2. React Frontend Web App Unit Test Code Design.....	122
70.2.3. Vue Frontend Web App Unit Test Code Design.....	122
70.2.4. JQuery Frontend Web App Unit Test Code Design.....	122
70.3. Backend Web App Unit Test Code Design.....	122
70.3.1. Python/Flask Backend Web App Unit Test Code Design.....	122
70.3.2. Go/Gin Backend Web App Unit Test Code Design.....	122
70.3.3. TypeScript/Node/Express Backend Web App Unit Test Code Design.....	122
70.3.4. PHP/Laravel Backend Web App Unit Test Code Design.....	122
71. Integration Test Code.....	122
71.1. Full-Stack Web App Integration Test Code Design.....	122
71.1.1. Python/Django Full-Stack Web App Integration Test Code Design.....	122
71.1.2. PHP/Laravel Full-Stack Web App Integration Test Code Design.....	122
71.1.3. Ruby on Rails Full-Stack Web App Integration Test Code Design.....	122
71.2. Frontend Web App Integration Test Code Design.....	122
71.2.1. Angular Frontend Web App Integration Test Code Design.....	122
71.2.2. React Frontend Web App Integration Test Code Design.....	122
71.2.3. Vue Frontend Web App Integration Test Code Design.....	122
71.2.4. JQuery Frontend Web App Integration Test Code Design.....	122
71.3. Backend Web App Integration Test Code Design.....	122
71.3.1. Python/Flask Backend Web App Integration Test Code Design.....	123
71.3.2. Go/Gin Backend Web App Integration Test Code Design.....	123
71.3.3. TypeScript/Node/Express Backend Web App Integration Test Code Design....	123
71.3.4. PHP/Laravel Backend Web App Integration Test Code Design.....	123
XIV. Deployment Methods.....	123
72. Deployment configuration.....	123
72.1. How to set up the backend process manager for automatic health checkup every 30 seconds and email admin if the backend is down.....	123
72.2. Replacement of applicable settings in source code files.....	123
72.3. References.....	123
72.3.1. Is CORS needed for subdomain.....	123
73. Manual Deployment Methods.....	123
73.1. Manual backend deployment methods.....	123
73.1.1. Manual AWS Lightsail Express.js backend deployment method.....	124
73.2. Manual frontend deployment methods.....	124
73.2.1. Manual AWS Lightsail React.js frontend deployment method.....	124
XV. Potential future updates.....	124
XVI. References.....	126

XVII. Software License.....126

 74. [4/19/2023 5:48 PM CST].....126

XVIII. =====.....126

XIX. =====.....126

XX. =====.....126

XXI. =====.....126

XXII. =====.....126

I. Overview

Allen Young's Stockmarket is a full-stack web application for operating a private online stockmarket in the cloud.

Allen Young's Stockmarket doesn't have all the features to be fully compliant to SEC and state private securities sales regulations. It is a demo for making online financial transactions, especially purchases.

1. Software program type

[1/26/2023 1:41 PM EST]

A single web app, both frontend and backend.

2. Web App Design

Not too complicated in both looks and function. Simple, elegant, modern, and pleasing-looking.

(note to myself [11/26/2022 9:29 PM CST] I'm launching this in several days at Stocks.Robocentric.com. I will build what I want: absolutely no compromise on that!)

(update [11/26/2022 9:54 PM CST] I LOVE this design. I absolutely LOVE it. I'll complete this! I surely will!)

3. HTTPS use, no nonce use, transmission of plain-text password to server via HTTPS

[3/16/2023 2:25 PM CST]

https://en.wikipedia.org/wiki/Cryptographic_nonce

4. Password salt, and salt-hashed password storage in the backend database

[3/16/2023 2:25 PM CST]

5. Cookie authentication

[4/29/2023 8:36 PM CST]

<https://swagger.io/docs/specification/authentication/cookie-authentication/>

https://en.wikipedia.org/wiki/Cross-site_request_forgery#Prevention

6. Full-stack web app software development and deployment overall process

[4/22/2023 12:25 PM CST]

1. Web app UI/UX, features, and architecture design and specification.
 - 1.1. Web app UI/UX design and specification.
 - 1.2. Web app features design and specification.
 - 1.3. Web app backend API OpenAPI and/or GraphQL API design and specification.
2. Backend implementation and testing.
 - 2.1. Backend main code writing.
 - 2.2. Backend test code writing. (Postman use if and as desired and required.)
3. Frontend implementation and testing.
 - 3.1. Frontend main code writing.
 - 3.2. Frontend test code writing. (E.g. using Jest if React frontend.)
4. Additional test code writing.
 - 4.1. Postman backend test code writing. (If and as desired and required.)
 - 4.2. Selenium end-to-end (E2E) test code writing.
5. CI/CD pipeline code writing.
 - 5.1. Jenkins pipeline code writing.
6. Initial deployment.
 - 6.1. Publication to source code repository.
 - 6.2. Initial manual or fully automated deployment.

7. Single-source-of-truth settings files usage in both frontend and backend implementations

[5/26/2023 7:03 PM CST]

8. Online payment processing

Payment processing via PayPal, with automatic processing fee addition. This requires "Buy stocks" or "Stock checkout" or "Stock purchase" webpage. Contact PayPal and verify if they allow accepting payments for private stocks. Check the PayPal Checkout acceptable usage policy first. If PayPal doesn't allow it, contact other payment processors, Square, Stripe, and Amazon. ([UPDATE] Paypal does allow accepting payments for private stocks.)

Fully automated transaction record backup system. Each transaction record is an encrypted file or object in a private AWS S3 bucket. A Python or Node.js CLI program running on a local office backup computer, that listens to AWS SNS notifications. Server auto-sends an AWS SNS notification at the completion of each transaction. The client program, upon receiving the notification, downloads the AWS S3 object to local directory. At each start of the client program, it syncs the AWS S3 bucket contents leaving the files on local storage that are missing on the server. The client program has a command that accepts decryption key and decrypts all the files or one file and saves decrypted file(s) in a user-assigned directory. (This cross-server backup feature is not implemented [yet].)

Buy feature only in the AllenYoung.dev v1.0 demo. I might add sell feature later. Adding sell requires depositing into bank or PayPal account. I'll have to look into that. Search "PayPal API send money" online. When selling stock, buyer pays 6% processing fee when buying the stock. A buy is final, non-refundable.

II. How to Use This Software

(command formats)

1. Demo trial at AllenYoung.dev/Stockmarket

[5/15/2023 9:49 PM EST]

A live demo of Allen Young's Stockmarket is available at AllenYoung.dev/Stockmarket.

2. How to deploy Allen Young's Stockmarket to an AWS LightSail multisite WordPress server

[5/15/2023 9:49 PM EST] [5/27/2023 4:16 PM CST]

For automated deployment, execute "How to set up an AWS LightSail multisite WordPress instance for Allen Young's Stockmarket deployment", "How to set up Jenkins on Ubuntu (22.04.2 LTS) Linux for Allen Young's Stockmarket deployment automation", and "How to manually deploy Allen Young's Stockmarket to an AWS LightSail multisite WordPress instance".

For manual deployment, execute "How to set up an AWS LightSail multisite WordPress instance for Allen Young's Stockmarket deployment" and "How to manually deploy Allen Young's Stockmarket to an AWS LightSail multisite WordPress instance".

2.1. How to set up an AWS account for Allen Young's Stockmarket deployment

[5/27/2023 5:19 PM CST]

1. Configure AWS S3 for Allen Young's Stockmarket deployment.
 - 1.1. Create an AWS S3 bucket with a folder that Allen Young's Stockmarket will store transaction and error records in.
2. Configure AWS SES for Allen Young's Stockmarket deployment.
 - 2.1. Set up AWS SES for sending and receiving emails to and from the domain name that you will configure Allen Young's Stockmarket to use for sending and receiving emails to and from.
3. Configure AWS IAM for Allen Young's Stockmarket deployment.
 - 3.1. Create an IAM user with a policy for using the above AWS S3 write access and AWS SES email sending and receiving access. Create and download the IAM user access-key ID and secret access key.

2.2. How to set up an AWS LightSail multisite WordPress instance for Allen Young's Stockmarket deployment

[5/26/2023 10:37 AM CST]

1. Log into the AWS LightSail instance via an SSH shell.
2. Install git by executing `sudo apt update` then `sudo apt install git` (on Debian 10).
3. Make sure that Let's Encrypt SSL certificate is installed. If not, install it according to Bitnami's official instruction.
4. Make sure that node and npm are installed. If not, install them.
5. Install Node.js http-server by executing `sudo npm install http-server -g`.
Execute `http-server --version` to check the installation.
6. Create `allenyong.dev/Stockmarket` WordPress page that redirects to `allenyong.dev:8889`.
7. In the AWS LightSail web console, open the port numbers 8889 and 8890.

2.2.1. References

[5/26/2023 10:45 AM CST]

2.2.1.1. How to check for Linux distribution and version

[5/26/2023 10:45 AM CST]

Search "How to check for Linux distribution and version" online.

<https://www.tecmint.com/check-linux-os-version/>
`cat /etc/os-release`

2.2.1.2. How to install Git on Debian

[5/26/2023 10:47 AM CST]

Search "How to install Git on Debian" online.

<https://www.digitalocean.com/community/tutorials/how-to-install-git-on-debian-10>
`sudo apt update`
`sudo apt install git`
`git --version`

2.2.1.3. How to clone Git repository

[5/26/2023 10:49 AM CST]

Search "How to clone Git repository" online.

<https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository>
<https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone>
<https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository>

`git clone https://github.com/libgit2/libgit2`
`git clone https://github.com/libgit2/libgit2 mylibgit`

`git clone https://github.com/AllenYoungDev/stockmarket.git`

2.2.1.4. Linux how to delete directory recursively

[5/26/2023 10:56 AM CST]

Search "Linux how to delete directory recursively" online.

<https://docs.oracle.com/cd/E19253-01/806-7612/files-20/index.html>

<https://www.cyberciti.biz/faq/linux-delete-folder-recursively/>

```
rm -r test
```

```
rm -r -f ~/stockmarket
```

2.2.1.5. Setting up a proxy server on Apache web server

[5/26/2023 11:11 AM CST]

setting up a proxy server on Apache

https://docs.oracle.com/cd/E24001_01/webcenter.1111/e12405/wcadm_rest.htm

<http://httpd.apache.org/docs>

https://httpd.apache.org/docs/2.4/mod/mod_substitute.html

search "how to check Apache server has mod_substitute module installed" online.

apache2ctl -M (command not present)

<https://serverfault.com/questions/778334/where-to-get-apache-modules-so-files>

search "AWS LightSail multisite WordPress how to check installed Apache modules" online.

<https://docs.bitnami.com/google/apps/wordpress-multisite/administration/check-installed-modules/>

<https://docs.bitnami.com/aws/apps/wordpress/administration/check-installed-modules/>

```
sudo /opt/bitnami/apache2/bin/apachectl -M (this works!)
```

~~search "AWS LightSail multisite WordPress how to install Apache module" online. (no useful result)~~

search "AWS LightSail Bitnami how to install Apache module" online.

<https://docs.bitnami.com/aws/infrastructure/lamp/configuration/enable-modules/> (no instruction for the mod_substitute module.)

~~Search "AWS LightSail Bitnami how to install Apache mod_substitute module" online. (yields nothing useful.)~~

Search "AWS Bitnami how to install Apache mod_substitute" online. (nothing useful.)

<https://docs.bitnami.com/aws/apps/modx/administration/use-single-domain/>

Search "how to install Apache mod_substitute" online.

https://corpocrat.com/2008/09/19/install-apache-mod_substitute/

https://vishnulinix.wordpress.com/2012/06/22/enable-apache-mod_substitute/

https://talk.plesk.com/threads/enable-mod_substitute-from-command-line.349894/

```
httpd -M | grep -ci -o "substitute_module"
```

Debian/Ubuntu? Use a2enmod

Search "Bitnami how to enable Apache mod_substitute" online. (nothing useful.)

Search "how to check Linux distribution" online.
cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME="Debian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
search "How to enable Apache server module on Debian" online.
<https://www.ducea.com/2006/05/30/managing-apache2-modules-the-debian-way/>
search "Debian Linux how to search installed package" online.
<https://linuxhint.com/check-installed-package-debian-ubuntu/>
<https://linuxize.com/post/how-to-list-installed-packages-on-debian/>
search "Debian query installed packages" online.
<https://itsfoss.com/list-installed-packages-ubuntu/>
apt list --installed | grep program_name
sudo apt list --installed | grep apache
sudo apt list --installed | grep libapache
dpkg-query -l | grep libapache
dpkg-query -l | grep apache (returns nothing)
<https://wiki.debian.org/ListInstalledPackages>
search "Debian query installable packages" online.
<https://phoenixnap.com/kb/dpkg-command>
https://linuxhint.com/debian_package_search_beginner_advanced/
apt-cache search libapache
apt-cache search libapache2-mod-substitute (returns nothing)
apt-cache search substitute (No Apache substitute module)
https://docs.oracle.com/cd/E24001_01/webcenter.1111/e12405/wcadm_rest.htm
Note that Apache versions 2.2.7 and later include mod_substitute by default.
search "how to check Apache server version" online.
httpd -v
Server version: Apache/2.4.53 (Unix)
Server built: Mar 14 2022 13:20:59
search "what is Apache so_module?" online.
<https://httpd.apache.org/docs/2.4/mod/>
https://httpd.apache.org/docs/2.4/mod/mod_so.html
search "How to install Apache server module" online.
<https://www.simplified.guide/apache/install-module>
sudo apt search libapache2-mod- #Ubuntu and Debian
<https://www.digitalocean.com/community/tutorials/how-to-install-configure-and-use-modules-in-the-apache-web-server>
<https://httpd.apache.org/docs/2.4/install.html>
(something is not right. httpd -M doesn't show PHP module. I think Apache PHP module must be there, since the Apache server runs WordPress.)

search "Ubuntu list installed Apache server modules" online. test this on my local Ubuntu Linux.

<https://www.liquidweb.com/kb/how-to-list-which-apache-modules-are-enabled-on-ubuntu/>
apache2ctl -M

apache2ctl -M | sort (no substitute module. maybe it's built in, and never listed?)

apache2ctl -v

Server version: Apache/2.4.52 (Ubuntu)

Server built: 2022-09-30T04:09:50

Search "Bitnami Apache server module directory" online.

<https://docs.bitnami.com/aws/faq/get-started/understand-directory-structure/>

/opt/bitnami directory

apache2/

The configuration file for the Apache Web server is stored in the /opt/bitnami/apache2/conf/ directory.

Good. The /opt/bitnami/apache2/modules directory contains mod_substitute.so. I'm in business, baby! I can use the mod_substitute httpd.conf settings in

https://docs.oracle.com/cd/E24001_01/webcenter.1111/e12405/wcadm_rest.htm!

Using the following httpd.conf settings.

```
ProxyRequests          Off
LoadModule              substitute_module
modules/mod_substitute.so
SetOutputFilter          SUBSTITUTE
```

SSLProxyEngine on

```
# allenyong.dev/demo-stockmarket/ proxy settings
ProxyPass               /demo-stockmarket/
https://allenyong.dev:8889/
ProxyPassReverse        /demo-stockmarket/
https://allenyong.dev:8889/
Substitute
s|https://allenyong.dev:8889/|https://allenyong.dev/demo-
stockmarket/|n
```

sudo /opt/bitnami/ctlscript.sh stop

sudo nano /opt/bitnami/apache2/conf/httpd.conf

sudo /opt/bitnami/ctlscript.sh start

sudo /opt/bitnami/ctlscript.sh restart

```
ProxyPass               /stockmarket
https://allenyong.dev:8889
ProxyPassReverse        /stockmarket
https://allenyong.dev:8889
```

~~Substitute~~
~~s|https://allenyoun~~~~g.dev:8889|https://allenyoun~~~~g.dev/stockmarket|~~
~~#~~

(don't use the above. the PayPal checkout page missing PayPal buttons. create allenyoun.g.dev/stockmarket page in WordPress, and redirect to allenyoun.g.dev/stockmarket/.) (the above doesn't successfully redirect. visiting https://allenyoun.g.dev/stockmarket/ seems to work. it seems like / at the end must be there for the substitution to take place. useallenyoun.g.dev/stockmarket/.)

2.2.1.5.1. Decision to not use Apache web server proxy

[5/26/2023 12:26 PM CST]

decide whether to use Apache web server proxy at all or not. I don't see a need. allenyoun.g.dev/stockmarket (likely this for ease of adding the redirect page) and/or stockmarket.allenyoun.g.dev can redirect to https://allenyoun.g.dev:8889 (the stockmarket frontend Node.js web server). Multi-domain name support settings in Apache server can be done later. I'm definitely leaning toward not using the Apache web server proxy.

decide and document whether to use allenyoun.g.dev/stockmarket or allenyoun.g.dev/demo-stockmarket. I'm leaning toward using allenyoun.g.dev/stockmarket; shorter is always better.

[5/26/2023 12:04 PM CST] FINAL DECISION [5/26/2023 12:15 PM CST] use allenyoun.g.dev/stockmarket.

2.2.1.5.2. Ports and URL used

[5/26/2023 12:22 PM CST]

allenyoun.g.dev:8889 for the frontend webserver.

allenyoun.g.dev:8890 for the backend webserver.

allenyoun.g.dev/Stockmarket WordPress page redirects to allenyoun.g.dev:8889.

2.2.1.6. Is - allowed in URL

[5/26/2023 11:37 AM CST]

Search "Is - allowed in URL" online.

<https://stackoverflow.com/questions/1856785/characters-allowed-in-a-url>

2.2.1.7. Does SSL certificate cover all the port numbers?

[5/26/2023 11:46 AM CST]

Search "Does SSL certificate cover all the port numbers?" online.

<https://security.stackexchange.com/questions/62900/do-ssl-x-509-certs-cover-all-ports-for-a-host>

2.3. How to set up Jenkins on Ubuntu (22.04.2 LTS) Linux for Allen Young's Stockmarket deployment automation

[5/25/2023 8:58 AM CST]

The process to use for installing and configuring local installation of Jenkins (not in Docker container) on Ubuntu (22.04.2 LTS) Linux for Node.js web app CI/CD automation.

(I'll later implement headless ChromeDriver usage in Jenkins pipeline for GUI-less Selenium end-to-end web app testing automation. The below instruction is for Selenium end-to-end web app testing automation with Google Chrome GUI. [5/25/2023 10:26 AM CST])

1. Log into your Ubuntu Linux computer in GUI as the Linux user that you use for software development purposes.
2. If not haven't done so already, install Google Chrome web browser on your Linux computer according to the instructions at <https://support.google.com/chrome/a/answer/9025903?hl=en>. The Google Chrome Linux download is available at <https://www.google.com/chrome/?platform=linux>.
3. Install Jenkins according to the instruction at <https://www.jenkins.io/doc/book/installing/>.
4. For enabling Node.js, npm, and Selenium packages installation for executing in Jenkinsfile pipeline, grant the sudo access to the jenkins user, by adding the jenkins user to the sudo group by executing `sudo usermod -aG sudo jenkins` in a terminal.
5. In a terminal, execute `xhost +SI:localuser:jenkins` to enable the jenkins user to use GUI (X Server) for launching Google Chrome GUI for Selenium end-to-end testing automation in Google Chrome GUI (not headless or GUI-less). Execute `xhost` to verify that the jenkins user has been added to the xhost authorized clients list. (Note that the X server client authorization granted by `xhost +SI:localuser:jenkins` is temporary; after the reboot, the permission will no longer be present.)
6. In a terminal, execute `echo 'xhost +SI:localuser:jenkins' >> ~/.profile` to auto-execute `xhost +SI:localuser:jenkins` after reboot for using Selenium testing automation in Jenkins pipeline. (In a later update, I'll look into whether using X authority file and xauto can add the jenkins user to the X-server authorized client list. Relevant references are at <https://manpages.ubuntu.com/manpages/jammy/man1/xauth.1.html> and <https://manpages.ubuntu.com/manpages/jammy/man7/Xsecurity.7.html>, and <https://manpages.ubuntu.com/manpages/jammy/man1/Xserver.1.html> [especially the GRANTING ACCESS section].)
7. In the terminal, execute `echo $DISPLAY` and `set | grep DISPLAY` to check the display that is used for Google Chrome GUI. The command returns something like `DISPLAY=:1`, which will be used later for setting the DISPLAY environment variable for the jenkins user for Google Chrome GUI Selenium automation by the jenkins user.
8. In the terminal, execute `sudo passwd jenkins` and assign a password to the jenkins user so that the password can be used during package installations using sudo. Write down the password somewhere safe.

9. In the terminal, execute `su - jenkins` to switch to the jenkins user.
10. In the terminal, execute `sudo apt update` to prepare for package installations.
11. Install Node.js and npm for the jenkins user.
 - 11.1. In the terminal, execute `sudo apt install nodejs` to install Node.js and npm.
 - 11.2. In the terminal, execute `node --version` to verify the Node.js installation.
 - 11.3. In the terminal, execute `npm --version` to verify the npm installation.
12. Install Node.js http-server for testing.
 - 12.1. In the terminal, execute `sudo npm install http-server -g`.
 - 12.2. In the terminal, execute `http-server --version` to verify the Node.js http-server installation.
13. Install Selenium for the jenkins user.
 - 13.1. In the terminal, execute `google-chrome --version` to verify that Google Chrome is available to the jenkins user and check the installed version.
 - 13.2. Visit <https://chromedriver.chromium.org/downloads>, and click the Google Chrome driver download link that matches your installed Google Chrome version. Make note of the `chromedriver_linux64.zip` file URL that will be downloaded and installed.
 - 13.3. In the terminal, execute `wget URL` to download the Google Chrome driver, using the `chromedriver_linux64.zip` file URL in place of URL.
 - 13.4. In the terminal, execute `unzip chromedriver_linux64.zip && sudo ln -s $PWD/chromedriver /usr/local/bin/chromedriver` to unzip and install the downloaded Google Chrome driver.
 - 13.5. If not installed Python and pip on your Linux computer already, install Python, and execute `sudo apt install python3-pip` to install pip.
 - 13.6. In the terminal, execute `python3 --version` to verify the Python installation.
 - 13.7. In the terminal, execute `pip3 --version` to verify the pip installation.
 - 13.8. In the terminal, execute `sudo pip3 install selenium` to install Selenium for Python.
 - 13.9. In the terminal, execute `google-chrome` to launch Google Chrome GUI. If an error that says "Missing X server or \$DISPLAY" is displayed, execute `export DISPLAY=display_value` to set the DISPLAY environment variable for the Jenkins user; `display_value` should be what you've gotten earlier using `echo $DISPLAY`; execute `echo $DISPLAY` in the terminal as the jenkins user to check that the DISPLAY environment variable is set for the jenkins user; execute `google-chrome` to verify that Google Chrome GUI can be launched as the jenkins user.
 - 13.10. Verify that the Selenium installation is working properly.
 - 13.10.1. In the terminal, execute `python3` to launch the Python Shell or REPL (Read, Evaluate, Print, Loop).
 - 13.10.2. In the Python Shell, execute `from selenium import webdriver`.
 - 13.10.3. In the Python Shell, execute `browser = webdriver.Chrome()` to launch a GUI Google Chrome instance.
 - 13.10.4. In the Python Shell, execute `browser.quit()` to close the GUI Google Chrome instance.
 - 13.10.5. In the Python Shell, execute `exit()` to exit the Python Shell.

14. Install other Python dependencies required for the Selenium end-to-end testing.
 - 14.1. In the terminal, execute `sudo pip3 install psutil` to install Selenium for Python.
 - 14.2. In the terminal, execute `sudo pip3 install requests` to install Selenium for Python.
15. Enable opening port 80 as non-root user for Jenkins pipeline Selenium end-to-end testing (for running a local frontend server for testing). (Reference: <https://stackoverflow.com/questions/60372618/nodejs-listen-eaccess-permission-denied-0-0-0-080>)
 - 15.1. In the terminal, execute `sudo apt install libcap2-bin`.
 - 15.2. In the terminal, execute `sudo setcap cap_net_bind_service=+ep `readlink -f `which node```.
16. In the terminal, execute `sudo apt install mkcert` to enable HTTPS on the backend server.
17. Add Newman to enable automating Postman collection test in Jenkins pipeline.
 - 17.1. In the terminal, execute `sudo npm install -g newman` to enable HTTPS on the backend server.
 - 17.2. (Optional if using Newman in Jenkins freestyle projects) Add Newman to Jenkins according to the instruction in <https://learning.postman.com/docs/collections/using-newman-cli/integration-with-jenkins/>, to enable automating Postman collection test in Jenkins freestyle projects.
18. In the terminal, execute `exit` to switch back to the original user.
19. Configure Jenkins to add the needed environment variables, credentials, and parameters.
 - 19.1. Visit the Jenkins homepage in a web browser, and log in.
 - 19.2. Add the Jenkins environment variables under "Dashboard -> Manage Jenkins -> System -> Global properties -> Environment variables".
 - 19.2.1. Add the DISPLAY environment variable.
 - 19.2.2. Add the DEBUG environment variable set to *. (For Node.js app testing.)
 - 19.3. Add the Jenkins global credentials.
 - 19.3.1. ...
 - 19.4. Add the Jenkins pipeline parameters.
 - 19.4.1. ...

2.3.1. NOTES

Install node, npm, and Selenium on the Jenkins server.

```
sudo apt update
sudo apt install nodejs npm
```

Install NodeJS plugin on Jenkins
(this plugin doesn't work in the latest Jenkins.
in <http://192.168.1.104:8080/manage/configure>,
try adding Environment variables, PATH with Jenkins server PATH environment variable value)

Try adding the jenkins user to the logged in user group,
try adding Environment variables, PATH with Jenkins server PATH environment variable value)
for npm execution allowance to the jenkins user.

<https://stackoverflow.com/questions/45914038/jenkins-unable-to-find-npm>
<https://plugins.jenkins.io/nodejs/>

UPDATE

<http://192.168.1.104:8080/manage/configureTools/>
NodeJS plugin configuration is in the above URL.
Dashboard -> Manage Jenkins -> Tools

<https://plugins.jenkins.io/nodejs/> is only available in
Jenkins freestyle project configuration, not in pipeline.

Try the script in <https://www.jhipster.tech/setting-up-ci-linux/>.
Update the PATH in Jenkins afterward.
actually, log in as jenkins user, then install node and npm.

UPDATE

<https://stackoverflow.com/questions/18068358/cant-su-to-user-jenkins-after-installing-jenkins>
sudo su -s /bin/bash jenkins

sudo apt update
sudo apt install nodejs npm

sudo su - jenkins (this bypasses asking for password in switch the user.)

<https://www.baeldung.com/linux/login-as-user-jenkins>
sudo passwd jenkins

su - jenkins (enter new password)
sudo apt update
sudo apt install nodejs npm
jenkins is not in the sudoers file. This incident will be reported.

no need to do the following then.
sudo passwd jenkins
do the following.
sudo su - jenkins (this bypasses asking for password in switch the user.)
then manually install node.js and npm as instructed in <https://www.jhipster.tech/setting-up-ci-linux/>.

export NODE_VERSION=18.16.0
wget [http://nodejs.org/dist/v\\$NODE_VERSION/node-v18.16.0.tar.gz](http://nodejs.org/dist/v$NODE_VERSION/node-v18.16.0.tar.gz)

TO TRY

Before installing Jenkins on Linux, create a user named jenkins. See if that works.
(is this a good approach? I feel that it probably isn't.)

For now, I feel that creating a Jenkins Docker image and using Docker container on local Linux development computer is the best approach.

try enabling running sudo with jenkins account. search the error online. for running Selenium, I need to be able to install Selenium to the jenkins account. [5/24/2023 8:57 PM CST]

For Selenium end-to-end testing automation, the Jenkins server must have GUI. So, it might have to be a VMI (Virtual Machine Image), not Docker image.

Search "Can Docker container Linux have GUI?" online.

<https://medium.com/nerd-for-tech/running-gui-based-applications-inside-a-docker-container-645399ca2ef0>

<https://www.cloudthat.com/resources/blog/run-gui-application-inside-docker-container>

<https://medium.com/geekculture/run-a-gui-software-inside-a-docker-container-dce61771f9>

<https://www.geeksforgeeks.org/running-gui-applications-on-docker-in-linux/>

<https://www.howtogeek.com/devops/how-to-run-gui-applications-in-a-docker-container/>

<https://plugins.jenkins.io/sshd/>

<https://www.jenkins.io/doc/book/managing/cli/>

If and as needed, search "how to execute ssh command in Jenkinsfile" online.

If and as needed, search "how to execute scp command in Jenkinsfile" online.

Actually, the jenkins user has ssh, scp, and sshpass by default.

No need to install them separately, unlike node.

I'm beginning to think that maybe using Jenkinsfile for pipeline is a wrong approach, since it's such a hassle to get Node.JS and npm work in Jenkinsfile.

Maybe the freestyle project with a Node.js script should be used?

I'll think about it.

Installing Node.js according to <https://www.jhipster.tech/setting-up-ci-linux/> might be the best option yet. I'll think about it.

sudo apt update

jenkins is not in the sudoers file. This incident will be reported.

Search "Ubuntu jenkins is not in the sudoers file. This incident will be reported." online.

<https://medium.com/fusionqa/how-to-fix-username-is-not-in-the-sudoers-file-this-incident-will-be-reported-error-42a13a11203f>

```
sudo chmod u+w,g+w,o+w /etc/sudoers
echo 'jenkins ALL=(ALL) ALL' >> /etc/sudoers
sudo chmod u-w,g-w,o-w /etc/sudoers
```

search "how to add write permission to all" online.
<https://www.pluralsight.com/blog/it-ops/linux-file-permissions>
<https://superuser.com/questions/126073/chmod-to-allow-read-and-write-permissions-for-directory>

search "chmod: changing permissions of '/etc/sudoers': Operation not permitted" online.
<https://superuser.com/questions/306182/chmod-sudoers-without-using-sudo>

fix sudo problem on the Linux computer, according to the instruction in
<https://superuser.com/questions/306182/chmod-sudoers-without-using-sudo>.
Search "Ubuntu how to escape to the bootloader menu" online.
<https://askubuntu.com/questions/16042/how-to-get-to-the-grub-menu-at-boot-time>
Menu will appear if you press and hold Shift during loading Grub, if you boot using BIOS.
When your system boots using UEFI, press Esc.

set the root user password, and record it in "OS logon passwords.doc".

try enabling running sudo with jenkins account. search the error online. for running Selenium, I need to be able to install Selenium to the jenkins account. [5/24/2023 8:57 PM CST]

<https://www.jhipster.tech/setting-up-ci-linux/>
The Node.js executables installation directory
`/var/lib/jenkins/node-v18.16.0/bin`

Enable root password on the local Ubuntu Linux development computer. (don't do this. no need for this. instead, use the command[s] in <https://linuxize.com/post/how-to-enable-and-disable-root-user-account-in-ubuntu/>.)

2.3.2. Deprecated

[5/25/2023 10:50 AM CST]

In the terminal, execute `google-chrome` to launch Google Chrome GUI. If an error that says "Missing X server or \$DISPLAY" is displayed, execute `export DISPLAY=display_value` to set the DISPLAY environment variable for the Jenkins user; `display_value` should be what you've gotten earlier using `echo $DISPLAY`; execute `echo $DISPLAY` in the terminal as the jenkins user to check that the DISPLAY environment variable is set for the jenkins user; execute `google-chrome` to verify that Google Chrome GUI can be launched as the jenkins user.

19.5. Install and execute Xvfb for running Google Chrome GUI.

19.5.1. In the terminal, execute `sudo apt install -y xvfb` to install Xvfb.

19.5.2. In the terminal, execute `Xvfb -ac :99 -screen 0 1920x1080x16` & to run Xvfb.

19.5.3. In the terminal, execute `export DISPLAY=:99` to set the DISPLAY environment variable needed by Google Chrome.

19.5.4. In the terminal, execute `google-chrome` to verify that Google Chrome GUI can be launched as the jenkins user.

Install mkcert according to the instruction at <https://github.com/FiloSottile/mkcert#installation> to enable HTTPS on the backend server. For installing homebrew on Linux, refer to <https://docs.brew.sh/Homebrew-on-Linux> and <https://brew.sh/>.

2.3.3. References

2.3.3.1. How to check Ubuntu version

[5/25/2023 9:00 AM CST]

Search "How to check Ubuntu version" online.

<https://www.ionos.com/digitalguide/server/know-how/check-ubuntu-version/>
`lsb_release -a`

2.3.3.2. How to enable root password on Ubuntu

[5/25/2023 9:03 AM CST]

Search "How to enable root password on Ubuntu" online.

<https://linuxize.com/post/how-to-enable-and-disable-root-user-account-in-ubuntu/>

This is not required. It is noted here as a reference.

2.3.3.3. How to boot into root user shell recovery mode in Ubuntu 22.04.2 LTS

[5/25/2023 9:06 AM CST]

For a single-boot Ubuntu installation with no GRUB menu display during early booting.

1. During the early booting process, repeatedly press Esc, to make the GRUB menu appear.
2. In the GRUB menu, select 'Advanced options for Ubuntu'.
3. Select 'Ubuntu, with Linux 5.19.0-42-generic (recovery mode)'.
4. Select 'root Drop to root shell prompt'.
5. Press the enter key as instructed.

2.3.3.4. How to install Selenium on Linux Ubuntu

[5/25/2023 9:28 AM CST]

Search "How to install Selenium on Linux Ubuntu" online.

<https://askubuntu.com/questions/937770/how-can-i-install-and-set-up-selenium-webdriver-on-ubuntu-16-04>

<https://tecadmin.net/setup-selenium-chromedriver-on-ubuntu/> (some instructions on using headless ChromeDriver? for automated testing.)
<https://chromedriver.chromium.org/downloads>

Search "What is Headless ChromeDriver?" online.
<https://developer.chrome.com/articles/new-headless/>

2.3.3.5. Linux apt vs. apt-get

[5/25/2023 10:02 AM CST]

Search "Linux apt vs. apt-get" online.
<https://itsfoss.com/apt-vs-apt-get-difference/>

In summary, apt is newer on Ubuntu.

2.3.3.6. Linux how to start Google Chrome in terminal

[5/25/2023 10:14 AM CST]

Search "Linux how to start Google Chrome in terminal" online.
<https://smallbusiness.chron.com/run-google-chrome-terminal-52497.html>
<https://stackoverflow.com/questions/34796888/how-to-open-google-chrome-from-terminal>

2.3.3.7. Linux Google Chrome "Missing X server or \$DISPLAY" error

[5/25/2023 10:19 AM CST]

Search "Linux Google Chrome Missing X server or \$DISPLAY error" online.
<https://github.com/puppeteer/puppeteer/issues/8148>
<https://stackoverflow.com/questions/60304251/unable-to-open-x-display-when-trying-to-run-google-chrome-on-centos-rhel-7-5>
<https://lightrun.com/solutions/puppeteer-puppeteer-bug-missing-x-server-or-display/>
<https://www.baeldung.com/linux/no-x11-display-error>

Search "Linux Ubuntu jenkins user Google Chrome Missing X server or \$DISPLAY" online.
<https://sqa.stackexchange.com/questions/45539/xvfb-and-selenium-on-ec2-unable-to-view-chrome-ui-on-vnc-viewer>
<https://community.jenkins.io/t/new-jenkins-installation-need-to-run-existing-python-selenium-scripts/4891>
<https://askubuntu.com/questions/47642/how-to-start-a-gui-software-on-a-remote-linux-pc-via-ssh>

2.3.3.8. Linux how to run Google Chrome as jenkins user

[5/25/2023 11:06 AM CST]

Search "<https://stackoverflow.com/questions/48278408/how-to-launch-chrome-browser-from-jenkins-directly-instead-of-using-code-in-ecli>" online.

2.3.3.9. Linux Ubuntu how to enable a user to use display

[5/25/2023 11:22 AM CST]

Search "Linux Ubuntu how to enable a user to use display" online.

<https://askubuntu.com/questions/663233/allow-display-access-for-a-user-not-logged-on-gui>

<https://help.ubuntu.com/stable/ubuntu-help/prefs-display.html.en>

<https://unix.stackexchange.com/questions/654125/set-display-settings-for-another-user-in-ubuntu>

<https://superuser.com/questions/85818/allow-only-one-extra-user-to-access-the-display>

Execute `xhost` in a terminal with the GUI logged in user.

Search "Linux Ubuntu how to add authorized client using xhost" online.

<https://unix.stackexchange.com/questions/593411/how-to-add-authorized-client-to-xhost>

<https://manpages.ubuntu.com/manpages/trusty/man7/Xsecurity.7.html>

```
xhost +SI:localuser:jenkins
```

Search "What is xhost in linux" online.

Description. The `xhost` command adds or deletes host names on the list of machines from which the X Server accepts connections. This command must be run from the machine with the display connection. You can remove a name from the access list by using the `-Host` parameter.

`xhost` Command - IBM

<https://www.ibm.com/docs/en/aix/7.1?topic=x-xhost-command>

Search "Linux Ubuntu make xhost client authorization permanent" online.

<https://unix.stackexchange.com/questions/593411/how-to-add-authorized-client-to-xhost>

<https://askubuntu.com/questions/877820/what-are-xhost-and-xhost-si>

<https://manpages.ubuntu.com/manpages/xenial/man1/xhost.1.html>

<https://manpages.ubuntu.com/manpages/trusty/man7/Xsecurity.7.html>

<https://www.linuxquestions.org/questions/linux-security-4/how-to-disable-x-security-permanently-165282/>

put the command `xhost +` in your startup file (e.g. `~/.profile` or `~/.bash_profile`).

Search "Linux Ubuntu how to add user to Xauthority file" online.

<https://manpages.ubuntu.com/manpages/xenial/man7/Xsecurity.7.html>

<https://serverfault.com/questions/51005/how-to-use-xauth-to-run-graphical-application-via-other-user-on-linux>

<https://docs.citrix.com/en-us/linux-virtual-delivery-agent/7-15-ltsr/configuration/configure-xauthority.html>

<https://manpages.ubuntu.com/manpages/jammy/man1/xauth.1.html>

<https://askubuntu.com/questions/960793/whats-the-right-place-to-set-the-xauthority-environment-variable>

`xauth` list

2.3.3.10. Linux Ubuntu how to install X server for jenkins user

[5/25/2023 10:56 AM CST]

~~Search "Linux Ubuntu how to install X server for jenkins user" online.~~

(I don't feel that this should be needed. The Linux computer already has an X server since the GUI runs. [5/25/2023 11:06 AM CST])

2.3.3.11. Linux Ubuntu how to check terminal shell type

[5/25/2023 12:04 PM CST]

Search "Linux Ubuntu how to check terminal shell type" online.

<https://linuxhandbook.com/shell-using/>

You can use the special shell parameter `$$`. (doesn't work on my Ubuntu. it says 'command not found'.)

<https://askubuntu.com/questions/590899/how-do-i-check-which-shell-i-am-using>

`echo $0` (this works on my Ubuntu)

2.3.3.12. What is .profile for

[5/25/2023 12:36 PM CST]

Search "what is .profile for" online.

<https://www.ibm.com/docs/en/aix/7.2?topic=files-profile-file>

<https://www.quora.com/What-is-profile-file-in-Linux>

2.3.3.13. When does .profile get executed?

[5/25/2023 12:36 PM CST]

Search "When does .profile get executed?" online.

<https://unix.stackexchange.com/questions/348156/when-is-profile-executed>

<https://superuser.com/questions/278433/whats-the-difference-between-profile-and-bash-profile-and-when-do-you-config>

<https://www.unix.com/shell-programming-and-scripting/269582-whats-profile-file-auto-executed-when-i-login.html>

2.4. How to auto-deploy Allen Young's Stockmarket to an AWS LightSail multisite WordPress instance using Jenkins

[5/27/2023 4:54 PM CST]

(This will be completed later when I get back my Linux development computer on 6/12/2023.)

2.5. How to manually deploy Allen Young's Stockmarket to an AWS LightSail multisite WordPress instance

[5/27/2023 4:54 PM CST]

1. In a terminal on Linux (or Git Bash on Windows), execute the ssh command to log into the AWS LightSail multisite WordPress instance (e.g. `ssh -i "C:\Users\Allen\Documents\Asian-American Man\AWS\LightsailDefaultKey-us-east-1.pem" bitnami@54.88.122.32`).
2. In the SSH terminal, kill, if any, all the running Node.js servers that are using the ports that your Allen Young's Stockmarket will use.
 - 2.1. In the SSH terminal, execute `killall -9 node` or `kill -9 (pid)` to kill all the running backend Node.js servers, or to kill the running backend Node.js server using the port that your Allen Young's Stockmarket backend Node.js server will use. In Ubuntu (22.04.2 LTS) Linux, `sudo netstat -tulpn | grep LISTEN` command can be used to list the used ports with the PID of the programs using the ports.
 - 2.2. In the SSH terminal, execute `killall -9 http-server` or `kill -9 (pid)` to kill all the running frontend Node.js servers, or to kill the running frontend Node.js server using the port that your Allen Young's Stockmarket frontend Node.js server will use. In Ubuntu (22.04.2 LTS) Linux, `sudo netstat -tulpn | grep LISTEN` command can be used to list the used ports with the PID of the programs using the ports.
3. In the SSH terminal, in the home directory, execute `rm -r -f ~/stockmarket`.
4. In the SSH terminal, in the home directory, execute `git clone https://github.com/AllenYoungDev/stockmarket.git`.
5. Create the backend single-source-of-truth settings files.
 - 5.1. Create `.env` file in the Express.js directory by copying `_example.env` file as `.env` file, then editing the `.env` file content.
 - 5.1.1. In the SSH terminal, execute `cp ~/stockmarket/backend/Express.js/_example.env ~/stockmarket/backend/Express.js/.env`.
 - 5.1.2. Edit the copied `.env` file content.
 - 5.1.2.1. In the SSH terminal, execute `nano ~/stockmarket/backend/Express.js/.env`.
 - 5.1.2.2. In nano editor, set each setting to your preference. Make sure that both the frontend and backend website URLs start with `https`, not `http`.
 - 5.2. Create `aws_config.json` file in the Express.js directory by copying `_example.aws_config.json` file as `aws_config.json` file, then editing the `aws_config.json` file content.
 - 5.2.1. In the SSH terminal, execute `cp ~/stockmarket/backend/Express.js/_example.aws_config.json ~/stockmarket/backend/Express.js/aws_config.json`.
 - 5.2.2. Edit the copied `.env` file content.
 - 5.2.2.1. In the SSH terminal, execute `nano ~/stockmarket/backend/Express.js/aws_config.json`.
 - 5.2.2.2. In nano editor, set each setting to your preference.
 - 5.3. Edit the `settings.js` file content.
 - 5.3.1. In the SSH terminal, execute `nano ~/stockmarket/backend/Express.js/public/settings.js`.
 - 5.3.2. In nano editor, set each setting to your deployed server setting.
6. Create the frontend single-source-of-truth settings files.

- 6.1. Edit the ServerUrls.js file content.
 - 6.1.1. In the SSH terminal, execute `nano ~/stockmarket/frontend/React/settings/ServerUrls.js`.
 - 6.1.2. In nano editor, set each setting to your deployed server setting.
7. Batch replace the development React library URLs to the production React library URLs, in the React app HTML files.
 - 7.1. In the SSH terminal, execute `sed 's+https://unpkg.com/react@18/umd/react.development.js+https://unpkg.com/react@18/umd/react.production.min.js+g' ~/stockmarket/frontend/React/*.html`.
 - 7.2. In the SSH terminal, execute `sed 's+https://unpkg.com/react-dom@18/umd/react-dom.development.js+https://unpkg.com/react-dom@18/umd/react-dom.production.min.js+g' ~/stockmarket/frontend/React/*.html`.
8. Start the Allen Young's Stockmarket backend Node.js server.
 - 8.1. In the SSH terminal, execute `cd ~/stockmarket/backend/Express.js`.
 - 8.2. In the SSH terminal, execute `npm install`.
 - 8.3. In the SSH terminal, execute `node server.js true true > /dev/null 2>&1 &` to start the backend Node.js server as a separate process.
9. Start the Allen Young's Stockmarket frontend Node.js server.
 - 9.1. In the SSH terminal, execute `cd ~/stockmarket/frontend/React`.
 - 9.2. In the SSH terminal, execute `http-server -p (frontend server port number, e.g. 8889) -S -C (SSL cert file full path, e.g. /opt/bitnami/letsencrypt/certificates/allenyoung.dev.crt) -K (SSL key file full path, e.g. /opt/bitnami/letsencrypt/certificates/allenyoung.dev.key) > /dev/null 2>&1 &` to start the frontend Node.js server as a separate process.
10. Update the admin user entry in the database.
 - 10.1. In a web browser, visit the deployed Allen Young's stockmarket website.
 - 10.2. Log in as the default admin account with the email address 'admin@allenyoung.dev' and password '12345abcde!'.
 - 10.3. Visit the account settings page.
 - 10.4. Change the admin info to your admin info, including the password.

2.5.1. References

[5/27/2023 5:58 PM CST]

2.5.1.1. Linux file copy command

[5/27/2023 5:58 PM CST]

Search "Linux file copy command" online.

2.5.1.2. Linux command for replacing string in multiple files

[5/27/2023 7:07 PM CST]

Search "Linux command for replacing string in multiple files" online.

<https://stackoverflow.com/questions/11392478/how-to-replace-a-string-in-multiple-files-in-linux-command-line>

<https://www.cyberciti.biz/faq/how-to-use-sed-to-find-and-replace-text-in-files-in-linux-unix-shell/>

<https://www.redhat.com/sysadmin/edit-text-bash-command>

<https://www.internalpointers.com/post/linux-find-and-replace-text-multiple-files>

```
cd /path/to/your/folder
```

```
sed -i 's/foo/bar/g' *
```

```
sed -i 's/user_/admin_/g' *.yaml
```

Sed command allows you to change the delimiter / to something else.

```
sed 's+http://+https://www.cyberciti.biz+g' input.txt
```

```
sed 's+http://+https://www.cyberciti.biz+g' *.html
```

Try the following.

```
sed 's+https://unpkg.com/react@18/umd/react.development.js+https://unpkg.com/react@18/umd/react.production.min.js+g' ~/stockmarket/frontend/React/*.html
```

```
sed 's+https://unpkg.com/react-dom@18/umd/react-dom.development.js+https://unpkg.com/react-dom@18/umd/react-dom.production.min.js+g' ~/stockmarket/frontend/React/*.html
```

2.5.1.3. React production library URLs

[5/27/2023 8:00 PM CST]

<https://legacy.reactjs.org/docs/add-react-to-a-website.html>

```
<!-- Note: when deploying, replace "development.js" with "production.min.js". -->
<script src="https://unpkg.com/react@18/umd/react.development.js" crossorigin></script>
<script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"
crossorigin></script>
```

```
https://unpkg.com/react@18/umd/react.production.min.js
```

```
https://unpkg.com/react-dom@18/umd/react-dom.production.min.js
```

2.5.1.4. Bash shell command string substitution

[5/28/2023 1:21 PM CST]

```
cp
```

```
"${HOME}/stockmarket/backend/Express.js/_example.aws_config.json"
```

```
"${HOME}/stockmarket/backend/Express.js/aws_config.json" is equivalent
```

```
to cp ~/stockmarket/backend/Express.js/_example.aws_config.json  
~/stockmarket/backend/Express.js/aws_config.json.
```

2.5.1.5. Linux direct command output to null

[5/28/2023 1:35 PM CST]

Search "Linux direct command output to null" online.

<https://www.cyberciti.biz/faq/how-to-redirect-output-and-errors-to-devnull/>

<https://unix.stackexchange.com/questions/119648/redirecting-to-dev-null>

2.5.1.6. Linux command on a separate process with output redirect

[5/28/2023 1:46 PM CST]

Search "Linux command on a separate process with output redirect" online.

<https://unix.stackexchange.com/questions/74520/can-i-redirect-output-to-a-log-file-and-background-a-process-at-the-same-time>

3. How to deploy Allen Young's Stockmarket to a running AWS Fargate instance

[5/27/2023 4:26 PM CST]

(SSH identity-file usage will be implemented later. [5/27/2023 4:37 PM CST])

3.1. How to create a Docker image for deploying Allen Young's Stockmarket to a running AWS Fargate instance

[5/25/2023 8:58 AM CST]

3.1.1. Dockerfile content

3.2. How to deploy a Docker image as a new AWS Fargate instance for deploying Allen Young's Stockmarket to a running AWS Fargate instance (word this better, something more specific than just 'a Docker image', something tied to Allen Young's Stockmarket)

[5/27/2023 4:41 PM CST]

3.3. How to deploy Allen Young's Stockmarket to a running AWS Fargate instance

[5/27/2023 4:42 PM CST]

4. How to deploy Allen Young's Stockmarket as an AWS Fargate blue/green deployment with Docker image auto-building by Jenkins

[5/27/2023 4:35 PM CST]

(This will be completed later)

5. How to deploy Allen Young's Stockmarket with load-balancing multiple frontend AWS Fargate instances, load-balancing multiple user-facing or frontend-facing backend AWS Fargate instances, and one database backend AWS Fargate instance

[5/27/2023 6:39 PM CST]

(This will be completed later)

6. How to deploy Allen Young's Stockmarket with load-balancing multiple frontend AWS Fargate instances, load-balancing multiple user-facing or frontend-facing backend AWS Fargate instances, and load-balancing multiple database backend AWS Fargate instances

[5/27/2023 6:39 PM CST]

(This will be completed later)

7. How to deploy Allen Young's Stockmarket with load-balancing and auto-scaling multiple frontend AWS Fargate instances, load-balancing and auto-scaling multiple user-facing or frontend-facing backend AWS Fargate instances, and load-balancing and auto-scaling multiple database backend AWS Fargate instances

[5/27/2023 6:39 PM CST]

(This will be completed later)

III. Web-App Frontend (UI) Design (with URIs for routing)

1. [3/9/2023 4:38 PM CST] [4/14/2023 1:43 PM CST](stock stats update)

1.1. Stockmarket home webpage

URI: /index.html

(Register|Login)|(My Account | Admin | Logout)
('Admin' is shown only when admin account is logged in)

Company Stockmarket

For investing in (xx)/(AI, robotics, human longevity biotech, and nuclear-fusion powered outer space tech R&D and commercialization)/(the future).

About (Company)/(Robocentric) (for the demo, links to AllenYoung.dev, not Robocentric.com/About)

Current Company valuation: US\$7 million

Number of authorized Company shares: 70,000

Number of issued Company shares: 24,500 (35% of total)

Number of outstanding Company shares: 24,500 (35% of total)

Number of Company shares available for purchase: 3,500 (5% of total) (auto-calculated using the values in stock stats table and stock reservation table. expired reservations in the stock reservation table is auto-removed while calculating this that's equal to or more than 6 minutes past the reservation starting time. all done by the backend route for retrieving stock stats.)~~(I don't think stock reservation table is needed; transactions table can probably do; check the~~

~~PayPal processing code to determine whether transactions table can provide the needed function-
if transactions table can do the job, no cron needed for checking and updating transactions table
entries, since the homepage visit will do the job. actually, stock reservation table is needed,
separate the stock reservation and payment processing in two separate backend routes.)~~

Price per share: US\$100

Cash value of the shares available for purchase: US\$350,000

[[Log in](#) or [register](#) to buy Company stocks]

(the above displayed, only if not logged in. when the above button is pressed, a floating center window for logging in.)

Number of Company shares I own: xx
(x% of the authorized Company shares)
(the above is displayed only if logged in)

(the below is grayed out and disabled if not logged in.)

Number of Company shares to buy: []

Price of Company shares to buy: US\$xx (auto-calculated)

[Proceed to checkout]

(when the above button is pressed, using the stock stats table and the stock reservation table, the number of available shares are calculated; if the number of available shares are smaller than the number of shares to buy, error message display with the latest number of available shares, "Number of Company shares available for purchase" update above, and staying on the page.)

FAQ (animate the FAQ section expansion and collapse using JQuery)

What's the current Company ownership structure?

Supervoting vs. non-supervoting Company stocks

Can I return my purchased Company share(s) and get a refund?

What the Company share liquidation plan?

Shares outstanding vs. Treasury stock vs. Issued shares vs. Authorized capital vs. Unissued shares

Company stock liquidation options (current and post-IPO)

Investor risk warning

Terms and conditions

Fees

About Company

(the above are all pop-up window links)

For inquiries and support requests, please send an email to support@allenyoung.dev.

(footer with relevant info and links)

Contact

1.1.1. References

https://en.wikipedia.org/wiki/Shares_outstanding

https://en.wikipedia.org/wiki/Issued_shares

https://en.wikipedia.org/wiki/Treasury_stock
https://en.wikipedia.org/wiki/Authorised_capital

1.2. Buy checkout webpage

Stock Purchase Checkout Webpage

[2/15/2023 10:24 PM CST]

URI: /Checkout (webpage backend generated at this route)

Company Stockmarket

My account

Checkout

Time remaining on Company shares reservation for purchase: (countdown starting at 5 min; when the time expires, automatically taken back to the home page and the stock reservation cancellation.)

Total number of Company shares to buy: xx

Total price of Company shares to buy: US\$xx (US\$xx per share plus x% processing fee)

After you make the payment, you'll be taken to a receipt page and a receipt will be email to you. The transaction record will be available in your account page.

(PayPal checkout form)

(successful checkout updates the stock stats table, and removes the entry in stock reservation table, in an atomic database transaction; if error, an email is auto-sent to the admin.)

(all the Robocentric stock purchase terms and conditions listed here.)

1.2.1. Purchase receipt page

URI: /Receipt (webpage backend generated at this route)

Company Stockmarket

My account

Purchase Receipt

(Date, time)

Current Company valuation: US\$4 million

Number of authorized Company shares: 40,000

Total number of Company shares bought: xx (x% of authorized shares)

Total price of Company shares bought: US\$xx

PayPal transaction ID: xx
Company transaction ID: xx
Payment type: xx
Payment source: xx
Name: xx
Email address: xx
Phone number: xx

Congratulations on purchasing Company share(s)!

A copy of this receipt has been emailed to you.

This transaction record is now available in your account page.

(all the Robocentric stock purchase terms and conditions listed here.)

1.2.2. Handling backward or forward web browser button to navigate back to the Checkout page

[5/1/2023 10:14 AM CST]

fully resolve the issue of going back to the checkout page from receipt page in web browser. [4/24/2023 11:09 AM CST] (NOTE [4/27/2023 4:41 PM CST] maybe /Checkout route method should redirect to the homepage if there is no reserved share[s] by the user. think about this. maybe this is the way to go.) (NOTE [4/29/2023 12:28 PM CST] think about what to do about this. because of the way PayPal does its online payment processing, no double purchase occurs when backed to the page and payment is submitted again. the web browser displays info using the old, cached info. using a (transaction) nonce in cookie might be an option; if nonce in cookie and in backend database match, redirect to the homepage, because it indicates that the payment is already processed and the cookie nonce is present due to the web browser using the cached data to display the webpage; set the nonce in cookie and backend database when the transaction successfully completes. I'm leaning toward potentially implementing this in the future; I don't feel that there's a compelling reason to do this now, given the time constraints I have. actually, set the nonce in cookie when the backend route method starts, then keep on executing instead of returning from the backend route method; store the nonce in backend database when the transaction successfully completes. I don't think you can send response twice in the same route method call; I couldn't before, so I'm sure you can't.) (UPDATE [4/29/2023 1:11 PM CST] So, what should be stored in the backend database is (the latest) clientToken returned by paypal.generateClientToken(). clientToken is provided as a parameter to checkout.ejs when rendering the webpage to return in response, as a <script> tag attribute, which seem to be directly accessible by the frontend JavaScript code. search "javascript how to access value in <script>" online. <https://stackoverflow.com/questions/17834293/javascript-how-to-get-value-from-a-script-tag>. so, the frontend JavaScript code in <head> element should use fetch() GET to check whether the client token in frontend HTML is in the backend database, and redirect to the homepage if it does. I think this is the way to go. it shouldn't be too hard or lengthy to implement. store the client token in the transactions table entry, when it is issued in the backend

route method. I'm leaning toward doing this now, not later. no reason to delay. search "javascript how to access HTML element attribute value" online. <https://www.javascripttutorial.net/dom/attributes/get-the-value-of-an-attribute/>) (FINAL DECISION [4/29/2023 1:48 PM CST] I'm implementing this now. no reason to delay.) (UPDATE [4/29/2023 1:57 PM CST] the transaction entry does not exist yet in /Checkout route method call, so it must be stored in the users table entry.) (UPDATE [4/29/2023 2:23 PM CST] hang on. I need something a bit more complex. when the checkout page loads, for the first time with a new client token, in the current design, the client token is already in the backend database in the users table, so the frontend webpage JavaScript code will redirect to the homepage. in order to resolve this issue, the transactions table entry must have the "Client token" column too that will be set only when the transaction is successful, and the frontend JavaScript function must call the backend API for checking client token in the transactions table, not users table. I'm doing this.) (UPDATE [4/29/2023 5:46 PM CST] shit! when backed up to the checkout webpage, the web browser uses cached fetch() value, hence 'no record' is used. search "JavaScript HTML force webpage refresh when backed up" online. <https://stackoverflow.com/questions/43043113/how-to-force-reloading-a-page-when-using-browser-back-button>. search "HTTP header no cache" online. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control>. no-cache doesn't work on Google Chrome. <https://developer.mozilla.org/en-US/docs/Web/API/PerformanceNavigationTiming>) (UPDATE [4/29/2023 6:15 PM CST] I'm not sure if using no-store for Cache-Control HTTP header is what I want for the Checkout backend page. I need to think about this, and then decide on this.) (UPDATE [4/29/2023 6:34 PM CST] for now, using no-store in Cache-Control HTTP header seems to be the best solution. all that circus of adding new table columns and backend API wasn't needed [after all]. think about what to do, then decide on what to do. maybe comment out the new table column usage function calls. because the client token value is the same from the web browser cache when backed to the checkout page, the web browser doesn't call fetch() again and uses the old fetch() value when backed up to the page. I'm leaning toward the rollback. database call is expensive, and doing the least of it is the best. I'll decide what to do about this tomorrow, not today; enough on this today already. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch#headers, https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API. <https://developer.mozilla.org/en-US/docs/Web/API/PerformanceNavigationTiming>, [PerformanceNavigationTiming.type](https://developer.mozilla.org/en-US/docs/Web/API/PerformanceNavigationTiming.type). search "how to access PerformanceNavigationTiming in JavaScript" online. <https://www.w3.org/TR/navigation-timing-2/>. search "PerformanceNavigationTiming examples" online. <https://stackoverflow.com/questions/53613071/determining-navigation-type-from-performancenavigationtiming> [this holds promise] [5/1/2023 9:56 AM CST][UPDATE I think I like this a lot!!!!!! a lot!!!!!!].) (OPTIONS [4/30/2023 2:54 PM EST] redirect to homepage using JavaScript. force reload using JavaScript. no-store in the cache control HTTP header. decide on which one to use, when I feel like deciding!) (TO DO [5/1/2023 9:58 AM CST] I like using the method [code] in <https://stackoverflow.com/questions/53613071/determining-navigation-type-from-performancenavigationtiming>, a lot! think and decide on whether to choose using this, and then document the decision, now, NEVER LATER! if using this, comment out or roll back using the extra table columns. I'm heavily leaning toward the roll back! Give myself at least five minutes of brain break and refresh before conclusively deciding on this matter, then keep moving forward! [5/1/2023 10:25 AM CST] I'm definitely heavily

leaning toward the roll back: no unnecessary table columns and backend code! [5/1/2023 10:29 AM CST] yeah, let's roll back: no reason not to! actually, no. why not just keep the client token in the transactions table entry? think about this. no, actually, roll back is probably the best. FINAL DECISION [5/1/2023 10:34 AM CST] Roll back: get rid of everything that is not needed and used! Implement the rollback, now!)

1.3. User registration webpage

[2/15/2023 10:24 PM CST]

URI: /registration.html

Company Stockmarket

Company Account Registration

First name: [textbox]

Last name: [textbox]

Email address: [textbox]

Phone number: [textbox]

Password: [textbox]

Password must be at least 10 characters, and must contain a letter, a number, and a special character from any of the following ~`!@#\$%^&*-_+=|\/?;:","<>.,{}[]().

Password (Retype): [textbox]

Your email address will be your login ID.

After pressing the register button below, you'll receive an email-address verification email with a link that you must click to verify your email address. You will not be able to log in before verifying your email address.

[Register]

1.3.1. Error notification message(s)

The email address you entered is waiting for verification. Please check your email inbox and click the verification link to verify your email address.

[Resend email-address verification email]

That email address is already registered. Please log in using that email address.

[Reset password via email]

1.3.2. Email address verification notification webpage

[2/15/2023 10:24 PM CST]

URI: /EmailAddressVerificationNotice.html

Company Stockmarket

Verify Your Email Address

Email address: xx

You cannot log in until you verify your email address.

Please check your email inbox and click the verification link to verify your email address.

[Resend email-address verification email]

1.3.3. Email address verification email content

Company Stockmarket

Verify Your Email Address

Email address: xx

You cannot log in until you verify your email address.

Please click the verification link to verify your email address.

[Verify my email address]

1.3.4. Email address verification success notification webpage

URL: /GetEmailAddressVerificationSuccessPage (webpage backend generated at this route)

Company Stockmarket

Email Address Verification Success

Email address: xx

Now you can log in to your Company account.

[~~Login~~]

1.3.5. Password reset email content

Company Stockmarket

Reset Your Password

Email address: xx

Click the button below to reset your password.

[Reset password]

(the above button links to /GetResetPasswordPage)

(the above button links to /PasswordReset.html)

1.3.6. Password reset page

URI: /PasswordReset.html (redirected to this URI by backend with parameters in the URI)

URI: ~~/GetResetPasswordPage (webpage backend generated at this route)~~

Company Stockmarket

Reset Your Password

Email address: xx

Enter your new password: []

Enter your new password again: []

[Reset password]

1.3.6.1. HTTPS URI encryption

[3/31/2023 7:14 PM CST]

Search "Does HTTPS encrypt URI parameters?" online.

<https://stackoverflow.com/questions/499591/are-https-urls-encrypted>

1.3.7. Password reset success notification webpage

URL: /ResetPassword (webpage backend generated at this route)

Company Stockmarket

Password Reset Success

Email address: xx

Now you can log in to your Company account with the new password.

[\[Login\]](#)

1.3.8. (References)

[3/15/2023 3:43 PM CST]

search "does https encrypt url" online.

When properly configured, an HTTPS connection guarantees three things: Confidentiality. The visitor's connection is encrypted, obscuring URLs, cookies, and other sensitive metadata.

Introduction to HTTPS - The HTTPS-Only Standard

<https://https.cio.gov/faq/>

Are HTTPS URLs Encrypted? | Baeldung on Computer Science

Baeldung

<https://www.baeldung.com/https-urls-encrypted>

Nov 6, 2022 — Yes, the full URL string is hidden, and all further communication, including the application-specific parameters. However, the Server Name ...

<https://www.baeldung.com/cs/https-urls-encrypted>

[4/24/2023 2:31 PM CST]

Search "JavaScript get number of characters in string" online.

<https://charactercounter.com/count-characters-in-javascript>

Search "javascript regular expression" online.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_expressions

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/match

Search "javascript regular expression how to match from a list of characters" online.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_expressions/Cheatsheet

1.4. Login webpage

URI: /login.html

Company Stockmarket

Register

Login

Email: []
Password: []

[Login]

(successful login takes to the stockmarket homepage.)
(User registration webpage error notification message(s) used.)

1.4.1. Error notification message(s)

The email address you entered is waiting for verification. Please check your email inbox and click the verification link to verify your email address.

[Resend email-address verification email]

Incorrect password. If you can't remember your login password, you can reset the password via your email address.

[Reset password via email]

1.5. User Account Webpage

URI: /account.html

Company Stockmarket

Admin | Logout

('Admin' is shown only when admin account is logged in)

Account Settings

First name: [textbox]

Last name: [textbox]

Email address: [textbox]

Phone number: [textbox]

Password: [textbox]

Password must be at least 10 characters, and must contain a letter, a number, and a special character from any of the following ~`!@#\$%^&*-_+=|V?;:'"<>.,{}[]().

Retype Password: [textbox]

[Update]

Your email address is your login ID and PayPal account ID.

Your email address will not be updated until you click the link in the email-address verification link.

Number of Company shares you own: xx

Company Stock Transaction History

< 1 2 3 ... (last page number) >

[textbox] ['Go to' button]

(What to do when account.html is visited without logging in. "Register | Login" display in the top bar; disabled blank input text boxes; disabled "Update" button; "No records to show."; disabled page number entering text box; disabled "Go to" button. [4/20/2023 3:06 PM CST])

1.6. Admin Webpage

Admin Webpages | admin account management webpage(s)

URI: /admin.html

Company Stockmarket

My Account | Logout

User List

< 1 2 3 ... (last page number) >

[textbox] ['Go to' button]

Transaction List

< 1 2 3 ... (last page number) >

[textbox] ['Go to' button]

1.7. Login required notification webpage (for backend URIs only)

URI: (request URI) (webpage backend generated at request URI)

Company Stockmarket

Login Required

Please [login](#) to use this service.

2. DEPRECATED [3/9/2023 4:38 PM CST]

[3/8/2023 6:37 PM EST]

(make this look what I want. nothing else matters!)

(only nine pages. this definitely can be implemented in a few days! I will complete this in a few days!!!! [3/8/2023 10:15 PM EST])

2.1. Stock Buy and Sell Webpage

URI: /index.html

(Register)|(Login)|(My Account)

Company Stockmarket

For investing in (xx)/(AI, robotics, human longevity biotech, and nuclear-fusion powered outer space tech R&D and commercialization)/(the future).

About (Company)/(Robocentric) (for the demo, links to AllenYoung.dev, not Robocentric.com/About)

Current Company valuation: US\$4 million

Number of authorized Company shares: 40,000

Number of issued Company shares: 16,000 (40% of total)

Number of outstanding Company shares: 16,000 (40% of total)

Number of Company shares available for purchase: 2,000 (5% of total)

Price per share: US\$100

Cash value of the shares available for purchase: US\$200,000

[Log in or register to buy Company stocks]

(the above displayed, only if not logged in. when the above button is pressed, a floating center window for logging in.)

Number of Company shares I own: xx
(x% of the authorized Company shares)
(the above is displayed only if logged in)

(the below is grayed out and disabled if not logged in.)
[Buy button][Sell button]

FAQ

What's the current Company ownership structure?

Supervoting vs. non-supervoting Company stocks

Can I return my purchased Company share(s) and get a refund?

What the Company share liquidation plan?

Shares outstanding vs. Treasury stock vs. Issued shares vs. Authorized capital vs. Unissued shares

Company stock liquidation options (current and post-IPO)

Investor risk warning

Terms and conditions

Fees

About Company

(the above are all pop-up window links)

For inquiries and support requests, please send an email to support@allenyoung.dev.

(footer with relevant info and links)

Contact

2.1.1. Notes

https://en.wikipedia.org/wiki/Shares_outstanding

https://en.wikipedia.org/wiki/Issued_shares

https://en.wikipedia.org/wiki/Treasury_stock

https://en.wikipedia.org/wiki/Authorised_capital

2.2. Buy page

URI: /buy.html

Company Stockmarket

[My account](#)

Buy Company Stocks

Share(s) purchased by someone else while you're on this page will be automatically removed from the below list. Pressing the checkout button reserves the selected stocks for 5 minutes; if you do not make the payment within 5 minutes, you'll be automatically redirected to this page.

a list of rows to select for buying, with next page navigation at the bottom if and as needed.

Company offering at the top, then listed in increasing share price.

(seller type, Company or investor) (Share price) (Fee per share) (Number of shares available)

(Number of shares to buy text box, and auto-calculated amount)

< 1 2 3 ... (last page number) >

[textbox] ['Go to' button]

(I need to resolve the issue of other people buying stocks. there must be a tick on this React webpage with auto-update of stock list, total number of shares to buy, and total price of shares to buy. Use React state to keep track of and repeatedly check the selected share list item ID and number of shares to buy. pressing the checkout button reserves the selected stocks for 5 minutes; the countdown on the Checkout page.)

Total number of Company shares to buy: xx

Total price of Company shares to buy: US\$xx

[Checkout]

2.3. Buy checkout page

Stock Purchase Checkout Webpage

[2/15/2023 10:24 PM CST]

URI: /checkout.html

Company Stockmarket

My account

Checkout

Total number of Company shares to buy: xx

Total price of Company shares to buy: US\$xx

Company share reservation time remaining: hh:mm:ss (automatic countdown)

After you make the payment, you'll be taken to a receipt page and a receipt will be email to you. The transaction record will be available in your account page.

(PayPal checkout form)

2.4. Purchase receipt page

URI: /receipt.html

Company Stockmarket

My account

Purchase Receipt

(Date, time)

Total number of Company shares bought: xx

Total price of Company shares bought: US\$xx

Transaction ID: xx

Congratulations on purchasing Company share(s)!

A copy of this receipt has been emailed to you.

This transaction record is now available in your account page.

2.5. Sell page

URI: /sell.html

Company Stockmarket

My account

List Company Stocks for Sale

A fee is paid by the buyer at the time of purchase, not by the seller. Listing your Company share(s) for sale will make your share(s) available for the buyers. You can retract your sell listing in your account page before it is sold. If your share(s) get sold, your proceeds will be sent to your PayPal account using your email address, when they become available; it may take up to 15 days or more before your proceeds are sent to your PayPal account.

Number of Company shares I own: xx

Number of shares to list for selling: [(TextBox)]

Sales price per share: [(TextBox)]

Total share price: xx

Fees: (US\$x, auto-calculated) (buyer pays the fee.)

Total share purchase price: US\$xx

['List share(s) for sale' button]

Pressing the above button will display a confirmation pop-up window

2.6. User registration webpage

[2/15/2023 10:24 PM CST]

URI: /registration.html

Company Stockmarket

Company Account Registration

First name:

Last name:

Email address:

Phone number:

Password: [textbox] (Must be at least 10 characters, and must contain a letter, a number, and a special character)

Your email address will be your login ID.

After pressing the register button below, you'll receive an email-address verification email with a link that you must click to verify your email address. You will not be able to log in before verifying your email address.

[Register]

2.7. Email address verification notification webpage

[2/15/2023 10:24 PM CST]

URI: /EmailAddressVerification.html

Company Stockmarket

Verify Your Email Address

...

[Resend email-address verification email]

2.8. User Account Webpage

URI: /account.html

Company Stockmarket

My account

Account Settings

First name: [textbox]

Last name: [textbox]

Email address: [textbox]

Phone number: [textbox]

Password: [textbox]

[Update]

Your email address is your login ID and PayPal account ID.

Your email address will not be updated until you click the link in the email-address verification link.

Company Stock Transaction History

< 1 2 3 ... (last page number) >

[textbox] ['Go to' button]

2.9. Admin Webpages | admin account management webpage(s)

URI: /admin.html

Company Stockmarket

My account

User List

< 1 2 3 ... (last page number) >

[textbox] ['Go to' button]

Transaction List

< 1 2 3 ... (last page number) >

[textbox] ['Go to' button]

3. DEPRECATED [3/9/2023 2:08 PM CST]

Current Company valuation: US\$40 million
Number of authorized Company shares: 400,000
Number of issued Company shares: 160,000 (40% of total)
Number of outstanding Company shares: 160,000 (40% of total)
Number of Company shares available for purchase: 20,000 (5% of total)
Price per share: US\$100
Cash value of the shares available for purchase: US\$2,000,000

4. DEPRECATED [3/8/2023 6:37 PM EST]

URI: /

My magic stocks: xx shares

(Register)(Login)(My Account)

Allen Young's Magic Stockmarket

Be a part of the force that creates the wonderful future of humankind:

Own Allen Young's magic stocks

Current Allen Young's magic stock ownership stats

(an HTML/CSS graphics, a bar graph, with percentage owned by Allen Young, not for issuance yet in percentage, remaining available shares for this fundraising round in percentage, and issued shares not owned by Allen Young in percentage.)

Robocentric valuation: US\$5 million
Outstanding/xx Robocentric stocks: xx shares
Issued Robocentric stocks: xx shares
Robocentric stock price: US\$xx
Remaining number of Robocentric stocks to issue in this fundraising round: xx shares (US\$xx)

Robocentric fundraising rounds
Supervoting vs. non-supervoting Robocentric stocks
Robocentric stock liquidation options (current and post-IPO)
Investor risk warning
Terms and conditions
Fees
About Robocentric
(the above are all pop-up window links)

[Log in or register to buy Robocentric stocks]

(the above displayed, only if not logged in. when the above button is pressed, a floating center window for logging in.)

(the below is grayed out and disabled if not logged in.)

My Robocentric stocks: xx shares

[Buy tab][Sell tab]

~~Buy Robocentric shares~~

~~Number of shares to buy: [(TextBox)] (US\$x)~~

~~[Buy]~~

~~(pressing the Buy button displays a
confirmation pop-up window with payment
entering or selecting form)~~

~~Sell Robocentric shares~~

~~Number of shares to sell: [(TextBox)] (US\$x)~~

~~[Sell]~~

~~(pressing the Sell button displays a
confirmation pop-up window with payment
entering or selecting form)~~

(footer with relevant info and links)

Contact

(what else?)

4.1. Buy tab

a list of rows to select for buying, with next page navigation at the bottom if and as needed.

Robocentric offering at the top.

~~(Seller name or Alias) (Share price) (Number of shares available) (Number of shares to buy text
box, and auto-calculated amount) (Buy button)~~

~~(pressing the Buy button displays a confirmation pop-up window with payment entering or
selecting form)~~

4.2. Sell tab

Share sell price: [(TextBox)]

Number of shares to list for selling: [(TextBox)] (US\$x)

Fees: (US\$x, auto-calculated)

[Sell]

(pressing the Sell button displays a confirmation pop-up window with fee payment entering or
selecting form)

5. Reference

5.1. UI/UX design guides

[4/20/2023 11:09 AM CST]

Search "ui ux guidelines" online.

<https://webstandards.ca.gov/design-principles/>

<https://developer.apple.com/design/tips/>

<https://developer.apple.com/design/human-interface-guidelines/guidelines/overview/>

<https://grow.google/certificates/ux-design/>

<https://medium.com/@revanrgh/user-interface-guideline-101-design-guide-for-academic-website-120f7ad71658>

https://grow.google/certificates/#?modal_active=none

5.2. Color code reference

[3/24/2023 8:46 AM EST]

<https://www.computerhope.com/htmcolor.htm>

5.3. HTML grid

[3/24/2023 10:57 AM CST]

Search "html grid gap top" online.

https://www.w3schools.com/cssref/pr_grid-gap.php

<https://developer.mozilla.org/en-US/docs/Web/CSS/gap>

IV. Web-App Backend Design

1. Stock Database

[7/10/2023 5:49 PM EST]

In the initial and current implementation, no separation of data across multiple databases. Only one database.

2. User Database

[7/10/2023 5:49 PM EST]

In the initial and current implementation, no separation of data across multiple databases. Only one database.

3. Payment Processing Via PayPal

Payment Processing Via ~~Stripe~~

~~[11/26/2022 10:27 PM CST] US\$1 minimum.~~

US\$100 minimum, multiple of US\$100 + processing fee of 5%.

4. RESTful API

user registration API

email address verification API

user login API

get available shares API

get number of owned shares API

get user info API

set/update user info API

get admin info API

set/update admin info API

5. GraphQL API

([2/18/2023 11:23 PM EST] same as RESTful API?)

V. Different Implementation Editions and Software Frameworks Used

Different implementation editions and ~~software technology stacks~~

1. Frontend

- Angular
- React
- Vue
- C# .NET Core
- JQuery
- C# ASP.NET Core Razor Pages
- C# ASP.NET Core MVC
- C# .NET Blazor Server
- C# .NET Blazor WebAssembly

2. Backend

- Node/Express.js
- Node/Next.js
- Node/Nest.js

- Python/Flask
- Golang/Gin
- PHP/Laravel
- Ruby/Roda
- C# ASP.NET Core controller-based API
- C# ASP.NET Core minimal API
- Java Spring Boot
- Kotlin Spring Boot
- Kotlin Ktor
- Implementing both RESTful API and GraphQL API
- Reactive programming (need ot be more specific on this.)

https://en.wikipedia.org/wiki/Reactive_programming

3. Full-stack

- Python/Django
- Ruby on Rails
- PHP/Laravel

VI. Backend I/O

1. Database schema

[3/9/2023 7:30 PM CST]

1.1. Stock stats table columns

- Primary key
- Current Company valuation
- Number of authorized Company shares
- Number of issued Company shares
- Number of outstanding Company shares
- Number of Company shares available for purchase (must be greater than or equal to 0)
- Price per share
- Cash value of the shares available for purchase (must be greater than or equal to 0.0)
- Stats entry datetime (integer, as Unix Time, the number of seconds since 1970-01-01 00:00:00 UTC.)

(a new entry added after each transaction. the latest stat set is used for display and calculation.)

1.2. Stock reservation table columns

- Primary key
- Stock stats table entry primary key
- Number of reserved stocks
- Stock reservation start datetime (integer, as Unix Time, the number of seconds since 1970-01-01 00:00:00 UTC.)
- Stock reservation end datetime (integer, as Unix Time, the number of seconds since 1970-01-01 00:00:00 UTC.)
- Reserving-user access token

(I don't think stock reservation table is needed; transactions table can probably do; check the PayPal processing code to determine whether transactions table can provide the needed function. if transactions table can do the job, no cron needed for checking and updating transactions table entries, since the homepage visit will do the job. actually, stock reservation table is needed; separate the stock reservation and payment processing in two separate backend routes.)

1.3. Users table columns

- Primary key
- First name
- Last name
- Email address
- Email address verified (true or false)
- Phone number
- Salt-hashed password
- Password salt
- Admin (true or false)
- Access token
- ~~• Access token expiration datetime (this is not needed for this demo)~~
- User deleted (true or false) (reserved for potential future use, for keeping the primary key unique in all cases)

1.4. Transactions table columns

- Primary key
- User primary key
- Stock stats table entry primary key
- Transaction start datetime (integer, as Unix Time, the number of seconds since 1970-01-01 00:00:00 UTC.)
- Transaction end datetime (integer, as Unix Time, the number of seconds since 1970-01-01 00:00:00 UTC.)

- PayPal transaction (order) ID
- Company stock transaction ID
- Number of shares
- ~~Price per share (not needed in the stock stats table.)~~
- Payment processing initiated (true or false) (a stock stats table entry is added with updated number of available shares for purchase at payment processing initiation. every hour, a cron program checks the transactions table for entries with payment processing initiation but no completion six minutes after the transaction start datetime [i.e. null or blank transaction end datetime]; failed entries result in adding an updated stock stats table entry with corrected number of available shares for purchase, using each failed entry's stock stats table entry with total number of shares and the latest stock stats table entry total number of shares, and payment processing failure is set to true in each failed entry.)
- Payment processing completed (true or false)
- Payment processing status ('success', 'payment decline' or 'payment timeout')
- Entry update counter (must be smaller than or equal to 1)

2. Backend routes and I/Os

[3/9/2023 7:22 PM CST]

webpage (HTML) output if and when applicable.

JSON output if and when applicable.

2.1. Stockmarket home page routes

/CheckAccessTokenValidity

- Inputs (into backend):
 - Access token (as HTTPS header entry, not by cookie)
- Output (from backend):
 - 'valid' or 'invalid'

/GetCompanyStockData

- Output (from backend):
 - Company stock data

/GetNumberOfCompanySharesOwnedByUser

- Inputs (into backend):
 - Access token (by cookie)
- Output (from backend):
 - If invalid access token, 'invalid access token'
 - If valid access token, number of Company shares owned by user

2.2. Login webpage routes

/Login

- Inputs (into backend):
 - User email address
 - User password
- Output (from backend):
 - If valid user email address and user password, access token (returned as a cookie at browser close, used in every access-controlled backend request processing), and admin status (returned as a cookie, true or false)
 - If invalid user email address and/or user password, 'invalid login credentials'

/Logout

2.3. Buy checkout page routes

/Checkout

- Inputs (into backend):
 - Access token (by cookie)
 - Admin status (by cookie)
 - Total number of Company shares to buy
- Output (from backend):
 - Buy checkout page response (HTML)
 - PayPal payment processing

/Checkout/api/orders

- Inputs (into backend):
 - Access token (by cookie)
 - Admin status (by cookie)
- Output (from backend):
 - If the stock reservation table entry with the reserving-user access token doesn't exist, an HTML page notifying the stock reservation requirement, and a stockmarket homepage link to start over.
 - If the reservation is expired, the stock reservation table entry removal, and an HTML page notifying the stock reservation expiration, and a stockmarket homepage link to start over.
 - If no error, a transactions table entry without the 'payment processing completed' and 'payment processing failure' values.

/Checkout/api/orders/:orderID/capture

- Inputs (into backend):
 - Access token (by cookie)
 - Admin status (by cookie)
- Output (from backend):
 - If payment failure, the transactions table entry update for transaction failure.

- If payment success, the transactions table entry update for transaction completion.
 - Purchase receipt (emailed to the user)
 - Purchase receipt page HTML file (saved on the server)
 - Purchase receipt page HTML file (saved on AWS S3)

/Receipt/:orderId

- Inputs (into backend):
 - Access token (by cookie)
 - Admin status (by cookie)
 - PayPal transaction (order) ID
- Output (from backend):
 - Purchase receipt page response (HTML)

2.4. User registration webpage routes

/CheckEmailAddress

- Inputs (into backend):
 - Email address
- Output (from backend):
 - 'waiting verification', 'already registered', 'available for registration', or 'invalid email address format'

/RegisterUser

- Inputs (into backend):
 - First name
 - Last name
 - Email address
 - Phone number
 - Password
- Output (from backend):
 - 'invalid email address format', 'invalid password format', 'success' or 'failure'
 - Access token

/SendEmailAddressVerificationEmail

- Inputs (into backend):
 - Access token
- Output (from backend):
 - If the access token doesn't exist, 'invalid access token'.
 - If the access token exists, the email address verification email sent (via AWS SES) to the email address in the users table with the access token.

/GetEmailAddressVerificationSuccessPage

- Inputs (into backend):
 - Access token

- Output (from backend):
 - If the access token doesn't exist, an HTML page indicating the error status.
 - If the access token exists, the 'email address verified' field value set to 'true' in the users table entry, and the email address verification success notification HTML page response.

/SendResetPasswordEmail

- Inputs (into backend):
 - Email address
- Output (from backend):
 - 'invalid email address format' if applicable.
 - If the input email address doesn't exist in the users table, 'invalid email address'.
 - If the input email address isn't verified yet, 'unverified email address'.
 - 'database operation failure' if applicable.
 - If database update operation success, the password reset email sent to the input email address (via AWS SES).

~~/GetResetPasswordPage~~

- ~~• Inputs (into backend):~~
 - ~~○ Access token~~
- ~~• Output (from backend):~~
 - ~~○ If the access token doesn't exist, return an HTML page indicating the error status.~~
 - ~~○ If the access token exists, return the password reset HTML page.~~

/ResetPassword

- Inputs (into backend):
 - Access token (as HTTPS header/body entry)
 - New password (as HTTPS header/body entry)
- Output (from backend):
 - 'invalid password format' if applicable.
 - If valid password format, the password in the users table update, and the password reset success notification HTML page response.

2.4.1. References

[3/15/2023 6:05 PM CST]

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>

<https://developer.mozilla.org/en-US/docs/Web/API/Headers/entries>

https://en.wikipedia.org/wiki/List_of_HTTP_header_fields

2.5. User Account Webpage routes

/GetUserAccountSettings

- Inputs (into backend):

- Access token (as cookie)
 - Admin status (as cookie)
- Output (from backend):
 - If the access token doesn't exist, 'invalid access token'.
 - If the access token exists in the users table, the applicable pieces of user info.
 - First name
 - Last name
 - Email address
 - Phone number

/UpdateUserAccountSettings

- Inputs (into backend):
 - Access token (as cookie)
 - Admin status (as cookie)
- Output (from backend):
 - If the access token doesn't exist in the users table, 'invalid access token'.
 - 'invalid email address format' if applicable.
 - 'invalid password format' if applicable.
 - If database error, 'database update error'; otherwise, 'success'.

/GetUserStockTransactionHistory

- Inputs (into backend):
 - Access token (as cookie)
 - Admin status (as cookie)
 - Page number (as HTTPS header/body entry)
- Output (from backend):
 - If the access token doesn't exist in the users table, 'invalid access token'.
 - If valid access token, Company stock transaction history page contents (JSON [or HTML]).

2.6. Admin Webpage routes

/GetUserList

- Inputs (into backend):
 - Access token (as cookie)
 - Admin status (as cookie)
 - Page number (as HTTPS header entry)
- Output (from backend):
 - If the access token doesn't exist in the users table, 'invalid access token'.
 - If valid access token, User List page contents (JSON [or HTML]).

/GetTransactionHistory

- Inputs (into backend):
 - Access token (as cookie)

- Admin status (as cookie)
 - Page number (as HTTPS header entry)
- Output (from backend):
 - If the access token doesn't exist in the users table, 'invalid access token'.
 - If valid access token, Transaction List page contents (JSON [or HTML]).

2.7. (References)

[4/5/2023 7:31 PM EST]

<https://expressjs.com/en/4x/api.html#app.METHOD>

<https://learning.postman.com/docs/sending-requests/requests/>

2.7.1. HTTP request types

[4/6/2023 2:08 PM CST]

Search "HTTP request types" online.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>

2.7.2. HTTP request type for login

[4/6/2023 2:14 PM CST]

Search "HTTP request type for login" online.

<https://stackoverflow.com/questions/5868786/what-method-should-i-use-for-a-login-authentication-request>

3. Cookies

access_token (UUID)

admin (true or false)

4. OpenAPI document specification strategies and references

[4/29/2023 8:38 PM CST]

<https://swagger.io/docs/specification/basic-structure/>

<https://swagger.io/docs/specification/authentication/cookie-authentication/>

<https://spec.openapis.org/oas/v3.1.0#security-scheme-object>

<https://swagger.io/docs/specification/authentication/api-keys/>

<https://swagger.io/docs/specification/authentication/cookie-authentication/>

https://en.wikipedia.org/wiki/Cross-site_request_forgery#Prevention

VII. Backend I/O Processes Spec

1. Stockmarket home webpage route I/O processes spec

1.1. /CheckAccessTokenValidity route I/O process spec

1. Check the input access token in the users table. If the access token exists in the database, return 'valid' in response; otherwise, return 'invalid'. If database access error, send an email to the server admin using AWS SES, and return 'database access error' (the frontend should display 'Operation error. Please try again later.'). If no access token in the cookie, return 'No accessToken cookie'.

GET HTTP request

1.2. /GetCompanyStockData route I/O process spec

1. In an atomic database transaction, update "Number of Company shares available for purchase" in the stock stats table for the stock reservations to be removed (using a subquery), and remove all the expired entries in the stock reservation table, and non-expired entry(ies) with the reserving-user access token (if the user is logged in). (Note that since this is a pure write atomic database transaction, no other write transaction can start in SQLite during this write-only atomic database transaction.)
2. In a pure write atomic database transaction, update the "Number of Company shares available for purchase" in the stock stats table for all the initiated, but uncompleted transaction after a timeout period (using a subquery); mark every initiated, but uncompleted transaction after a timeout period as 'payment timeout' in the "Payment processing status" field. Ordinarily, there should be no initiated, but uncompleted transaction, so email the admin when such a transaction exists.
3. Return the latest company stock data using the latest entry in the stock stats table.

If database access error, send an email to the server admin using AWS SES, and return 'database access error' (the frontend should display 'Operation error. Please try again later.').

GET HTTP request

1.3. /GetNumberOfCompanySharesOwnedByUser route I/O process spec

1. Check the input access token in the users table. If the access token doesn't exist in the database, return 'invalid access token' in response; otherwise, store in a data variable the user primary key corresponding to the access token.

2. Using the user primary key, and the transactions table, add up all the numbers of purchased shares in all the completed transactions, with a single SQL command; return the total number of purchased shares in response.

If database access error, send an email to the server admin using AWS SES, and return 'database access error' (the frontend should display 'Operation error. Please try again later.').

GET HTTP request

2. Login webpage route I/O processes spec

2.1. /Login route I/O process spec

1. Check if the input user email address exists in the users table; if not, return 'invalid login credentials'. If unverified email address, return 'unverified email address'.
2. Using the input user email address, retrieve the salt-hashed password and the password salt from the users table in database.
3. Using the password salt, salt-hash the input user password.
4. Compare the salt-hashed password in database, and the computed salt-hashed password; if they do not match, return 'invalid login credentials'; if they match, generate an access token, store it in the users table in database, and return the access token and the admin status as cookies (i.e. set cookies).

If database access error, send an email to the server admin using AWS SES, and return 'database access error' (the frontend should display 'Operation error. Please try again later.').

POST HTTP request

2.1.1. Login cybersecurity measures

[4/26/2023 2:44 PM EST]

The login API should not give out information that can be potentially used in hacking a user's account.

For example, if the login API returns a value such as 'invalid email address' or 'unregistered email address' or 'invalid password' (which indicates that the email address is present on the system), a hacker can find out whether an account exists or not with an email address.

As such, if the input email and/or password is incorrect, instead of indicating whether an invalid or incorrect email address and/or password has been given, the login API should return 'invalid login credentials', giving no information about the correctness of email address and password.

When the login API returns 'invalid login credentials', the frontend should display "If you have an account with us with the email address you've entered, and you cannot remember your password, [click here](#) to reset your password via email."

When "[click here](#)" is clicked, the /SendResetPasswordEmail/:emailAddress API should return a webpage with the following message, that does not indicate whether the provided email address is on the system or not (hence valid or not): "If you have an account with us with the email address you've provided, we've sent a password reset email to the email address you've provided. Please check your email inbox for resetting your login password."

The /Login and /SendResetPasswordEmail/:emailAddress route methods must implement these login cybersecurity measures.

Also, the login API should return "You have not verified your email address yet. If you need another email address verification email sent to your email address, [click here](#).", if the email address is not yet verified. Since people verify their email address rather quickly, exposing this information via the login API poses minimal or no risk to hacking.

([4/26/2023 3:15 PM CST] Actually, scratch the above idea. The registration API can be used to check whether an email address is on the system or not, since the API must return whether the email address is available for registration or not.)

ORIGINAL NOTES

Update /Login and/or /SendResetPasswordEmail route methods so that only 'invalid login credentials' is sent, and it is not possible for hacker to check whether an email address is registered or not. this may not have to be done for now, but think about it. [4/18/2023 3:25 PM CST]

2.1.2. Reference

[3/16/2023 1:32 PM CST]

search "how to salt password in JavaScript" online.

[https://en.wikipedia.org/wiki/Salt_\(cryptography\)](https://en.wikipedia.org/wiki/Salt_(cryptography))

<https://heynode.com/blog/2020-04/salt-and-hash-passwords-bcrypt/>

<https://www.makeuseof.com/nodejs-bcrypt-hash-verify-salt-password/>

<https://stackoverflow.com/questions/17201450/salt-and-hash-password-in-nodejs-w-crypto>

<https://www.techtarget.com/searchsecurity/definition/salt>

https://en.wikipedia.org/wiki/Cryptographic_hash_function

search "most popular password hashing algorithms" online.

<https://codesigningstore.com/what-is-the-best-hashing-algorithm>

<https://codesigningstore.com/what-is-the-most-secure-hashing-algorithm>

To the time of writing, SHA-256 is still the most secure hashing algorithm out there. It has never been reverse engineered and is used by many software organizations and institutions, including the U.S. government, to protect sensitive information. It's so secure that it's even a central part of Bitcoin's cryptographic protocol.

search "How long does it take to crack SHA-256 with known salt?" online.
To crack a hash, you need not just the first 17 digits to match the given hash, but all 64 of the digits to match. So, extrapolating from the above, it would take $10 * 3.92 * 10^{56}$ minutes to crack a SHA256 hash using all of the mining power of the entire bitcoin network.
How long would it take to crack SHA-256? - Quora
<https://www.quora.com/How-long-would-it-take-to-crack-SHA-256>

search "Is it safe to transmit cryptographic salt?" online.
<https://crypto.stackexchange.com/questions/72430/is-it-safe-to-reuse-the-salt-for-generating-symmetric-keys-for-a-chat-applicatio>
<https://stackoverflow.com/questions/28680211/is-it-acceptable-to-publicly-expose-salts>
https://en.wikipedia.org/wiki/Cryptographically_secure_pseudorandom_number_generator

search "Do you need nonce when using HTTPS?" online.
<https://security.stackexchange.com/questions/165919/what-is-the-use-of-nonces-in-https-apis>
<https://www.quora.com/Do-you-need-to-use-a-nonce-if-you-are-using-SSL-1>
https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/nonce

2.2. /Logout route I/O process spec

1. Delete the cookies in the client web browser.

DELETE HTTP request

2.2.1. Reference

[4/25/2023 5:11 PM CST]
Search "What HTTP method should log out use?" online.
<https://www.baeldung.com/logout-get-vs-post>
<https://stackoverflow.com/questions/3521290/logout-get-or-post>

Search "http delete request" online.
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods/DELETE>

Search "using http delete for logout" online.
<https://stackoverflow.com/questions/3521290/logout-get-or-post>
<https://softwareengineering.stackexchange.com/questions/196871/what-http-verb-should-the-route-to-log-out-of-your-web-app-be>

Search "document.cookie" online.
<https://developer.mozilla.org/en-US/docs/Web/API/Document/cookie>

Search "How to set document.cookie to blank" online.

<https://stackoverflow.com/questions/15914744/javascript-document-cookie-always-returns-empty-string>

HttpOnly cookies cannot be accessed from Javascript and session cookies are usually set as HttpOnly cookies. See also this StackOverflow question: How to read a secure cookie using JavaScript

So... check whether the cookie you want to read has the 'HttpOnly' flag set... If so, you know the culprit. It's not a bug, it's a feature!

3. Checkout webpage route I/O processes spec

3.1. /Checkout/:numberOfSharesToBuy route I/O process spec

1. Check the input access token in the users database table; if it doesn't exist, return 'invalid access token'.
2. Using the input access token, check the admin status in the users database table; if the database admin status does not match the input admin status, set the admin-status cookie to the correct value; store the database admin status value in a data variable.
3. If it is admin, return an HTML page notifying that an admin cannot purchase stocks, with a link to take back to the stockmarket homepage.
4. In a pure write atomic database transaction, update the "Number of Company shares available for purchase" in the stock stats table for the stock reservations to be removed (using a subquery), and remove all the expired entries in the stock reservation table, and non-expired entry(ies) with the reserving-user access token. (Note that since this is a pure write atomic database transaction, no other write transaction can start in SQLite during this write-only atomic database transaction. Use CHECK or NON NULL constraint as appropriate.)
5. In a pure write atomic database transaction, update the "Number of Company shares available for purchase" in the stock stats table for all the initiated, but uncompleted transaction after a timeout period (using a subquery); mark every initiated, but uncompleted transaction after a timeout period as 'payment timeout' in the "Payment processing status" field. Ordinarily, there should be no initiated, but uncompleted transaction, so email the admin when such a transaction exists.
6. Perform a pure write atomic database transaction for checking the available number of shares to buy using the stock stats table and the 'total number of Company shares to buy' input, and adding an applicable entry to the stock reservation table, and updating the "Number of Company shares available for purchase" and "Cash value of the shares available for purchase" in the stock stats table, when enough number of shares are available for purchase. (Make this a pure write atomic database transaction, and use CHECK or NON NULL constraint as appropriate. https://www.sqlite.org/lang_createtable.html#ckconst.)
 - 6.1. If not enough number of shares are available for purchase, return an HTML page notifying that not enough number of shares are available for purchase, with a link to take back to the stockmarket homepage.

- 6.2. If enough number of shares are available for purchase, generate a PayPal client token for this transaction, and return an appropriate buy checkout HTML page with the PayPal order info.

If database access error, send an email to the server admin using AWS SES, and return 'database access error' (the frontend should display 'Operation error. Please try again later.').

POST HTTP request

3.2. /Checkout/api/orders route I/O process spec

1. Check the input access token in the users database table; if it doesn't exist, return 'invalid access token'.
2. Using the input access token, check the admin status in the users database table; if the database admin status does not match the input database status, set the admin-status cookie to the correct value; store the database admin status value in a data variable.
3. If it is admin, return an HTML page notifying that an admin cannot purchase stocks, with a link to take back to the stockmarket homepage.
4. Retrieve the stock reservation table entry with the reserving-user access token.
 - 4.1. If the stock reservation table entry with the reserving-user access token doesn't exist, return an HTML page notifying the stock reservation requirement, and a stockmarket homepage link to start over.
 - 4.2. If the reservation is expired, in an atomic database transaction, remove the stock reservation table entry with the reserving-user access token, update "Number of Company shares available for purchase" in the stock stats table for the removed stock reservations, then return an HTML page notifying the stock reservation expiration, and a stockmarket homepage link to start over.
5. Create a PayPal order.
 - 5.1. If PayPal order creation error, return the error.
 - 5.2. If no error, perform the following.
 - 5.2.1. In a pure write atomic database transaction, perform the following. (Note that since this is a pure write atomic database transaction, no other write transaction can start in SQLite during this write-only atomic database transaction.) If database operation error, email the admin, and return an error.
 - 5.2.1.1. Create a transactions table entry, using the retrieved reserving-user stock reservation table entry data, without the 'payment processing completed' and 'payment processing failure' values. (Use a subquery in the INSERT statement to get the number of shares to purchase in the stock reservation table; use CHECK or NON NULL constraint to ensure that the value from the subquery with the reserving-user access token is not NULL.
https://www.sqlite.org/lang_transaction.html,
https://www.sqlite.org/lang_conflict.html,
https://www.sqlite.org/lang_createtable.html#ckconst,
https://www.sqlite.org/lang_createtable.html#notnullconst.)
 - 5.2.1.2. Remove the stock reservation table entry with the reserving-user access token.

5.2.2. Return the PayPal order.

POST HTTP request

OpenAPI callback for /Checkout? (Find out, and use the answer!)

<https://spec.openapis.org/oas/v3.1.0#callback-object-examples>

A linked operation of /Checkout? (Find out, and use the answer!)

<https://spec.openapis.org/oas/v3.1.0#examples>

3.2.1. Double web-browser tab checkout case handling

[3/22/2023 12:18 PM CST]

Consider the following scenario.

1. In the first web-browser tab, the user visits the stockmarket homepage, enters 5 stocks to buy, and presses the "Proceed to Checkout" button and goes over to the checkout page.
2. In the second web-browser tab, the user visits the stockmarket homepage, enters 10 stocks to buy, and presses the "Proceed to Checkout" button and goes over to the checkout page.
3. In the second web-browser tab, the user makes a payment with a credit card that is successfully processed, paying for 10 stocks, instead of 5 stocks, even though in the webpage, it says 5 stocks, due to the second web-browser tab use.

The current design does the above.

In order to mitigate the above problem, in the checkout page, in the frontend code, a timer must constantly run every five seconds to check and update the number of shares to buy. (This is what is implemented.)

The last stock reservation is used when making a payment transaction.

3.2.2. Notes

([3/16/2023 3:47 PM CST] issues to resolve. access token, total number of Company shares to buy [storage place], Transactions table entry, error raising and handling [in Express.js route method])

([3/16/2023 4:12 PM CST] maybe use a cookie to store the number of shares to buy, on the client side [instead of storing it on the server side]? Perform an atomic database transaction for adding a Transactions table entry with number of available shares check?)

([3/16/2023 4:35 PM CST] make sure that app.js handles error response from /Checkout/api/orders and /Checkout/api/orders/:orderId/capture properly.)

([3/16/2023 4:40 PM CST] using cookie for number of shares to buy and returning error status from /Checkout/api/orders are what I'm leaning toward currently.)

([3/16/2023 5:03 PM CST] Express.js res.cookie() and res.clearCookie() use in backend.)

([3/17/2023 11:47 AM CST] an issue to resolve first. calculating the number of available shares to buy. in the stock reservation table, the stock reservation for this transaction is already

recorded. however, nothing from the client side can be trusted. so, what should be done regarding that? answer: use the reserving-user access token in the stock reservation table entry.)

(FINAL DECISION [3/17/2023 11:58 AM CST] no cookie use for the 'total number of Company shares to buy'. instead, use the stock reservation table entry.)

3.3. /Checkout/api/orders/:orderId/capture route I/O process spec

1. If payment failure, in a pure write atomic database transaction, update the transactions table entry for transaction payment failure (in the SQL command, use CHECK constraint to ensure that the transaction isn't marked for completion ~~and failure~~ by checking that "Payment processing status" field is NULL, if already completed, return an HTML page notifying as so, if and as needed, perform two separate SQL commands, one updating "Payment processing completed" field and another updating "Payment processing status" field, in an atomic database transaction to check for completion), and update (rollback) "Number of Company shares available for purchase" and "Cash value of the shares available for purchase" in the stock stats table using the "Number of shares" entry in the transactions table. (Note that since this is a pure write atomic database transaction, no other write transaction can start in SQLite during this write-only atomic database transaction. Use CHECK or NON NULL constraint as appropriate.) (UPDATE [4/3/2023 11:27 AM CST] not logging payment failure to database, since the user can retry making a payment on the checkout page.)
2. If payment success, the transactions table entry update for the successful transaction completion, and "Number of issued Company shares" and "Number of outstanding Company shares" updates in the stock stats table (in the SQL command, use CHECK constraint to ensure that the transaction isn't marked for completion ~~and failure~~ by checking that "Payment processing status" field is NULL, if already completed, return an HTML page notifying as so, if and as needed, perform two separate SQL commands, one updating "Payment processing completed" field and another updating "Payment processing status" field, in an atomic database transaction to check for completion).
 - 2.1. Save the payment record file on the server.
 - 2.2. Save the payment record file on AWS S3.

If the transaction is already marked for completion, and the payment succeeded, return an HTML page notifying the user that this transaction ~~(isn't valid due to the invalid delay between PayPal payment order creation and the payment capture)~~(has already failed), and that contacting the support with the PayPal order ID is required for the refund; also, email the admin (since this is an unusual case); if the Payment processing status is 'payment decline' or 'payment timeout', return an HTML page notifying the user appropriately.

This design properly handles calling /Checkout/api/orders/:orderId/capture route with the same orderId multiple times, even if PayPal API doesn't.

POST HTTP request

OpenAPI callback for /Checkout? (Find out, and use the answer!)

<https://spec.openapis.org/oas/v3.1.0#callback-object-examples>

A linked operation of /Checkout? (Find out, and use the answer!)

<https://spec.openapis.org/oas/v3.1.0#examples>

3.3.1. PayPal API order expiration

[3/22/2023 2:51 PM CST]

Search "PayPal API order expiration" online.

Authorized payments are best captured within three days of authorization but are available to capture for up to 29 days. After the three-day honor period, the original authorized payment expires and you must re-authorize the payment.

Orders API - PayPal Developer

<https://developer.paypal.com/docs/api/orders/v2/>

~~If the transaction is already marked for completion and failure, in a pure write atomic transaction, decrease "Number of Company shares available for purchase" in the stock stats table using the "Number of shares" entry in the transactions table, update the transaction entry to mark success, and email the admin (since this is an unusual case).~~

3.4. /Receipt/:orderID route I/O process spec

1. Retrieve the transaction record in the transactions table, using the access token, the stock stats table, the users table, and the PayPal transaction (order) ID. If no transaction record with that information exists, return an HTML page indicating the error, with a link to go back to the stockmarket homepage, and support contact info.
2. Return the purchase receipt HTML page. Email the purchase receipt to the user.

GET HTTP request

3.5. Design case considerations

[3/22/2023 2:08 PM CST]

3.5.1. User logout and login during transaction and access token change

[3/22/2023 2:12 PM CST]

If the user logs out and logs back in after proceeding to checkout, /Checkout/api/orders route will properly handle the case of the new access token for the user. /GetCompanyStockData route and /Checkout route will handle the expired stock reservation.

If the user logs out and logs back in after executing `/Checkout/api/orders` route and before executing `/Checkout/api/orders/:orderID/capture` route (programmatically, most likely), and the correct PayPal transaction (order) ID is provided from the client, `/Checkout/api/orders/:orderID/capture` route will handle the request properly; if incorrect PayPal transaction (order) ID is provided from the client, an error will be returned in response.

3.5.2. Transaction initiation success, payment completion failure

[3/22/2023 2:19 PM CST]

In a pure write atomic database transaction, the "Number of outstanding Company shares" in the stock stats table must be updated for all the initiated, but uncompleted transaction after a timeout period; and every initiated, but uncompleted transaction after a timeout period must be marked as 'payment timeout' in the "Payment processing status" field. Ordinarily, there should be no initiated, but uncompleted transaction, so email the admin when such a transaction exists.

3.6. References

[3/16/2023 4:30 PM CST]

<https://developer.paypal.com/home>

<https://developer.paypal.com/api/rest/>

<https://developer.paypal.com/sdk/js/>

<https://developer.paypal.com/sdk/js/reference/#link-createorder>

4. User registration webpage route I/O processes spec

4.1. /CheckEmailAddress route I/O process spec

1. Check the input email address in the users table. Return 'waiting verification', 'already registered', 'available for registration', or 'invalid email address format', as appropriate.

GET HTTP request

(NOTE [5/14/2023 11:11 AM CST] This API end point is not used.)

4.2. /RegisterUser route I/O process spec

1. Check the input email address format. Return 'invalid email address format' if applicable.
2. Check the email address. If it already exists in database, return 'unavailable email address'.
3. Check the input password format. Return 'invalid password format' if applicable.
4. Salt the password, and add the user entry in the users table, with an access token for email verification purposes. Return the access token.

POST HTTP request

4.3. / SendEmailAddressVerificationEmail/:accessToken/:emailAddress route I/O process spec

1. If the accessToken route parameter is not 'null', check the access token in the users table. If the access token doesn't exist, return 'invalid access token'. If the access token exists, send the email address verification email to the email address in the users table with the access token.
2. If the accessToken route parameter is 'null', check the emailAddress route parameter; if the emailAddress route parameter is 'null', return 'email address required' in a 400 HTTP error. If the emailAddress route parameter is not 'null', check the email address in the database; if the email address is not present in the database, return 'invalid email address' in a 400 HTTP error; if the email address is present in the database, and is already verified, return 'verified email address' in a 400 HTTP error; if the email address is present in the database, and is not verified, send the email address verification email to the email address in the users table with the access token.

GET HTTP request

NOTE [4/16/2023 2:13 PM CST] Calling the JavaScript function fetch() in the frontend for a GET request with body generates the following error: "fetch error: TypeError: Failed to execute 'fetch' on 'Window': Request with GET/HEAD method cannot have body." As such, all the request parameters must be in the URL.

4.4. / GetEmailAddressVerificationSuccessPage/:accessToken route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist, return an HTML page indicating the error status. If the access token exists, set the 'email address verified' field value to 'true' in the users table entry, and return the email address verification success notification HTML page.

GET HTTP request

4.5. /SendResetPasswordEmail/:emailAddress route I/O process spec

1. Check the input email address format. Return 'invalid email address format' if applicable.

2. If the input email address doesn't exist in the users table, return 'invalid email address'.
3. If the input email address isn't verified yet, return 'unverified email address'.
4. Update the user entry in the users table, with an auto-generated access token for password reset purposes. Return 'database operation failure' if applicable. If database update operation success, send the password reset email to the input email address, with a button link to the password-reset frontend HTML page with the access token parameter in the URL.

GET HTTP request

4.6. /ResetPassword route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist, return an HTML page indicating the error status.
2. Check the input password format. Return 'invalid password format' if applicable.
3. If valid password format, update the password in the users table, and return the password reset success notification HTML page.

POST HTTP request

5. User account webpage route I/O processes spec

5.1. /GetUserAccountSettings route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist, return 'invalid access token'.
2. If the access token exists in the users table, return the applicable pieces of user info.

GET HTTP request

5.2. /UpdateUserAccountSettings route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist in the users table, return 'invalid access token'.
2. Check the input email address format. Return 'invalid email address format' if applicable.
3. Check the input password format, if it is not blank. Return 'invalid password format' if applicable.
4. Update the user account settings in the users table. If database error, return 'database update error'; otherwise, return 'success'.

POST HTTP request

5.3. /GetNumberOfUserStockTransactions route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist in the users table, return 'invalid access token'. If the access token is an admin's, return 'invalid user'.
2. Return the number of user's stock transactions.

GET HTTP request

5.4. /GetUserStockTransactionHistory/:pageNumber route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist in the users table, return 'invalid access token'.
2. Generate and return the Company stock transaction history page contents (JSON ~~or HTML~~) using the input page number; if no transaction history for the input page number, return blank.

GET HTTP request

6. Admin webpage route I/O processes spec

6.1. /GetTotalNumberOfUsers route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist in the users table, return 'invalid access token'. If the access token isn't an admin's, return 'invalid user'.
2. Return the total number of users.

GET HTTP request

6.2. /GetUserList/:pageNumber route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist in the users table, return 'invalid access token'. If the user is not admin, return 'access prohibited'.
2. Generate and return the User List page contents (JSON ~~or HTML~~) using the input page number; if no transaction history for the input page number, return blank.

GET HTTP request

6.3. /GetTotalNumberOfTransactions route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist in the users table, return 'invalid access token'. If the access token isn't an admin's, return 'invalid user'.
2. Return the total number of transactions.

GET HTTP request

6.4. /GetTransactionHistory/:pageNumber route I/O process spec

1. Check the access token in the users table. If the access token doesn't exist in the users table, return 'invalid access token'. If the user is not admin, return 'access prohibited'.
2. Generate and return the Transaction List page contents (JSON ~~or HTML~~) using the input page number; if no transaction history for the input page number, return blank.

GET HTTP request

7. References

[3/16/2023 11:37 AM CST]

7.1. Difference between HTTP GET and POST

[3/16/2023 11:37 AM CST]

Search "difference between HTTP get and post" online.

Both GET and POST method is used to transfer data from client to server in HTTP protocol but Main difference between POST and GET method is that GET carries request parameter appended in URL string while POST carries request parameter in message body which makes it more secure way of transferring data from client to ...

1 a. Differentiate between GET and POST. b. Explain web ...

https://old.mu.ac.in/wp-content/uploads/2017/11/WP_CBSGS_Paper-3-May-2017_Solution.pdf

https://www.w3schools.com/tags/ref_httpmethods.asp

7.2. HTTP error types

[4/7/2023 4:14 PM CST]

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

VIII. Implementation Methods

[7/10/2023 2:24 PM EST]

Refer to Allen Young's *React.js, Express.js, SQLite3, Jest, Postman, Selenium, and Jenkins Usage Techniques* Udemy course for the specific implementation techniques used in this demo.

1. Project directory structure

[2/19/2023 7:41 AM EST]

- Frontends
 - Angular
 - React
 - Vue
 - jQuery
 - Razor Pages (C# and ASP.NET Core frontend)
 - Blazor (C# and ASP.NET Core frontend)
- Backends
 - Express
 - NestJS
 - NextJS
 - Flask
 - Gin
 - Laravel
 - ~~Roda (or Ruby — not Ruby, do Roda, always do more, never less!)~~
 - Controller-based API (C# and ASP.NET Core backend)
 - Minimal API (C# and ASP.NET Core backend)
- Full-stack
 - Django
 - Ruby
 - Laravel
 - MVC (C# and ASP.NET Core full-stack)
- CI/CD
 - Jenkins

TO DO [2/27/2023 8:21 PM EST] add all of the C# and ASP.NET Core projects

Razor Pages, MVC, Blazor

Controller-based API, Minimal API

~~Real-time web app~~

~~Remote Procedure Call app~~

2. Multiple database management systems support method

[4/7/2023 5:19 PM CST]

Create database-access-(database management system name).js and database-access-(database management system name)-test.js per supported database management system, with the same API.

Use conditional import with dynamic module loading. Use the database management system name from the program argument.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules>
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules#dynamic_module_loading

IX. (move this to Udemy product document) Backend Implementation Methods

1. Nest.js implementation methods

1.1. Nest.js backend database access method

[4/16/2023 11:32 PM EST]

The database-access module developed for the Express.js backend is used in the Nest.js backend.

Since both Express.js and Nest.js run on Node.js, there is no need for a separate database-access module.

2. Next.js implementation methods

2.1. Next.js backend database access method

[4/16/2023 11:32 PM EST]

The database-access module developed for the Express.js backend is used in the Next.js backend.

Since both Express.js and Next.js run on Node.js, there is no need for a separate database-access module.

3. Python Flask implementation methods

3.1. Python Flask backend database access method

[4/16/2023 11:38 PM EST]

3.1.1. Python SQLite3 library

[4/16/2023 11:38 PM EST]

Search "Python SQLite3 library" online.

4. Golang Gin implementation methods

<https://go.dev/doc/tutorial/web-service-gin>

with https, authentication, and authorization. (pull this off! make this happen!)

all the Go package file(s) in the same directory.

<https://go.dev/ref/spec#Packages>

Go package identifier: allen_young_stockmarket

4.1. Golang-Gin backend database access method

[4/16/2023 11:38 PM EST]

4.1.1. Golang SQLite3 library

[4/16/2023 11:38 PM EST]

Search "Golang SQLite3 library" online.

5. PHP Laravel implementation methods

5.1. PHP Laravel backend database access method

[4/16/2023 11:38 PM EST]

5.1.1. PHP SQLite3 library

[4/16/2023 11:38 PM EST]

Search "PHP SQLite3 library" online.

6. Ruby Roda implementation methods

6.1. Ruby Roda backend database access method

[4/16/2023 11:38 PM EST]

6.1.1. Ruby SQLite3 library

[4/16/2023 11:38 PM EST]

Search "Ruby SQLite3 library" online.

7. C# ASP.NET Core controller-based API backend implementation methods

[6/8/2023 1:31 PM CST]

7.1. C# backend database access method

[4/16/2023 11:38 PM EST]

7.1.1. C# SQLite3 library

[4/16/2023 11:38 PM EST]

Search "C# SQLite3 library" online.

8. C# ASP.NET Core minimal API backend implementation methods

[6/8/2023 1:31 PM CST]

9. Java Spring Boot implementation methods

10. Kotlin Spring Boot implementation methods

11. Kotlin Ktor implementation methods

12. Online payment processing implementation

12.1. Paypal payment processing implementation

PayPal error to consider and handle

```
{ "name": "SERVICE_UNAVAILABLE", "message": "Service Unavailable", "debugId":  
"242d24050d469", "links": [ { "href": "https://developer.paypal.com/docs/api/overview/#error",  
"rel": "information_link" } ] }
```

<https://developer.paypal.com/api/rest/responses/>

X. (move this to Udemy product document) Frontend Implementation Methods

1. React Native implementation methods

[4/12/2023 9:46 PM EST]

1.1. React Native implementation main strategy

[4/12/2023 9:46 PM EST]

Straight porting of the React implementation.

2. Angular implementation methods

2.1. Angular implementation main strategy

[3/31/2023 8:23 PM CST]

Implement each React component as an Angular component.

2.2. Angular project directory structure

[3/24/2023 6:23 PM EST]

In the Angular project root directory, an Angular template HTML file for each frontend webpage to be rendered by Angular.

A sub-directory for each Angular component.

All of the following Angular components and project sub-directories.

account-settings-update-form
number-of-shares-owned-by-user
company-stock-transaction-history-page
account-registration-form
user-list-page
transaction-list-page
user-email-address
register-login-or-my-account
company-stock-stats
login-register-or-buy
login-form

2.3. Angular project files

[3/24/2023 6:24 PM EST]

In each Angular component directory, an Angular component template HTML file, and an Angular component TypeScript file.

3. Vue implementation methods

4. jQuery implementation methods

5. C# ASP.NET Core Razor Pages frontend implementation methods

[6/8/2023 1:32 PM CST]

6. C# ASP.NET Core MVC frontend implementation methods

[6/8/2023 1:32 PM CST]

7. C# .NET Blazor Server frontend implementation methods

[6/8/2023 1:32 PM CST]

8. C# .NET Blazor WebAssembly frontend implementation methods

[6/8/2023 1:32 PM CST]

XI. (move this to Udemy product document) Full-stack Implementation Methods

1. Python Django full-stack implementation methods

2. PHP Laravel full-stack implementation methods

3. Ruby on Rails full-stack implementation methods

4. C# and ASP.NET Core full-stack implementation methods

XII. Testing Methods

1. Frontend testing methods

[4/11/2023 10:47 AM CST]

1.1. Jest, Selenium, and Postman frontend-testing sequence

[4/11/2023 10:47 AM CST]

1. (Local backend server start)
2. index.html (without login)
3. account.html (without login)
4. admin.html (without login)
5. registration.html
6. EmailAddressVerificationNotice.html
7. (Email verification URL visit)
8. login.html (for password reset testing)
9. (Password-reset verification URL visit)
10. PasswordReset.html
11. login.html (to log in as non-admin)
12. index.html (with login as non-admin) (checkout testing)
13. account.html (with login as non-admin)
14. admin.html (with login as non-admin)
15. (logout)
16. login.html (to log in as admin)
17. account.html (with login as admin)
18. admin.html (with login as admin)
19. index.html (with login as admin) (checkout testing)
20. (Local backend server end)
21. (Database cleanup)

1.2. Jest testing source code files

[4/15/2023 10:36 AM CST]

1.2.1. Snapshot testing source code file

[4/15/2023 10:36 AM CST]

1.2.2. DOM testing source code file

[4/15/2023 10:36 AM CST]

Create an account, buy a stock, validate everything in every webpage.

2. Postman CI/CD backend testing automation

[4/4/2023 6:08 AM EST]

postman-backend-test.js

2.1. Postman backend API test implementation strategy

[5/5/2023 9:33 AM CST]

~~The exact same test sequence as in the existing, first, Node.js only backend API test code. Database CLI command execution in Postman test script, if and as needed.~~

UPDATE [5/13/2023 6:05 PM CST]

The above is no longer valid. Test every API, in the most appropriate order for testing.

2.2. Postman backend API test request sequence

[5/13/2023 6:09 PM CST]

Write out the API test request sequence, with all the settings or parameters and expected response in every request. [5/13/2023 5:57 PM CST]

1. Pre-account creation test requests.
 - 1.1. Stockmarket home webpage API end-point test requests.
 - 1.1.1. /CheckAccessTokenValidity
 - 1.1.1.1. Send an invalid access token in the cookie.
 - 1.1.1.2. Expect status 400, and 'invalid' text/html response.
 - 1.1.2. /GetCompanyStockData
 - 1.1.2.1. Send no access token in the cookie.
 - 1.1.2.2. Expect the JSON company stock data response.
 - 1.1.3. /GetNumberOfCompanySharesOwnedByUser
 - 1.1.3.1. Send an invalid access token in the cookie.
 - 1.1.3.2. Expect status 401, and 'invalid access token' text/html response.
 2. Account creation test requests.
 - 2.1. User registration webpage API end-point test requests.
 - 2.1.1. /RegisterUser
 - 2.1.1.1. Send a batch of valid user registration info.
 - 2.1.1.2. Expect status 200, and an access token response.
 - 2.1.2. ~~/SendEmailAddressVerificationEmail/:accessToken/:emailAddress~~
 - 2.1.2.1. ([5/14/2023 11:20 AM CST] Not testing this now. In the future, a potential way to automate testing email sending is to use a CLI email client that can access [and maybe even delete] emails with CLI commands. For example, a test email can be sent to an AWS email address with auto-saving received email to an AWS S3 bucket, that can be check with an AWS CLI command.)
 - 2.1.3. /GetEmailAddressVerificationSuccessPage/:accessToken
 - 2.1.3.1. Send the valid access token received from the /RegisterUser API end-point operation.
 - 2.1.3.2. Expect the GetEmailAddressVerificationSuccessPage_success_notification backend-rendered webpage with the email address.
 - 2.1.4. ~~/SendResetPasswordEmail/:emailAddress~~

- 2.1.4.1. ([5/14/2023 11:31 AM CST] Not testing this now. In the future, a potential way to automate testing email sending is to use a CLI email client that can access [and maybe even delete] emails with CLI commands. For example, a test email can be sent to an AWS email address with auto-saving received email to an AWS S3 bucket, that can be check with an AWS CLI command.)
- 2.1.5. /ResetPassword
 - 2.1.5.1. Send a valid new password.
 - 2.1.5.2. Expect status 200, and the ResetPassword_password-reset-success-notification backend-rendered webpage with the email address.
- 3. Post-account creation test requests.
 - 3.1. Login webpage API end-point test requests.
 - 3.1.1. /Login
 - 3.1.1.1. Send a valid login email address and password.
 - 3.1.1.2. Expect status 200, and 'success' text/html response.
 - 3.1.2. /Logout
 - 3.1.2.1. Expect clear-cookie header entries, status 200, and 'success' text/html response.
 - 3.2. Stockmarket home webpage API end-point test requests.
 - 3.2.1. /CheckAccessTokenValidity
 - 3.2.1.1. Send the valid access token in the cookie.
 - 3.2.1.2. Expect status 200, and 'valid' text/html response.
 - 3.2.2. /GetNumberOfCompanySharesOwnedByUser
 - 3.2.2.1. Send the valid access token in the cookie.
 - 3.2.2.2. Expect status 200, and '0' text/html response.
 - 3.3. Checkout webpage API end-point test requests.
 - ~~3.3.1. /Checkout/:numberOfSharesToBuy~~
 - ~~3.3.1.1. Send the valid access token in a cookie.~~
 - ~~3.3.1.2. Expect status 200, and the checkout backend-rendered webpage with the email address.~~
 - 3.3.1.3. ([5/14/2023 12:24 PM CST] Test this in the Selenium end-to-end test. Due to the payment info sending in the frontend JavaScript code, doing this now is an unnecessary complication for a demo development.)
 - ~~3.3.2. /Checkout/api/orders~~
 - ~~3.3.2.1. Send the valid access token in a cookie.~~
 - ~~3.3.2.2. Expect status 200, and the JSON order object.~~
 - 3.3.2.3. ([5/14/2023 12:24 PM CST] Test this in the Selenium end-to-end test. Due to the payment info sending in the frontend JavaScript code, doing this now is an unnecessary complication for a demo development.)
 - ~~3.3.3. /Checkout/api/orders/:orderID/capture~~
 - 3.3.3.1. ([5/14/2023 12:24 PM CST] Test this in the Selenium end-to-end test. Due to the payment info sending in the frontend JavaScript code, doing this in Postman backend API testing is an unnecessary complication for a demo development.)
 - ~~3.3.4. /Receipt/:orderID~~

- 3.3.4.1. ([5/14/2023 12:24 PM CST] Test this in the Selenium end-to-end test. Due to the payment info sending in the frontend JavaScript code, doing this in Postman backend API testing is an unnecessary complication for a demo development.)
- 3.4. User account webpage API end-point test requests.
 - 3.4.1. /GetUserAccountSettings
 - 3.4.1.1. Send the valid access token in a cookie.
 - 3.4.1.2. Expect status 200, and the JSON user account settings object.
 - 3.4.2. /UpdateUserAccountSettings
 - 3.4.2.1. Send the valid access token in a cookie, and a set of new user account settings.
 - 3.4.2.2. Expect status 200, and 'success' text/html response.
 - 3.4.3. /GetNumberOfUserStockTransactions
 - 3.4.3.1. Send the valid access token in a cookie.
 - 3.4.3.2. Expect status 200, and '0' text/html response.
 - 3.4.4. /GetUserStockTransactionHistory/:pageNumber
 - 3.4.4.1. Send the valid access token in a cookie.
 - 3.4.4.2. Expect status 200, and the blank JSON array object.
- 3.5. Admin webpage API end-point test requests.
 - 3.5.1. /GetTotalNumberOfUsers
 - 3.5.1.1. Send the valid admin access token in a cookie.
 - 3.5.1.2. Expect status 200, and '2' text/html response.
 - 3.5.2. /GetUserList/:pageNumber
 - 3.5.2.1. Send the valid admin access token in a cookie, with '1' for pageNumber.
 - 3.5.2.2. Expect status 200, and the JSON user list page contents object.
 - 3.5.3. /GetTotalNumberOfTransactions
 - 3.5.3.1. Send the valid admin access token in a cookie.
 - 3.5.3.2. Expect status 200, and '0' text/html response.
 - 3.5.4. /GetTransactionHistory/:pageNumber
 - 3.5.4.1. Send the valid access token in a cookie.
 - 3.5.4.2. Expect status 200, and the blank JSON array object.

2.3. References

[4/18/2023 11:18 AM CST]

<https://learning.postman.com/docs/writing-scripts/script-references/postman-sandbox-api-reference/>

<https://www.postmanlabs.com/postman-collection/Request.html>

Search "Does Postman script run on Node.js?" online.

<https://learning.postman.com/docs/writing-scripts/intro-to-scripts/>

<https://learning.postman.com/docs/collections/running-collections/intro-to-collection-runs/>

<https://learning.postman.com/docs/postman-cli/postman-cli-overview/>

<https://learning.postman.com/docs/collections/using-newman-cli/integration-with-jenkins/>

<https://learning.postman.com/docs/postman-cli/postman-cli-overview/#deciding-which-command-line-companion-to-use>

<https://learning.postman.com/docs/writing-scripts/script-references/postman-sandbox-api-reference/>

<https://learning.postman.com/docs/collections/using-newman-cli/newman-with-docker/>

<https://learning.postman.com/docs/sending-requests/troubleshooting-api-requests/#using-the-console>

2.3.1. Cookie HTTP header

[5/15/2023 9:50 AM CST]

Search "Cookie HTTP header" online.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cookie>

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>

The Cookie HTTP request header contains stored HTTP cookies associated with the server (i.e. previously sent by the server with the Set-Cookie header or set in JavaScript using `Document.cookie`).

Syntax

Cookie: <cookie-list>

Cookie: name=value

Cookie: name=value; name2=value2; name3=value3

3. Jest React frontend testing automation

[4/18/2023 11:19 AM CST]

jest-frontend-test.js

3.1. Jest snapshot testing error troubleshooting

[5/2/2023 4:21 PM CST]

3.1.1. CLI error

[5/2/2023 4:21 PM CST]

```
npm test
```

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

```
FAIL ./index.test.js
```

- Test suite failed to run

The error below may be caused by using the wrong test environment, see <https://jestjs.io/docs/configuration#testenvironment-string>.

Consider using the "jsdom" test environment.

ReferenceError: window is not defined

```
> 1 | window.frontendServerUrl = 'http://localhost';
    | ^
    2 | window.backendServerUrl = 'https://localhost:8888';
```

```
at Object.window (ServerUrl.js:1:1)
at Object.require (index.js:11:1)
at Object.require (index.test.js:2:1)
```

Test Suites: 1 failed, 1 total

Tests: 0 total

Snapshots: 0 total

Time: 2.275 s

Ran all test suites.

3.1.2. Jest testing environments

[5/2/2023 4:23 PM CST]

<https://jestjs.io/docs/configuration#testenvironment-string>

3.1.3. CLI error 2

[5/2/2023 4:26 PM CST]

```
npm test
```

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

```
FAIL ./index.test.js
```

- Test suite failed to run

- Validation Error:

Test environment jest-environment-jsdom cannot be found. Make sure the testEnvironment configuration option points to an existing node module.

Configuration Documentation:
<https://jestjs.io/docs/configuration>

As of Jest 28 "jest-environment-jsdom" is no longer shipped by default, make sure to install it separately.

Test environment jest-environment-jsdom cannot be found. Make sure the testEnvironment configuration option points to an existing node module.

Configuration Documentation:
<https://jestjs.io/docs/configuration>

As of Jest 28 "jest-environment-jsdom" is no longer shipped by default, make sure to install it separately.

Test Suites: 1 failed, 1 total
Tests: 0 total
Snapshots: 0 total
Time: 0.098 s
Ran all test suites.

3.1.4. jest-environment-jsdom installation

```
[5/2/2023 4:27 PM CST]
npm install --save-dev jest babel-jest @babel/preset-env
@babel/preset-react react-test-renderer jest-environment-jsdom
```

3.1.5. CLI error 3

[5/2/2023 4:28 PM CST]

```
npm test
```

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

```
FAIL ./index.test.js
● Test suite failed to run
```

ReferenceError: React is not defined

```
> 1 | export var footer = React.createElement(
    |                               ^
    2 |   React.Fragment,
    3 |   null,
    4 |   React.createElement(
```

```
    at Object.React (footer.js:1:21)
    at Object.require (index.js:14:1)
    at Object.require (index.test.js:6:1)
```

Test Suites: 1 failed, 1 total

Tests: 0 total

Snapshots: 0 total

Time: 4.48 s

Ran all test suites.

3.1.6. Importing React library in Jest snapshot test file

[5/2/2023 4:30 PM CST]

Search "JavaScript can you import from URL?" online.

<https://tutorials.yax.com/articles/javascript/import/index.html>

<https://stackoverflow.com/questions/34607252/es6-import-module-from-url>

Download and save the React development library files in the React project directory. Then, place the following at the top of the Jest snapshot test file.

```
import * as React from './react.development.js';
import * as ReactDOM from './react-dom.development.js';
```

<https://legacy.reactjs.org/docs/react-dom.html>

The same error still produced. Try placing the following at the top of index.js.

The same error still produced. Try placing the following at the top of footer.js.

3.1.7. CLI error 4

[5/2/2023 4:46 PM CST]

npm test

```
> Allen Young's Stockmarket Demo@1.0.0 test
```

```
> jest
```

```
FAIL ./index.test.js
```

- Test suite failed to run

```
TypeError: Cannot read properties of undefined (reading 'unstable_cancelCallback')
```

```
4316 | var ReactInternals =
React. __SECRET_INTERNALS_DO_NOT_USE_OR_YOU_WILL_BE_FIRED;
4317 | var _ReactInternals$Sched = ReactInternals.Scheduler,
> 4318 |   unstable_cancelCallback = _ReactInternals$Sched.unstable_cancelCallback,
      |                                             ^
4319 |   unstable_now = _ReactInternals$Sched.unstable_now,
4320 |   unstable_scheduleCallback = _ReactInternals$Sched.unstable_scheduleCallback,
4321 |   unstable_shouldYield = _ReactInternals$Sched.unstable_shouldYield,
```

```
at unstable_cancelCallback (react-dom.development.js:4318:55)
at factory (react-dom.development.js:11:66)
at Object.<anonymous> (react-dom.development.js:14:2)
at Object.require (footer.js:2:1)
at Object.require (index.js:14:1)
at Object.require (index.test.js:6:1)
```

```
Test Suites: 1 failed, 1 total
```

```
Tests: 0 total
```

```
Snapshots: 0 total
```

```
Time: 6.531 s
```

```
Ran all test suites.
```

3.1.8. React test renderer

[5/2/2023 4:49 PM CST]

<https://legacy.reactjs.org/docs/test-renderer.html>

<https://github.com/tmpvar/jsdom>

<https://jestjs.io/docs/en/snapshot-testing>

Search "React TestRenderer.create options" online.

<https://github.com/reactjs/react.dev/issues/1809>

<https://stackoverflow.com/questions/48129799/react-test-renderer-how-to-create-a-component-with-context>

(nothing useful.)

3.1.9. Downloading and using production React library files

[5/2/2023 4:47 PM CST]

<https://legacy.reactjs.org/docs/add-react-to-a-website.html#tip-minify-javascript-for-production>

https://unpkg.com/react@18/umd/react.production.min.js
https://unpkg.com/react-dom@18/umd/react-dom.production.min.js

3.1.10. CLI error 5

[5/2/2023 5:00 PM CST]

npm test

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

FAIL ./index.test.js

- Test suite failed to run

TypeError: Cannot read properties of undefined (reading 'unstable_scheduleCallback')

```
221 | borderImageSlice:!0,borderImageWidth:!0,boxFlex:!0,boxFlexGroup:!
0,boxOrdinalGroup:!0,columnCount:!0,columns:!0,flex:!0,flexGrow:!0,flexPositive:!
0,flexShrink:!0,flexNegative:!0,flexOrder:!0,gridArea:!0,gridRow:!0,gridRowEnd:!
0,gridRowSpan:!0,gridRowStart:!0,gridColumn:!0,gridColumnEnd:!0,gridColumnSpan:!
0,gridColumnStart:!0,fontWeight:!0,lineClamp:!0,lineHeight:!0,opacity:!0,order:!0,orphans:!
0,tabSize:!0,widows:!0,zIndex:!0,zoom:!0,fillOpacity:!0,floodOpacity:!0,stopOpacity:!
0,strokeDasharray:!0,
222 | strokeDashoffset:!0,strokeMiterlimit:!0,strokeOpacity:!0,strokeWidth:!
0},al=["Webkit","ms","Moz","O"];Object.keys(dc).forEach(function(a){al.forEach(function(b)
{b=b+a.charAt(0).toUpperCase()+a.substring(1);dc[b]=dc[a]}));var jj=E({menuItem:!0},{area:!
0,base:!0,br:!0,col:!0,embed:!0,hr:!0,img:!0,input:!0,keygen:!0,link:!0,meta:!0,param:!0,source:!
0,track:!0,wbr:!0}),ze=null,se=null,Eb=null,Fb=null,xg=function(a,b){return a(b)},yg=function()
{{,te=!1,Oe=!1;if(Ia)try{var Lc={};Object.defineProperty(Lc,
> 223 | "passive",{get:function(){Oe=!
0}}});window.addEventListener("test",Lc,Lc);window.removeEventListener("test",Lc,Lc)}catch(
a){Oe=!1}var lj=function(a,b,c,d,e,f,g,h,k){var
m=Array.prototype.slice.call(arguments,3);try{b.apply(c,m)}catch(t){this.onError(t)}},gc=!
1,Sc=null,Tc=!1,ue=null,mj={onError:function(a){gc=!
0;Sc=a}},Ba=mb.__SECRET_INTERNALS_DO_NOT_USE_OR_YOU_WILL_BE_FIRED.Sc
heduler,Jg=Ba.unstable_scheduleCallback,Kg=Ba.unstable_NormalPriority,xh=Jg,Li=Ba.unstabl
e_cancelCallback,Qk=Ba.unstable_shouldYield,
^
224 |
Tk=Ba.unstable_requestPaint,P=Ba.unstable_now,Ej=Ba.unstable_getCurrentPriorityLevel,De=
Ba.unstable_ImmediatePriority,Mg=Ba.unstable_UserBlockingPriority,ad=Kg,Fj=Ba.unstable_L
```

```
owPriority,Ng=Ba.unstable_IdlePriority,Uc=null,Ca=null,ta=Math.clz32?
Math.clz32:qj,rj=Math.log,sj=Math.LN2,Wc=64,Rd=4194304,z=0,Ae=!
1,Yc=[],Va=null,Wa=null,Xa=null,jc=new Map,kc=new Map,Ya=[],Cj="mousedown mouseup
touchcancel touchend touchstart auxclick dblclick pointercancel pointerdown pointerup dragend
dragstart drop compositionend compositionstart keydown keypress keyup input textInput copy
cut paste click change contextmenu reset submit".split(" "),
  225 | Gb=Sa.ReactCurrentBatchConfig,Zc=
0,$c=null,Za=null,Ee=null,bd=null,Yb={eventPhase:0,bubbles:0,cancelable:0,timeStamp:functio
n(a){return a.timeStamp||Date.now()}},defaultPrevented:0,isTrusted:0},He=ka(Yb),Mc=E({},Yb,
{view:0,detail:0}),bk=ka(Mc),ag,bg,Nc,Yd=E({},Mc,
{screenX:0,screenY:0,clientX:0,clientY:0,pageX:0,pageY:0,ctrlKey:0,shiftKey:0,altKey:0,meta
Key:0,getModifierState:Fe,button:0,buttons:0,relatedTarget:function(a){return void
0===a.relatedTarget?a.fromElement===a.srcElement?a.toElement:a.fromElement:
  226 | a.relatedTarget},movementX:function(a){if("movementX"in a)return a.movementX;a!
==Nc&&(Nc&&"mousemove"===a.type?(ag=a.screenX-Nc.screenX,bg=a.screenY-
Nc.screenY):bg=ag=0,Nc=a);return ag},movementY:function(a){return"movementY"in a?
a.movementY:bg}),ih=ka(Yd),bl=E({},Yd,{dataTransfer:0}),Xj=ka(bl),cl=E({},Mc,
{relatedTarget:0}),Pe=ka(cl),dl=E({},Yb,
{animationName:0,elapsedTime:0,pseudoElement:0}),Zj=ka(dl),el=E({},Yb,
{clipboardData:function(a){return"clipboardData"in a?
a.clipboardData>window.clipboardData}}),

  at unstable_scheduleCallback (react-dom.production.min.js:223:393)
  at mb (react-dom.production.min.js:13:85)
  at react-dom.production.min.js:13:222
  at Object.<anonymous> (react-dom.production.min.js:267:2)
  at Object.require (footer.js:2:1)
  at Object.require (index.js:14:1)
  at Object.require (index.test.js:6:1)
```

```
Test Suites: 1 failed, 1 total
Tests:      0 total
Snapshots:  0 total
Time:       5.083 s
Ran all test suites.
```

3.1.11. Installing React and ReactDOM npm modules in React web app project directory

[5/2/2023 5:00 PM CST]

Search "How to install React and ReactDOM npm modules" online.

<https://stackoverflow.com/questions/60856500/how-to-install-react-and-reactdom>

<https://www.npmjs.com/package/react-dom>

```
npm install react react-dom
```


Try the following.

```
import React from "react";  
import ReactDOM from "react-dom";  
import ReactDOM from "react-dom/client";
```

(see the error below, CLI error 6.)

3.1.12. CLI error 6

[5/2/2023 5:10 PM CST]

npm test

```
> Allen Young's Stockmarket Demo@1.0.0 test  
> jest
```

console.error

Warning: You are importing createRoot from "react-dom" which is not supported. You should instead import it from "react-dom/client".

```
298 |  
299 | var domNodeDynamicContents = document.getElementById('dynamic-contents');  
> 300 | var rootDynamicContents = ReactDOM.createRoot(domNodeDynamicContents);  
    |                                     ^  
301 | rootDynamicContents.render(React.createElement(DynamicContents, null));  
302 |  
303 | /*  
*****  
*****
```

```
    at printWarning (node_modules/react-dom/cjs/react-dom.development.js:86:30)  
    at error (node_modules/react-dom/cjs/react-dom.development.js:60:7)  
    at Object.createRoot$1 [as createRoot] (node_modules/react-dom/cjs/react-dom.development.js:29796:7)  
    at Object.createRoot (index.js:300:36)  
    at Object.require (index.test.js:6:1)
```

FAIL ./index.test.js

- Test suite failed to run

createRoot(...): Target container is not a DOM element.

```
298 |  
299 | var domNodeDynamicContents = document.getElementById('dynamic-contents');  
> 300 | var rootDynamicContents = ReactDOM.createRoot(domNodeDynamicContents);  
    |                                     ^  
301 | rootDynamicContents.render(React.createElement(DynamicContents, null));
```

```
302 |  
303 | /*  
*****  
*****
```

```
    at createRoot (node_modules/react-dom/cjs/react-dom.development.js:29345:11)  
    at Object.createRoot$1 [as createRoot] (node_modules/react-dom/cjs/react-dom.development.js:29800:10)  
    at Object.createRoot (index.js:300:36)  
    at Object.require (index.test.js:6:1)
```

Test Suites: 1 failed, 1 total
Tests: 0 total
Snapshots: 0 total
Time: 3.157 s
Ran all test suites.

3.1.13. CLI error 7

[5/2/2023 5:12 PM CST]

npm test

```
> Allen Young's Stockmarket Demo@1.0.0 test  
> jest
```

FAIL ./index.test.js

- Test suite failed to run

createRoot(...): Target container is not a DOM element.

```
298 |  
299 | var domNodeDynamicContents = document.getElementById('dynamic-contents');  
> 300 | var rootDynamicContents = ReactDOM.createRoot(domNodeDynamicContents);  
    |                                     ^  
301 | rootDynamicContents.render(React.createElement(DynamicContents, null));  
302 |  
303 | /*  
*****  
*****
```

```
    at createRoot (node_modules/react-dom/cjs/react-dom.development.js:29345:11)  
    at Object.createRoot$1 [as createRoot] (node_modules/react-dom/cjs/react-dom.development.js:29800:10)  
    at Object.<anonymous>.exports.createRoot (node_modules/react-dom/client.js:12:16)  
    at Object.createRoot (index.js:300:36)  
    at Object.require (index.test.js:6:1)
```

Test Suites: 1 failed, 1 total
Tests: 0 total
Snapshots: 0 total
Time: 3.036 s
Ran all test suites.

3.1.14. CLI error 7 troubleshooting

[5/2/2023 5:14 PM CST]

Search "React Jest Test suite failed to run createRoot(...): Target container is not a DOM element" online.

<https://stackoverflow.com/questions/71732421/createroot-target-container-is-not-a-dom-element-in-react-test-file>

<https://bobbyhadz.com/blog/react-createroot-target-container-not-dom-element>

I've commented out the following code in index.js, for Jest testing.

```
const domNodeDynamicContents = document.getElementById('dynamic-contents');  
const rootDynamicContents = ReactDOM.createRoot(domNodeDynamicContents);  
rootDynamicContents.render(<DynamicContents />);
```

And the Jest snapshot test passed with the following CLI output.

```
-----  
npm test  
  
> Allen Young's Stockmarket Demo@1.0.0 test  
> jest  
  
console.log  
  DynamicContents() cookies on client: .  
  
    at log (index.js:118:11)  
  
console.log  
  RegisterLoginOrMyAccountAdminLogout() start.  
  
    at log (index.js:313:11)  
  
console.log  
  RegisterLoginOrMyAccountAdminLogout() loginState prop value: .  
  
    at log (index.js:314:11)  
  
PASS ./index.test.js  
  ✓ changes the class when hovered (124 ms)
```

› 1 snapshot written.

Snapshot Summary

› 1 snapshot written from 1 test suite.

Test Suites: 1 passed, 1 total

Tests: 1 passed, 1 total

Snapshots: 1 written, 1 total

Time: 3.108 s

Ran all test suites.

This tells me that the import and the above code execution must be done only during testing. Exporting a module level variable named 'allenYoungStockmarketHomepageTest' that is set to false by default, and importing the React and ReactDOM modules and executing the above code only when the test variable is set to true, is one way to handle the issue.

3.1.15. JavaScript how to import only when module is imported

[5/2/2023 5:30 PM CST]

Search "JavaScript how to import only when module is imported" online.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/import>

<https://mitya.uk/articles/javascript-modules-conditional-import>

Search "JavaScript conditional import" online.

<https://stackoverflow.com/questions/36367532/how-can-i-conditionally-import-an-es6-module>

Conditional import can be used with 'await' keyword, but it's a no-no in a React app js file.

What should I do to incorporate Jest snapshot and DOM testing on React app js files?

How about trying the following?

```
export var allenYoungStockmarketHomepageTest = false;

if (allenYoungStockmarketHomepageTest) {
  const React = await import('react');
  const ReactDOM = await import('react-dom/client');
}
```

Try, and see what happens.

No, no, no, no, no. Changing the value of allenYoungStockmarketHomepageTest will not rerun the import code. Something else must be done.

I think the React component rendering code should be in HTML script element, for accommodating Jest snapshot and DOM testing. Write this down in the React Jest snapshot and DOM testing paradigm spec as a separate sub-section, and publish it at AllenYoung.dev.

So, my question is, how can I avoid using Webpack in React app development and deployment, and forgo packaging React and React DOM libraries into the deployment package (which seems like an unnecessary inconvenience and additional step to me at this point)?

I think my answer is the following.

In the React app JavaScript file, declare React and ReactDOM variables set to null initially, declare an exported function for conditional import of React and ReactDOM without await keyword, declare another exported function for checking the values of React and ReactDOM variables, then in the Jest test file, call the exported conditional import function, then using the synchronous sleep function, wait until both React and ReactDOM variables are no longer null, then do the React component rendering for test.

In the footer React JSX JS file, do conditional import as in <https://mitya.uk/articles/javascript-modules-conditional-import>, using .then, to create the footer React element.

It's wild, but try it and see what happens.

Actually, better yet, in the Jest test file, create global variables React and ReactDOM that can be accessed in all the modules. This is what's done when the React library files are imported in the HTML script elements. So, do this. This is much cleaner and faster! Doing this requires only taking the rendering code in React JS file to HTML script element; no acrobatic, circus JavaScript code for importing the React and ReactDOM libraries! Easy and fast!

Try using global or globalThis in the Jest test file.

https://developer.mozilla.org/en-US/docs/Glossary/Global_object

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/globalThis

UPDATE [5/2/2023 7:01 PM CST]

`if (typeof React !== 'undefined') { ... }` cannot be used. Visual Studio Code says `export` modifier cannot appear in that statement. What to do now?

[5/2/2023 7:05 PM CST] HERE'S ONE SOLUTION

Declare the footer as a React component function.

~~Implement the React and ReactDOM library import function with synchronous wait in a separate Node.js file.~~

...

3.1.15.1. Try 1 (fail)

[5/2/2023 6:50 PM CST]

`globalThis.React = React;`

```
globalThis.ReactDOM = ReactDOM;
```

declaring the above in the Jest test JS file generates the following error.

```
npm test
```

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

```
FAIL ./index.test.js
```

- Test suite failed to run

```
ReferenceError: React is not defined
```

```
  2 | //import ReactDOM from "react-dom/client";
  3 |
>  4 | export var footer = React.createElement(
    |                        ^
  5 |   React.Fragment,
  6 |   null,
  7 |   React.createElement(
```

```
    at Object.React (footer.js:4:21)
    at Object.require (index.js:14:1)
    at Object.require (index.test.js:9:1)
```

```
Test Suites: 1 failed, 1 total
```

```
Tests:      0 total
```

```
Snapshots:  0 total
```

```
Time:       3.137 s
```

```
Ran all test suites.
```

3.1.15.2. Try 2 (fail)

```
[5/2/2023 6:52 PM CST]
```

```
globalThis.React = await import("react");
```

```
globalThis.ReactDOM = await import("react-dom/client");
```

declaring the above in the Jest test JS file generates the following error.

```
npm test
```

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

FAIL ./index.test.js

- Test suite failed to run

Jest encountered an unexpected token

Jest failed to parse a file. This happens e.g. when your code or its dependencies use non-standard JavaScript syntax, or when Jest is not configured to support such syntax.

Out of the box Jest supports Babel, which will be used to transform your files into valid JS based on your Babel configuration.

By default "node_modules" folder is ignored by transformers.

Here's what you can do:

- If you are trying to use ECMAScript Modules, see <https://jestjs.io/docs/ecmascript-modules> for how to enable it.
- If you are trying to use TypeScript, see <https://jestjs.io/docs/getting-started#using-typescript>
- To have some of your "node_modules" files transformed, you can specify a custom "transformIgnorePatterns" in your config.
- If you need a custom transformation specify a "transform" option in your config.
- If you simply want to mock your non-JS modules (e.g. binary assets) you can stub them out with the "moduleNameMapper" config option.

You'll find more details and examples of these config options in the docs:

<https://jestjs.io/docs/configuration>

For information about custom transformations, see:

<https://jestjs.io/docs/code-transformation>

Details:

```
C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003
AAM Stockmarket\demo files\frontend\React\index.test.js:13
  globalThis.React = await Promise.resolve().then(function () {
                        ^^^^^^
```

SyntaxError: await is only valid in async functions and the top level bodies of modules

at Runtime.createScriptFromCode (node_modules/jest-runtime/build/index.js:1495:14)

Test Suites: 1 failed, 1 total

Tests: 0 total

Snapshots: 0 total

Time: 2.581 s

Ran all test suites.

3.1.15.3. Try 3 (fail)

[5/2/2023 6:54 PM CST]

```
it('changes the class when hovered', async () => {
  globalThis.React = await import("react");
  globalThis.ReactDOM = await import("react-dom/client");

  const component = renderer.create(
    <DynamicContents />,
  );
  let tree = component.toJSON();
  expect(tree).toMatchSnapshot();
});
```

declaring the above in the Jest test JS file generates the following error.

npm test

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

FAIL ./index.test.js

- Test suite failed to run

ReferenceError: React is not defined

```

  2 | //import ReactDOM from "react-dom/client";
  3 |
> 4 | export var footer = React.createElement(
    |                        ^
  5 |   React.Fragment,
  6 |   null,
  7 |   React.createElement(
```

```

    at Object.React (footer.js:4:21)
    at Object.require (index.js:14:1)
    at Object.require (index.test.js:9:1)
```

Test Suites: 1 failed, 1 total

Tests: 0 total

Snapshots: 0 total

Time: 2.968 s

Ran all test suites.

3.1.15.4. Try 4 (fail)

[5/2/2023 6:54 PM CST]


```
it('changes the class when hovered', async () => {
  global.React = await import("react");
  global.ReactDOM = await import("react-dom/client");

  const component = renderer.create(
    <DynamicContents />,
  );
  let tree = component.toJSON();
  expect(tree).toMatchSnapshot();
});
```

declaring the above in the Jest test JS file generates the following error.

npm test

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

FAIL ./index.test.js

- Test suite failed to run

ReferenceError: React is not defined

```
2 | //import ReactDOM from "react-dom/client";
3 |
> 4 | export var footer = React.createElement(
    |                        ^
5 |   React.Fragment,
6 |   null,
7 |   React.createElement(
```

```
at Object.React (footer.js:4:21)
at Object.require (index.js:14:1)
at Object.require (index.test.js:9:1)
```

Test Suites: 1 failed, 1 total

Tests: 0 total

Snapshots: 0 total

Time: 2.968 s

Ran all test suites.

3.1.15.5. Try 5 (fail)

[5/2/2023 6:54 PM CST]

```
it('changes the class when hovered', async () => {
  self.React = await import("react");
  self.ReactDOM = await import("react-dom/client");
```

```
const component = renderer.create(
  <DynamicContents />,
);
let tree = component.toJSON();
expect(tree).toMatchSnapshot();
});
```

declaring the above in the Jest test JS file generates the following error.

```
npm test
```

```
> Allen Young's Stockmarket Demo@1.0.0 test
> jest
```

```
FAIL ./index.test.js
  ● Test suite failed to run
```

ReferenceError: React is not defined

```
2 | //import ReactDOM from "react-dom/client";
3 |
> 4 | export var footer = React.createElement(
    |           ^
5 |   React.Fragment,
6 |   null,
7 |   React.createElement(
```

```
at Object.React (footer.js:4:21)
at Object.require (index.js:14:1)
at Object.require (index.test.js:9:1)
```

Test Suites: 1 failed, 1 total
Tests: 0 total
Snapshots: 0 total
Time: 2.968 s
Ran all test suites.

3.1.15.6. Try 6 (success!)

[5/2/2023 7:23 PM CST]

Wow, I create the code that works!

Declaring the following in the Jest snapshot test code at the module level at the top below the import statements, and not declaring any import statements in index.js and footer.js, successfully executes the jest snapshot test code.

```
if (typeof React === 'undefined') {
  console.log(`Importing React...`)
  import("react").then(exports => {
    //do something with @exports...
    console.log(`typeof exports:  ${typeof exports}.`);
    globalThis.React = exports;
    console.log(`typeof React after import:  ${typeof
React}.`);
  });
}

if (typeof ReactDOM === 'undefined') {
  import("react-dom/client").then(exports => {
    //do something with @exports...
    globalThis.ReactDOM = exports;
  });
}
```

I don't like the above code, because it is asynchronous and does not wait for the import to complete.

3.1.15.7. Try 7 (fail)

Declaring the following in the Jest snapshot test code at the module level at the top below the import statements, generates error.

```
if (typeof React === 'undefined') {
  globalThis.React = await import("react");
}
if (typeof ReactDOM === 'undefined') {
  globalThis.ReactDOM = await import("react-dom/client");
}
//the above creates error.  SyntaxError: await is only valid in
async functions and the top level bodies of modules
```

<https://www.javascripttutorial.net/javascript-top-level-await/>

3.1.15.8. Try 6 (success) (use this. the most proper [synchronous] code.)

[5/2/2023 7:41 PM CST]

Declaring the following in the Jest snapshot test code at the module level at the top below the import statements, and not declaring any import statements in index.js and footer.js, successfully executes the jest snapshot test code.

```
(async () => {  
  if (typeof React === 'undefined') {  
    globalThis.React = await import("react");  
  }  
  if (typeof ReactDOM === 'undefined') {  
    globalThis.ReactDOM = await import("react-dom/client");  
  }  
}) ();
```

The above code waits for import to complete.

<https://www.javascripttutorial.net/javascript-top-level-await/>

3.1.16. Can JavaScript module detect when it is being imported?

[5/2/2023 5:50 PM CST]

Search "Can JavaScript module detect when it is being imported?" online.

<https://stackoverflow.com/questions/61790963/how-to-determine-if-javascript-module-was-imported-or-loaded-via-script-src>

<https://stackoverflow.com/questions/61797713/how-to-programmatically-determine-within-the-module-if-javascript-module-was-l>

It doesn't seem to be possible.

3.1.17. JavaScript how to check if a variable is defined?

[5/2/2023 6:01 PM CST]

Search "JavaScript how to check if a variable is defined?" online.

<https://stackoverflow.com/questions/5113374/javascript-check-if-variable-exists-is-defined-initialized>

```
if (typeof variable !== 'undefined') {  
  // the variable is defined  
}
```

3.1.18. JavaScript how to create global variable that can be accessed in all modules

[5/2/2023 6:28 PM CST]

Search "JavaScript how to create global variable that can be accessed in all modules" online.

<https://medium.com/codex/global-variables-and-javascript-modules-ce674a869164>

<https://www.edureka.co/community/52900/how-do-i-share-global-variables-across-modules-python>

<https://stackoverflow.com/questions/66420688/how-do-i-create-global-variables-i-can-use-across-my-modules>

<https://general.support.brightcove.com/developer/concepts-javascript-module-design-pattern.html>

<https://codedamn.com/news/javascript/variables>

Nothing useful for my need.

3.1.19. JavaScript global object

[5/2/2023 6:34 PM CST]

Search "JavaScript global object" online.

https://developer.mozilla.org/en-US/docs/Glossary/Global_object

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/globalThis

3.1.20. JavaScript import to global variable

[5/2/2023 6:39 PM CST]

Search "JavaScript import to global variable" online.

<https://stackoverflow.com/questions/52427167/access-a-global-variable-in-main-file-with-an-imported-javascript-function-es6>

<https://medium.com/codex/global-variables-and-javascript-modules-ce674a869164>

3.1.21. JavaScript import to data variable

[5/2/2023 6:44 PM CST]

Search "JavaScript import to data variable" online.

<https://stackoverflow.com/questions/29168433/how-to-perform-a-variable-es6-import>

Search "JavaScript import Mozilla" online.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import>

3.1.22. JavaScript what is top level body of module

[5/2/2023 7:31 PM CST]

Search "JavaScript what is top level body of module" online.

<https://www.javascripttutorial.net/javascript-top-level-await/>

3.1.23. How to call exported JavaScript function in HTML

[5/2/2023 8:10 PM CST]

Search "How to call exported JavaScript function in HTML" online.

<https://stackoverflow.com/questions/50176213/accessing-exported-functions-from-html-file>

3.2. React frontend Jest snapshot and DOM unit test cases

([5/4/2023 4:27 PM CST] these look like true unit test cases! I LOVE these!)

3.2.1. index.js Jest snapshot and DOM unit test cases

- index.js Jest snapshot unit test cases (i.e. different webpage rendering test cases)
 - No-login mock Jest snapshot test
 - Non-admin-user-login mock Jest snapshot test
 - Admin-user-login mock Jest snapshot test
-
- index.js Jest DOM unit test cases (i.e. different user-input event handling test cases)
 - Number-of-Company-shares-to-buy input entering Jest DOM tests
 - Proceed-to-checkout button click Jest DOM test with button-click event handler function mocking (mock only the fetch() call portion that should reside in a separate module for mocking)

3.2.2. registration.js Jest snapshot and DOM unit test cases

- registration.js Jest snapshot unit test cases (i.e. different webpage rendering test cases)
 - Jest snapshot test
-
- registration.js Jest DOM unit test cases (i.e. different user-input event handling test cases)
 - No-login Jest DOM test
 - Password show/hide text click Jest DOM test
 - Register button click Jest DOM tests with button-click event handler function mocking (mock only the fetch() call portion that should reside in a separate module for mocking) (registration input validation Jest DOM tests)

3.2.3. EmailAddressVerificationNotice.js Jest snapshot and DOM unit test cases

- EmailAddressVerificationNotice.js Jest snapshot unit test cases (i.e. different webpage rendering test cases)
 - Jest snapshot test (with the email address passed as a param to the top-level React element)
-
- EmailAddressVerificationNotice.js Jest DOM unit test cases (i.e. different user-input event handling test cases)

- "Resend email-address verification email" button click Jest DOM test (do this or not? think about it. ANSWER [5/12/2023 4:25 PM CST] maybe later with backend mocking.)

3.2.4. login.js Jest snapshot and DOM unit test cases

- login.js Jest snapshot unit test cases (i.e. different webpage rendering test cases)
 - No-login mock Jest snapshot test
 - Non-admin-user-login mock Jest snapshot test
 - Admin-user-login mock Jest snapshot test
-
- login.js Jest DOM unit test cases (i.e. different user-input event handling test cases)
 - No-login mock Jest DOM test (for input-control enablement check)
 - Non-admin-user-login mock Jest DOM test (for input-control disablement check)
 - Admin-user-login mock Jest DOM test (for input-control disablement check)
 -
 - Password show/hide text click Jest DOM test
 - Login button click Jest DOM tests with button-click event handler function mocking (mock only the fetch() call portion that should reside in a separate module for mocking) (login input validation Jest DOM tests, email and password format validity checks)

3.2.5. PasswordReset.js Jest snapshot and DOM unit test cases

- PasswordReset.js Jest snapshot unit test cases (i.e. different webpage rendering test cases)
 - Jest snapshot test
-
- PasswordReset.js Jest DOM unit test cases (i.e. different user-input event handling test cases)
 - Password show/hide text click Jest DOM test
 - Password-reset button click Jest DOM tests with button-click event handler function mocking (mock only the fetch() call portion that should reside in a separate module for mocking) (password format validity and retype-match checks)

3.2.6. account.js Jest snapshot and DOM unit test cases

- account.js Jest snapshot unit test cases (i.e. different webpage rendering test cases)
 - No-login mock Jest snapshot test
 - Non-admin-user-login mock Jest snapshot test (with user-transaction list fetch() mock)
 - Admin-user-login mock Jest snapshot test
-
- account.js Jest DOM unit test cases (i.e. different user-input event handling test cases)
 - No-login mock Jest DOM test (for input-control disablement check)

- Non-admin-user-login mock Jest DOM test (for input-control enablement check)
- Admin-user-login mock Jest DOM test (for input-control disablement check)
-
- Password show/hide text click Jest DOM test
- Update button click Jest DOM tests with button-click event handler function mocking (mock only the fetch() call portion that should reside in a separate module for mocking) (email and password input format validity check.)
-
- User-transactions-list-section page number entering Jest DOM test
- User-transactions-list-section page go-to button click Jest DOM test (with fetch() wrapper function mock for returning mock data, and mock data display validation)
- User-transactions-list-section page navigation-link click Jest DOM test (with fetch() wrapper function mock for returning mock data, and mock data display validation)

3.2.7. admin.js Jest snapshot and DOM unit test cases

- admin.js Jest snapshot unit test cases (i.e. different webpage rendering test cases)
 - No-login mock Jest snapshot test
 - Non-admin-user-login mock Jest snapshot test
 - Admin-user-login mock Jest snapshot test (with user and transaction list fetch() mocks)
-
- admin.js Jest DOM unit test cases (i.e. different user-input event handling test cases)
 - No-login mock Jest DOM test (for input-control disablement check)
 - Non-admin-user-login mock Jest DOM test (for input-control disablement check)
 - Admin-user-login mock Jest DOM test (for input-control enablement check)
 -
 - User-list-section page number entering Jest DOM test
 - User-list-section page go-to button click Jest DOM test (with fetch() wrapper function mock for returning mock data, and mock data display validation)
 - User-list-section page navigation-link click Jest DOM test (with fetch() wrapper function mock for returning mock data, and mock data display validation)
 -
 - Transactions-list-section page number entering Jest DOM test
 - Transactions-list-section page go-to button click Jest DOM test (with fetch() wrapper function mock for returning mock data, and mock data display validation)
 - Transactions-list-section page navigation-link click Jest DOM test (with fetch() wrapper function mock for returning mock data, and mock data display validation)

4. Selenium end-to-end testing automation

[4/6/2023 7:16 PM EST] UI I/O testing automation

selenium-e2e-test.py

4.1. Selenium end-to-end test process spec

[5/14/2023 6:43 PM EST]

1. Backend server start (~~in code as an asynchronous execution [can't do this; the Selenium test code is in Python, not JavaScript]~~ by CLI command execution in code in a separate process, wait until server runs by checking stdout)
2. Frontend server start (by CLI command execution in code in a separate process, wait until server runs by checking stdout)
3. Homepage visit
4. Registration link click
5. Account creation
6. Email-address verification status update database CLI command execution
7. Log in, login verification on home page
8. Stock purchase, receipt verification
9. User account page visit and verification
10. Homepage visit
11. Log out and homepage update verification

4.1.1. Original notes

[5/5/2023 9:35 AM CST]

Backend server start (in code), frontend server start (by CLI command execution in code, wait until server runs by checking stdout), homepage visit, registration link click, account creation, email-address verification status update database CLI command execution, log in, stock purchase, user account page visit and verification. (Anything else? Most likely not?)

4.2. Python dependencies to install

[5/16/2023 9:56 AM CST]

```
pip install psutil
pip install requests
pip install selenium
pip install Uenum34
pip install Uenum
```

4.3. CLI commands to execute for Selenium end-to-end testing

[5/15/2023 3:20 PM CST] [5/15/2023 5:00 PM CST]

Windows

```
Python "C:\Users\Allen\Documents\Asian-American Man\
AllenYoung.dev\demo software\00003 AAM Stockmarket\git-
repository\CI_CD\selenium-e2e-test.py" "C:\Users\Allen\Documents\
Asian-American Man\AllenYoung.dev\demo software\00003 AAM
Stockmarket\git-repository\backend\Express.js" "C:\Users\Allen\
Documents\Asian-American Man\AllenYoung.dev\demo software\00003
AAM Stockmarket\git-repository\frontend\React"
```

(The above CLI command doesn't launch or start frontend and backend servers.)

```
Python "C:\Users\Allen\Documents\Asian-American Man\
AllenYoung.dev\demo software\00003 AAM Stockmarket\git-
repository\CI_CD\selenium-e2e-test.py" "C:\Users\Allen\Documents\
Asian-American Man\AllenYoung.dev\demo software\00003 AAM
Stockmarket\git-repository\backend\Express.js" "C:\Users\Allen\
Documents\Asian-American Man\AllenYoung.dev\demo software\00003
AAM Stockmarket\git-repository\frontend\React" True
```

(The above CLI command does launch or start frontend and backend servers.)

4.4. Email-address verification status update database Python commands

[5/16/2023 1:39 PM CST]

Windows

```
backendServerProgramDirectoryFullPath = 'C:\\Users\\Allen\\
Documents\\Asian-American Man\\AllenYoung.dev\\demo software\\
00003 AAM Stockmarket\\demo files\\backend\\Express.js'
```

```
userEmailAddress = 'AllenYoung@AllenYoung.dev'
```

```
databaseFileFullName = 'allen_young_stockmarket.db'
```

```
emailAddressVerificationStatusUpdateCliCommandWindows = 'sqlite3
"' + os.path.join(backendServerProgramDirectoryFullPath,
databaseFileFullName) + '" "UPDATE users SET "'Email address
verified"' = \'true\' WHERE "'Email address"'
= \'{}\';"'.format(userEmailAddress)
```

Linux

4.5. Python commands to execute for launching backend and frontend servers in the Selenium end-to-end test Python program

Windows

```
-----
completedprocessCli = subprocess.run('set DEBUG=* & node
server.js true true', stdout=subprocess.PIPE,
stderr=subprocess.PIPE, encoding='utf-8', errors='utf-8',
cwd="C:\\Users\\Allen\\Documents\\Asian-American Man\\
AllenYoung.dev\\demo software\\00003 AAM Stockmarket\\demo
files\\backend\\Express.js", shell=True)

completedprocessCli = subprocess.run('http-server -p 80',
stdout=subprocess.PIPE, stderr=subprocess.PIPE, encoding='utf-8',
errors='utf-8', cwd="C:\\Users\\Allen\\Documents\\Asian-American
Man\\AllenYoung.dev\\demo software\\00003 AAM Stockmarket\\demo
files\\frontend\\React", shell=True)
```

4.5.1. What doesn't work

4.5.1.1. [5/15/2023 4:53 PM CST]

Windows

```
-----
completedprocessCli = subprocess.run('"C:\\Program Files\\
nodejs\\node.exe" server.js true true', stdout=subprocess.PIPE,
stderr=subprocess.PIPE, encoding='utf-8', errors='utf-8',
env={"DEBUG": ""}, cwd="C:\\Users\\Allen\\Documents\\Asian-
American Man\\AllenYoung.dev\\demo software\\00003 AAM
Stockmarket\\demo files\\backend\\Express.js", shell=True,
timeout=None, check=True)
```

(Likely reason on why the above exits immediately in Python IDLE: Because the subprocess created gets destroyed immediately after the execution due to exiting, so the spawned server process gets terminated also. A thread may have to be used to launch a test web server in a Python program.)

(UPDATE [5/15/2023 4:51 PM CST] The above might not work, because all the environment variables are missing.)

```
completedprocessCli = subprocess.run('http-server -p 80',
stdout=subprocess.PIPE, stderr=subprocess.PIPE, encoding='utf-8',
errors='utf-8', cwd="C:\\Users\\Allen\\Documents\\Asian-American
```

```
Man\\AllenYoung.dev\\demo software\\00003 AAM Stockmarket\\demo
files\\frontend\\React", shell=True, timeout=None, check=True)
```

(The above works. Only the backend server run in Python IDLE quits immediately, not the frontend. Why?)

4.6. Frontend and backend servers start failure troubleshooting

4.6.1. Node.js execution file location in Windows

[5/15/2023 3:55 PM CST]

Search "Node.js execution file location in Windows" online.

4.6.2. Python subprocess run env example

[5/15/2023 4:02 PM CST]

Search "python subprocess run env example" online.

<https://stackoverflow.com/questions/2231227/python-subprocess-popen-with-a-modified-environment>

4.6.3. Python multiprocessing subprocess run failure

[5/15/2023 5:09 PM CST]

Search "Python multiprocessing subprocess run failure OSError: [WinError 87] The parameter is incorrect" online.

<https://stackoverflow.com/questions/32379953/oserror-winerror87-the-parameter-is-incorrect>

4.6.4. PermissionError: [WinError 5] Access is denied

[5/15/2023 5:16 PM CST]

Search "PermissionError: [WinError 5] Access is denied" online.

<https://techisours.com/winerror-5-access-is-denied/>

[https://learn.microsoft.com/en-us/answers/questions/1165616/vs-code-and-python-error-permissionerror-\(winerror](https://learn.microsoft.com/en-us/answers/questions/1165616/vs-code-and-python-error-permissionerror-(winerror)

NOTE [5/15/2023 5:19 PM CST] This might be caused by the directories being in C:\Users.

Search "Python Windows how to give program access permission to users directory" online.

<https://stackoverflow.com/questions/12168110/how-to-set-folder-permissions-in-windows>

<https://answers.microsoft.com/en-us/windows/forum/all/file-permissions/db8d0750-f4a1-4cd5-885d-ac7927ffd9c5>

<https://www.sololearn.com/Discuss/2042848/solved-python-get-permission-to-write-in-program-files>

Executing the Selenium e2e Python program still produces the following error.

```
C:\windows\system32>cd C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins
```

```
C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins>Python "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py" "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\backend\Express.js" "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\frontend\React"
```

```
backendServerProgramDirectoryFullPath: C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\backend\Express.js
```

```
frontendServerProgramDirectoryFullPath: C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\frontend\React
```

```
Process Process-1:
```

```
Traceback (most recent call last):
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\process.py", line 314, in _bootstrap
```

```
    self.run()
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\process.py", line 108, in run
```

```
    self._target(*self._args, **self._kwargs)
```

```
File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 33, in
```

```
    _LaunchBackendServer
```

```
    with open('tempLog-backend{0}.log'.format(os.getpid(), 'w+')) as f:
```

```
FileNotFoundError: [Errno 2] No such file or directory: 'tempLog-backend8472.log'
```

```
Traceback (most recent call last):
```

```
File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 178, in <module>
```

```
aysset.ExecuteAllenYoungStockmarketSeleniumE2eTest(backendServerProgramDirectoryFullPath,
```

```
File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 84, in
```

```
ExecuteAllenYoungStockmarketSeleniumE2eTest
```

```
    self._frontendServerProcess.start()
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\process.py", line 121, in start
```

```
    self._popen = self._Popen(self)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\context.py", line 224, in _Popen
```

```
    return _default_context.get_context().Process._Popen(process_obj)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\context.py", line 336, in _Popen
```

```
    return Popen(process_obj)
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
popen_spawn_win32.py", line 93, in __init__
    reduction.dump(process_obj, to_child)
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
reduction.py", line 60, in dump
    ForkingPickler(file, protocol).dump(obj)
TypeError: cannot pickle 'weakref.ReferenceType' object
```

C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM
Stockmarket\demo files\CI_CD\Jenkins>Traceback (most recent call last):

```
File "<string>", line 1, in <module>
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
spawn.py", line 102, in spawn_main
    source_process = _winapi.OpenProcess(
OSError: [WinError 87] The parameter is incorrect
```

More errors

C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM
Stockmarket\demo files\CI_CD\JenkinsPython "C:\Users\Allen\Documents\Asian-American
Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\
selenium-e2e-test.py" "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo
software\00003 AAM Stockmarket\demo files\backend\Express.js" "C:\Users\Allen\Documents\
Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\
frontend\React"

backendServerProgramDirectoryFullPath: C:\Users\Allen\Documents\Asian-American Man\
AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\backend\Express.js
frontendServerProgramDirectoryFullPath: C:\Users\Allen\Documents\Asian-American Man\
AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\frontend\React

Process Process-1:

Traceback (most recent call last):

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
process.py", line 314, in _bootstrap
    self.run()
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
process.py", line 108, in run
    self._target(*self._args, **self._kwargs)
File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003  
AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 33, in
_launchBackendServer
    with open('c:\\tempLog-backend{0}.log'.format(os.getpid(), 'w+')) as f:
FileNotFoundError: [Errno 2] No such file or directory: 'c:\\tempLog-backend3776.log'
```

Traceback (most recent call last):

File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 178, in <module>

aysset.ExecuteAllenYoungStockmarketSeleniumE2eTest(backendServerProgramDirectoryFullPath,

File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 84, in

ExecuteAllenYoungStockmarketSeleniumE2eTest

self.frontendServerProcess.start()

File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\process.py", line 121, in start

self._popen = self._Popen(self)

File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\context.py", line 224, in _Popen

return _default_context.get_context().Process._Popen(process_obj)

File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\context.py", line 336, in _Popen

return Popen(process_obj)

File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\popen_spawn_win32.py", line 93, in __init__

reduction.dump(process_obj, to_child)

File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\reduction.py", line 60, in dump

ForkingPickler(file, protocol).dump(obj)

TypeError: cannot pickle 'weakref.ReferenceType' object

C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins>Traceback (most recent call last):

File "<string>", line 1, in <module>

File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\spawn.py", line 107, in spawn_main

new_handle = reduction.duplicate(pipe_handle,

File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\reduction.py", line 79, in duplicate

return _winapi.DuplicateHandle(

PermissionError: [WinError 5] Access is denied

Search "Python create file for write if it doesn't exist" online.

<https://stackoverflow.com/questions/2967194/open-in-python-does-not-create-a-file-if-it-doesnt-exist>

Stupid typo in cause caused "FileNotFoundError: [Errno 2] No such file or directory: 'c:\\tempLog-backend3776.log'". Now fixed.

Still error.

```
C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM
Stockmarket\demo files\CI_CD\JenkinsPython "C:\Users\Allen\Documents\Asian-American
Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\
selenium-e2e-test.py" "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo
software\00003 AAM Stockmarket\demo files\backend\Express.js" "C:\Users\Allen\Documents\
Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\
frontend\React"
```

```
backendServerProgramDirectoryFullPath: C:\Users\Allen\Documents\Asian-American Man\
AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\backend\Express.js
```

```
frontendServerProgramDirectoryFullPath: C:\Users\Allen\Documents\Asian-American Man\
AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\frontend\React
```

Traceback (most recent call last):

```
File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003
AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 178, in <module>
```

```
aysset.ExecuteAllenYoungStockmarketSeleniumE2eTest(backendServerProgramDirectoryFullP
ath,
```

```
File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003
AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 84, in
```

```
ExecuteAllenYoungStockmarketSeleniumE2eTest
```

```
self._frontendServerProcess.start()
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
process.py", line 121, in start
```

```
self._popen = self._Popen(self)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
context.py", line 224, in _Popen
```

```
return _default_context.get_context().Process._Popen(process_obj)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
context.py", line 336, in _Popen
```

```
return Popen(process_obj)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
popen_spawn_win32.py", line 93, in __init__
```

```
reduction.dump(process_obj, to_child)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
reduction.py", line 60, in dump
```

```
ForkingPickler(file, protocol).dump(obj)
```

```
TypeError: cannot pickle 'weakref.ReferenceType' object
```

```
C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM
Stockmarket\demo files\CI_CD\Jenkins>Traceback (most recent call last):
```

```
File "<string>", line 1, in <module>
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
spawn.py", line 102, in spawn_main
```

```
source_process = _winapi.OpenProcess(
```

```
OSError: [WinError 87] The parameter is incorrect
```


tempLog-backend9176.log is created, however.

More than two arguments in process.argv. process.argv[2]: true. Server start condition met.
corsServer: true.

Mon, 15 May 2023 22:28:09 GMT express:application set "x-powered-by" to true

Mon, 15 May 2023 22:28:09 GMT express:application set "etag" to 'weak'

Mon, 15 May 2023 22:28:09 GMT express:application set "etag fn" to [Function: generateETag]

Mon, 15 May 2023 22:28:09 GMT express:application set "env" to 'development'

Mon, 15 May 2023 22:28:09 GMT express:application set "query parser" to 'extended'

Mon, 15 May 2023 22:28:09 GMT express:application set "query parser fn" to [Function: parseExtendedQueryString]

Mon, 15 May 2023 22:28:09 GMT express:application set "subdomain offset" to 2

Mon, 15 May 2023 22:28:09 GMT express:application set "trust proxy" to false

Mon, 15 May 2023 22:28:09 GMT express:application set "trust proxy fn" to [Function: trustNone]

Mon, 15 May 2023 22:28:09 GMT express:application booting in development mode

Mon, 15 May 2023 22:28:09 GMT express:application set "view" to [Function: View]

Mon, 15 May 2023 22:28:09 GMT express:application set "views" to 'C:\\Users\\Allen\\Documents\\Asian-American Man\\AllenYoung.dev\\demo software\\00003 AAM Stockmarket\\demo files\\backend\\Express.js\\views'

Mon, 15 May 2023 22:28:09 GMT express:application set "jsonp callback name" to 'callback'

Mon, 15 May 2023 22:28:09 GMT express:application set "view engine" to 'ejs'

Mon, 15 May 2023 22:28:09 GMT express:router use '/' query

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'

Mon, 15 May 2023 22:28:09 GMT express:router use '/' expressInit

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'

Mon, 15 May 2023 22:28:09 GMT express:router use '/' serveStatic

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'

Mon, 15 May 2023 22:28:09 GMT express:router use '/' cookieParser

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'

Mon, 15 May 2023 22:28:09 GMT express:router use '/' jsonParser

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'

Mon, 15 May 2023 22:28:09 GMT express:router use '/' urlencodedParser

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'

Mon, 15 May 2023 22:28:09 GMT express:router use '/' corsMiddleware

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'

Mon, 15 May 2023 22:28:09 GMT express:router:route new '/CheckAccessTokenValidity'

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/CheckAccessTokenValidity'

Mon, 15 May 2023 22:28:09 GMT express:router:route get '/CheckAccessTokenValidity'

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'

Mon, 15 May 2023 22:28:09 GMT express:router:route new '/GetCompanyStockData'

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/GetCompanyStockData'

Mon, 15 May 2023 22:28:09 GMT express:router:route get '/GetCompanyStockData'

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new
'/GetNumberOfCompanySharesOwnedByUser'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new
'/GetNumberOfCompanySharesOwnedByUser'
Mon, 15 May 2023 22:28:09 GMT express:router:route get
'/GetNumberOfCompanySharesOwnedByUser'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/Login'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/Login'
Mon, 15 May 2023 22:28:09 GMT express:router:route post '/Login'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/Logout'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/Logout'
Mon, 15 May 2023 22:28:09 GMT express:router:route delete '/Logout'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/Checkout'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/Checkout'
Mon, 15 May 2023 22:28:09 GMT express:router:route post '/Checkout'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/Checkout/:numberOfSharesToBuy'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/Checkout/:numberOfSharesToBuy'
Mon, 15 May 2023 22:28:09 GMT express:router:route get '/Checkout/:numberOfSharesToBuy'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/Checkout/api/orders'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/Checkout/api/orders'
Mon, 15 May 2023 22:28:09 GMT express:router:route post '/Checkout/api/orders'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new
'/Checkout/api/orders/:orderID/capture'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new
'/Checkout/api/orders/:orderID/capture'
Mon, 15 May 2023 22:28:09 GMT express:router:route post
'/Checkout/api/orders/:orderID/capture'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/Receipt/:orderID'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/Receipt/:orderID'
Mon, 15 May 2023 22:28:09 GMT express:router:route get '/Receipt/:orderID'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/CheckEmailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/CheckEmailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:route get '/CheckEmailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/RegisterUser'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/RegisterUser'
Mon, 15 May 2023 22:28:09 GMT express:router:route post '/RegisterUser'

Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new
'/SendEmailAddressVerificationEmail/:accessToken/:emailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new
'/SendEmailAddressVerificationEmail/:accessToken/:emailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:route get
'/SendEmailAddressVerificationEmail/:accessToken/:emailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new
'/GetEmailAddressVerificationSuccessPage/:accessToken'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new
'/GetEmailAddressVerificationSuccessPage/:accessToken'
Mon, 15 May 2023 22:28:09 GMT express:router:route get
'/GetEmailAddressVerificationSuccessPage/:accessToken'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new
'/SendResetPasswordEmail/:emailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new
'/SendResetPasswordEmail/:emailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:route get
'/SendResetPasswordEmail/:emailAddress'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/ResetPassword'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/ResetPassword'
Mon, 15 May 2023 22:28:09 GMT express:router:route post '/ResetPassword'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/GetUserAccountSettings'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/GetUserAccountSettings'
Mon, 15 May 2023 22:28:09 GMT express:router:route get '/GetUserAccountSettings'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/UpdateUserAccountSettings'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/UpdateUserAccountSettings'
Mon, 15 May 2023 22:28:09 GMT express:router:route post '/UpdateUserAccountSettings'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new
'/GetNumberOfUserStockTransactions'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new
'/GetNumberOfUserStockTransactions'
Mon, 15 May 2023 22:28:09 GMT express:router:route get
'/GetNumberOfUserStockTransactions'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new
'/GetUserStockTransactionHistory/:pageNumber'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new
'/GetUserStockTransactionHistory/:pageNumber'

```
Mon, 15 May 2023 22:28:09 GMT express:router:route get
'/GetUserStockTransactionHistory/:pageNumber'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/GetTotalNumberOfUsers'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/GetTotalNumberOfUsers'
Mon, 15 May 2023 22:28:09 GMT express:router:route get '/GetTotalNumberOfUsers'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/GetUserList/:pageNumber'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/GetUserList/:pageNumber'
Mon, 15 May 2023 22:28:09 GMT express:router:route get '/GetUserList/:pageNumber'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new '/GetTotalNumberOfTransactions'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/GetTotalNumberOfTransactions'
Mon, 15 May 2023 22:28:09 GMT express:router:route get '/GetTotalNumberOfTransactions'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
Mon, 15 May 2023 22:28:09 GMT express:router:route new
'/GetTransactionHistory/:pageNumber'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new
'/GetTransactionHistory/:pageNumber'
Mon, 15 May 2023 22:28:09 GMT express:router:route get
'/GetTransactionHistory/:pageNumber'
Mon, 15 May 2023 22:28:09 GMT express:router:layer new '/'
HTTPS server started.
node:events:491
  throw er; // Unhandled 'error' event
  ^
```

```
Error: listen EADDRINUSE: address already in use :::8888
    at Server.setupListenHandle [as _listen2] (node:net:1432:16)
    at listenInCluster (node:net:1480:12)
    at Server.listen (node:net:1568:7)
    at runServer (file:///C:/Users/Allen/Documents/Asian-American%20Man/AllenYoung.dev/
demo%20software/00003%20AAM%20Stockmarket/demo%20files/backend/Express.js/
server.js:1899:17)
Emitted 'error' event on Server instance at:
    at emitErrorNT (node:net:1459:8)
    at processTicksAndRejections (node:internal/process/task_queues:83:21) {
  code: 'EADDRINUSE',
  errno: -4091,
  syscall: 'listen',
  address: '::',
  port: 8888
}
```

Ok, closing Python IDLE killed the running backend server instance.

Still more error.

```
C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins
```

```
C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\JenkinsPython "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py" "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\backend\Express.js" "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\frontend\React"
```

```
backendServerProgramDirectoryFullPath: C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\backend\Express.js
```

```
frontendServerProgramDirectoryFullPath: C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\frontend\React
```

Traceback (most recent call last):

```
File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 178, in <module>
```

```
aysset.ExecuteAllenYoungStockmarketSeleniumE2eTest(backendServerProgramDirectoryFullPath,
```

```
File "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\CI_CD\Jenkins\selenium-e2e-test.py", line 84, in
```

```
ExecuteAllenYoungStockmarketSeleniumE2eTest
```

```
self._frontendServerProcess.start()
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\process.py", line 121, in start
```

```
self._popen = self._Popen(self)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\context.py", line 224, in _Popen
```

```
return _default_context.get_context().Process._Popen(process_obj)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\context.py", line 336, in _Popen
```

```
return Popen(process_obj)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\popen_spawn_win32.py", line 93, in __init__
```

```
reduction.dump(process_obj, to_child)
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\reduction.py", line 60, in dump
```

```
ForkingPickler(file, protocol).dump(obj)
```

```
TypeError: cannot pickle 'weakref.ReferenceType' object
```

Traceback (most recent call last):

```
File "<string>", line 1, in <module>
```

```
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\spawn.py", line 116, in spawn_main
```

```
exitcode = _main(fd, parent_sentinel)
File "C:\Users\Allen\AppData\Local\Programs\Python\Python310\lib\multiprocessing\
spawn.py", line 126, in _main
    self = reduction.pickle.load(from_parent)
EOFError: Ran out of input
```

Removing 30 in join() call, and trying again.

Actually, removing join() entirely. Using time.sleep(20).

4.6.5. EOFError: Ran out of input

[5/15/2023 5:38 PM CST]

Search "EOFError: Ran out of input" online.

<https://stackoverflow.com/questions/24791987/why-do-i-get-pickle-eoferror-ran-out-of-input-reading-an-empty-file>

Search "Python multiprocessing EOFError: Ran out of input" online.

<https://stackoverflow.com/questions/44704086/eoferror-ran-out-of-input-inside-a-class>

Now, both backend and frontend servers run in a non-admin Windows terminal.

4.7. Frontend server termination failure troubleshooting

Look into why the backend and frontend server processes are still running after calling terminate(), and why they terminate after sending request to them (e.g. via a web browser).

[5/15/2023 6:08 PM CST] It's probably because of the frontend server asking for verification input. (no, it doesn't.) [5/15/2023 6:12 PM CST] UPDATE [5/15/2023 6:30 PM CST] only the frontend server doesn't die when its parent process dies; it dies when the process that created its parent process dies (i.e. the Windows terminal). this probably won't be a problem in CI/CD pipeline CLI execution; but I want to think a bit more about it; I want a clean exit. Document everything. UPDATE [5/16/2023 8:57 AM CST] search "Python multiprocessing subprocess remaining keeping running in Windows terminal" online. UPDATE [5/15/2023 7:33 PM CST] Search "Python how to get all the running subprocesses" online.

Search "Node http-server" online.

<https://www.npmjs.com/package/http-server>

Search "Python multiprocessing send keys to process" online.

<https://stackoverflow.com/questions/51746928/exchanging-keystrokes-based-data-among-multiple-python-processes>

Search "Python send a request to URL" online.

<https://reqbin.com/code/python/ixa9gi7o/python-http-request-example>

Search "Windows terminal list running processes" online.

<https://superuser.com/questions/914782/how-do-you-list-all-processes-on-the-command-line-in-windows>

<https://www.makeuseof.com/windows-save-running-processes-list/>

Search "Python how to get all the running subprocesses" online.

<https://stackoverflow.com/questions/3162096/how-do-you-list-all-child-processes-in-python>

Before frontend and backend servers termination.

Child pid is 11160

Child pid is 19824

Child pid is 14260

Child pid is 20456

Child pid is 5688

Child pid is 8344

self._backendServerProcess.is_alive(): False

self._frontendServerProcess.is_alive(): False

After frontend and backend servers termination.

No child process after the termination. Still, something is running in the Windows terminal.

search "Python multiprocessing subprocess remaining keeping running in Windows terminal" online.

(Nothing useful comes up.)

[5/16/2023 9:30 AM CST] I'll install and use the latest Python, and see what happens.

Search "How do I use a specific version of Python?" online.

<https://stackoverflow.com/questions/11170827/how-do-i-tell-a-python-script-to-use-a-particular-version>

<https://medium.com/swlh/how-to-run-a-different-version-of-python-from-your-terminal-fe744276ff22>

`virtualenv -p C:\Users\Allen\AppData\Local\Programs\Python\Python311`

Search "Python how to use virtualenv" online.

<https://docs.python.org/3/library/venv.html>

<https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/>

<https://realpython.com/python-virtual-environments-a-primer/>

`python -m venv C:\Users\Allen\AppData\Local\Programs\Python\Python311`

Search "Python how to install pip" online.

<https://pip.pypa.io/en/stable/installation/>
<https://www.dataquest.io/blog/install-pip-windows/>

Search "Selenium ImportError: cannot import name 'StrEnum' from 'enum'" online.
<https://stackoverflow.com/questions/44666136/importerror-cannot-import-name-intenum>
<https://techoverflow.net/2019/07/16/how-to-fix-python-importerror-cannot-import-name-enum/>
pip install -U enum34

Search "Selenium Python StrEnum import error" online.

NOTE [5/16/2023 10:19 AM CST]

Execute Python programs in Windows PowerShell, in the specific Python version directory such as C:\Users\Allen\AppData\Local\Programs\Python\Python311.

Specify the current directory Python executable as in ./python and ./pip.

I'm removing all the Python installations on my Windows work computer, rebooting the computer, then installing only the latest Python; I'll see what happens then.

UPDATE [5/16/2023 11:13 AM CST]

After uninstalling and deleting all the Python installations and files, and installing the latest Python version of 3.11.3, selenium-e2e-test.py successfully executes with the Selenium-operated Google Chrome instance (as expected). However, Windows terminal still behaves funky after the program end; something (most likely the frontend server) is still running, and it quits either after visiting the local frontend website in web browser or after pressing Ctrl+C. That at least tells that the problem persists in the latest Python runtime, and so it's probably not a Python runtime problem.

Search "Windows list all the processes running in the current terminal" online.
(This is pointless. Can't run a program in the terminal.)
(Two instances of "Node.js JavaScript Runtime" in the Windows Task Manager. I bet those are the frontend and backend server instances still running after the selenium-e2e-test.py end.)

Search "Windows list all the processes running with certain name" online.
<https://learn.microsoft.com/en-us/windows-hardware/drivers/debugger/finding-the-process-id>

tasklist /?
TASKLIST /FI "IMAGENAME eq node.exe"

Image Name	PID	Session Name	Session#	Mem Usage
node.exe	11776	Console	1	50,484 K
node.exe	12464	Console	1	30,628 K

Before frontend and backend servers termination.


```
Child pid is 14372
Child pid is 14488
Child pid is 14768
Child pid is 12464
Child pid is 14756
Child pid is 11776
self._backendServerProcess.is_alive(): False
self._frontendServerProcess.is_alive(): False
After frontend and backend servers termination.
```

Note that the above two node.exe processes were created by selenium-e2e-test.py.

Pressing Ctrl+C in the Windows terminal that selenium-e2e-test.py was executed in, kills the above two node.exe processes (they disappear in Windows Task Manager).

Search "Python how to kill a process with certain name"
(Not doing this. I want to check out psutil documentation first.)

Search "psutil documentation" online.
<https://psutil.readthedocs.io/en/latest/>
<https://psutil.readthedocs.io/en/latest/#psutil.Process>

```
current_process = psutil.Process()
children = current_process.children(recursive=True)
for child in children:
    print('Child pid is {}; child exe is {}; child name is {}'.format(child.pid, child.exe(),
child.name()))
```

Before frontend and backend servers termination.

```
Child pid is 3764; child exe is C:\Users\Allen\AppData\Local\Programs\Python\Python311\
python.exe; child name is python.exe;
Child pid is 9968; child exe is C:\Users\Allen\AppData\Local\Programs\Python\Python311\
python.exe; child name is python.exe;
Child pid is 12904; child exe is C:\Windows\System32\cmd.exe; child name is cmd.exe;
Child pid is 14348; child exe is C:\Program Files\nodejs\node.exe; child name is node.exe;
Child pid is 14944; child exe is C:\Windows\System32\cmd.exe; child name is cmd.exe;
Child pid is 7220; child exe is C:\Program Files\nodejs\node.exe; child name is node.exe;
self._backendServerProcess.is_alive(): False
self._frontendServerProcess.is_alive(): False
After frontend and backend servers termination.
```

UPDATE [5/16/2023 11:40 AM CST] I'll try saving references to, and killing, the above two node.exe processes in selenium-e2e-test.py using psutil, after terminating the Python processes that launched them.

psutil successfully killed the node.exe processes. The following CLI command verifies.

```
TASKLIST /FI "IMAGENAME eq node.exe"
```

INFO: No tasks are running which match the specified criteria.

I solved the problem. I solved yet another problem through persistence; I'm quite happy with the solution; I can live with it. Alright, I'm moving on; the problem is completely solved.

4.8. Python how to exit a program with an exit code

[5/16/2023 12:17 PM CST]

Search "Python how to exit a program with an exit code" online.

<https://www.scaler.com/topics/how-to-end-program-in-python/>

https://www.geeksforgeeks.org/python-exit-commands-quit-exit-sys-exit-and-os-_exit/

4.9. Checkout page rendering failure troubleshooting

[5/16/2023 2:26 PM CST]

Troubleshoot the PayPal buttons missing and the card input fields being disabled on the checkout page. [5/16/2023 2:20 PM CST]

UPDATE [5/16/2023 2:23 PM CST] Looking at the server debug outputs, it looks like the error has something to do with using /Checkout/:numberOfSharesToBuy end point with GET operation, instead of using /Checkout end point with POST operation.

[5/16/2023 2:30 PM CST] This is a big problem. It might mean needing to update the frontend and Jest-test code.

[5/16/2023 2:33 PM CST] Yeah, it looks like I need to roll back the frontend homepage code, and update the Jest-test code. Is there a way to update app.js code to handle this?

```
express:router dispatching GET /Checkout/app.js +29ms
express:router query : /Checkout/app.js +1ms
express:router expressInit : /Checkout/app.js +1ms
express:router serveStatic : /Checkout/app.js +0ms
send stat "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\
00003 AAM Stockmarket\demo files\backend\Express.js\public\Checkout\app.js" +1s
express:router cookieParser : /Checkout/app.js +6ms
express:router jsonParser : /Checkout/app.js +2ms
body-parser:json skip empty body +1s
express:router urlencodedParser : /Checkout/app.js +3ms
body-parser:urlencoded skip empty body +12ms
express:router corsMiddleware : /Checkout/app.js +16ms
server /Checkout/:numberOfSharesToBuy route method start. 1684265363516 +58ms
express:view lookup "Checkout_invalid-number-of-shares-to-buy.ejs" +10ms
express:view stat "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo
software\00003 AAM Stockmarket\demo files\backend\Express.js\views\Checkout_invalid-
number-of-shares-to-buy.ejs" +1ms
```

```
express:view render "C:\Users\Allen\Documents\Asian-American Man\AllenYoung.dev\demo software\00003 AAM Stockmarket\demo files\backend\Express.js\views\Checkout_invalid-number-of-shares-to-buy.ejs" +1ms
```

The following error shows up in the web browser console when visiting /Checkout webpage.

```
GET https://localhost:8888/Checkout/app.js net::ERR_ABORTED 400 (Bad Request)
```

Updating "<script src='app.js'></script>" to " <script src='<%= totalNumberOfSharesToBuy %>/app.js'></script>" in checkout.js might solve this problem.

Doing the above update produced the following error in the web browser console when visiting /Checkout webpage.

```
GET https://localhost:8888/Checkout/1/app.js net::ERR_ABORTED 404 (Not Found)
```

Refused to execute script from 'https://localhost:8888/Checkout/1/app.js' because its MIME type ('text/html') is not executable, and strict MIME type checking is enabled.

Visiting https://localhost:8888/Checkout/app.js redirects to the "Invalid number of shares to buy" webpage.

UPDATE [5/16/2023 2:57 PM CST] I'm thinking, visiting https://localhost:8888/Checkout/app.js is visiting /Checkout/:numberOfSharesToBuy; that's why it redirects to the "Invalid number of shares to buy" webpage. I got a decision to make: How should I resolve this?

Search "Express.js how does public folder work?" online.
<https://expressjs.com/en/starter/static-files.html>

Looking at <https://expressjs.com/en/starter/static-files.html> gives me an idea. I should try "<script src='../app.js'></script>" in checkout.js. Maybe just being able to download app.js might solve the problem.

[5/16/2023 3:03 PM CST] Yup, it solved the problem! Great! Moving forward!

4.10. Troubleshooting the frontend JavaScript rendered Paypal buttons and card input fields not being found by Selenium API

[5/16/2023 3:36 PM CST]

Troubleshoot the frontend JavaScript rendered Paypal buttons and card input fields not being found by Selenium API.

Maybe this error is being caused by the frontend rendered content using iframe. The container HTML elements can still be found by Selenium API.

Search "Selenium Python how to find a child element of an element" online.

<https://stackoverflow.com/questions/24795198/get-all-child-elements>

<https://www.tutorialspoint.com/locating-child-nodes-of-webelements-in-selenium>

<https://medium.com/@a.k.h.i.l/get-all-child-elements-of-a-parent-element-selenium-8595baf4dbdb>

[5/16/2023 3:48 PM CST] Solved. Find the child elements of a parent element.

4.11. References

[5/16/2023 12:30 PM CST]

4.11.1. Selenium Python documentation

[5/16/2023 12:31 PM CST]

Search "Selenium Python documentation" online.

<https://selenium-python.readthedocs.io/>

<https://selenium-python.readthedocs.io/locating-elements.html>

<https://selenium-python.readthedocs.io/api.html>

<https://selenium-python.readthedocs.io/page-objects.html>

https://en.wikipedia.org/wiki/Software_design_pattern

4.11.2. Python how to check for OS type

[5/16/2023 1:17 PM CST]

Search "Python how to check for OS type" online.

<https://stackoverflow.com/questions/1854/how-to-identify-which-os-python-is-running-on>

<https://note.nkmk.me/en/python-platform-system-release-version/>

4.11.3. Selenium Python get text

[5/17/2023 11:23 AM CST]

Search "Selenium Python get text" online.

<https://stackoverflow.com/questions/20996392/how-to-get-text-with-selenium-webdriver-in-python>

4.11.4. Selenium Python how to check if element is disabled

[5/17/2023 12:11 PM CST]

Search "Selenium Python how to check if element is disabled" online.

<https://stackoverflow.com/questions/23498151/how-to-verify-if-a-button-is-enabled-and-disabled-in-webdriver-python>

4.11.5. Selenium Python scroll to element

[5/17/2023 1:22 PM CST]

Search "Selenium Python scroll to element" online.

<https://stackoverflow.com/questions/41744368/scrolling-to-element-using-webdriver>

4.11.6. How to fix Python was not found on Windows

[5/20/2023 7:25 PM CST]

Search "How to fix Python was not found on Windows" online.

<https://stackoverflow.com/questions/13596505/python-command-not-working-in-command-prompt>

<https://bobbyhadz.com/blog/python-was-not-found-run-without-arguments-to-install>

Search "Python was not found; run without arguments to install from the Microsoft Store, or disable this shortcut from Settings > Manage App Execution Aliases." online.

4.11.7. How to remove an installed Python version on Windows

[5/20/2023 7:36 PM CST]

Search "How to remove an installed Python version on Windows" online.

<https://linuxhint.com/uninstall-python-windows/>

XIII. Test Code Design

1. Unit Test Code Design

1.1. Full-Stack Web App Unit Test Code Design

1.1.1. Python/Django Full-Stack Web App Unit Test Code Design

1.1.2. PHP/Laravel Full-Stack Web App Unit Test Code Design

1.1.3. Ruby on Rails Full-Stack Web App Unit Test Code Design

1.2. Frontend Web App Unit Test Code Design

1.2.1. Angular Frontend Web App Unit Test Code Design

1.2.2. React Frontend Web App Unit Test Code Design

1.2.3. Vue Frontend Web App Unit Test Code Design

1.2.4. JQuery Frontend Web App Unit Test Code Design

1.3. Backend Web App Unit Test Code Design

1.3.1. Python/Flask Backend Web App Unit Test Code Design

1.3.2. Go/Gin Backend Web App Unit Test Code Design

1.3.3. TypeScript/Node/Express Backend Web App Unit Test Code Design

1.3.4. PHP/Laravel Backend Web App Unit Test Code Design

2. Integration Test Code

2.1. Full-Stack Web App Integration Test Code Design

2.1.1. Python/Django Full-Stack Web App Integration Test Code Design

2.1.2. PHP/Laravel Full-Stack Web App Integration Test Code Design

2.1.3. Ruby on Rails Full-Stack Web App Integration Test Code Design

2.2. Frontend Web App Integration Test Code Design

2.2.1. Angular Frontend Web App Integration Test Code Design

2.2.2. React Frontend Web App Integration Test Code Design

2.2.3. Vue Frontend Web App Integration Test Code Design

2.2.4. JQuery Frontend Web App Integration Test Code Design

2.3. Backend Web App Integration Test Code Design

2.3.1. Python/Flask Backend Web App Integration Test Code Design

2.3.2. Go/Gin Backend Web App Integration Test Code Design

2.3.3. TypeScript/Node/Express Backend Web App Integration Test Code Design

2.3.4. PHP/Laravel Backend Web App Integration Test Code Design

XIV. Deployment Methods

1. Deployment configuration

[4/16/2023 12:53 PM CST]

1.1. How to set up the backend process manager for automatic health checkup every 30 seconds and email admin if the backend is down

[4/16/2023 12:53 PM CST]

1.2. Replacement of applicable settings in source code files

[4/18/2023 9:09 PM EST]

e.g. backend server URL

changing React runtime script include in HTML files to production version

(list all!)

1.3. References

1.3.1. Is CORS needed for subdomain

[5/1/2023 9:39 PM EST]

Search "Is CORS needed for subdomain" online.

<https://stackoverflow.com/questions/26724137/do-i-need-to-enable-cors-when-my-api-is-on-a-subdomain-of-my-main-website>

2. Manual Deployment Methods

[3/31/2023 11:25 PM CST]

2.1. Manual backend deployment methods

2.1.1. Manual AWS Lightsail Express.js backend deployment method

2.2. Manual frontend deployment methods

2.2.1. Manual AWS Lightsail React.js frontend deployment method

XV. Potential future updates

add input check code in the homepage form. do not proceed with button press if the no valid share number is entered. [4/19/2023 1:22 PM CST] (currently, there is no event handler function implemented for the button press; it is done with the form submission, hence no customization. I'm not adding the code now. I'm not adding the event handler function now. maybe later, maybe not. [4/26/2023 4:00 PM CST])

[4/26/2023 6:33 PM CST] Record the transaction details returned from PayPal in database (especially the payment method info), so that it can be displayed in the receipt by the /Receipt/:orderID route method.

Update the code to add payment type and source in the receipt using the payment data from PayPal. [4/19/2023 2:31 PM CST]

"Allen Young's Stockmarket Demo payment record" email sent has a very messy content. think about whether to change it or not. [4/19/2023 1:51 PM CST] ([4/26/2023 7:24 PM CST] for now, I'm leaving it as it is. In the future, it can be prettified.)

transaction receipt display link for each successful transaction in account.html webpage. [4/24/2023 11:08 AM CST]

[4/28/2023 8:17 PM CST] On the homepage, in the company stock data section, implement the waiting swirl animated icon.

[4/29/2023 4:53 PM CST] On each applicable page, initially set the tab stop on the default button, so that pressing enter will press the button.

on account.html, make sure that "Your email address will not be updated until you click the link in the email-address verification link." happens as notified. [4/24/2023 5:01 PM CST] ([5/1/2023 11:40 AM CST] ok, I need to think about this. I think this should be a future update. I'm leaning toward not doing this now, due to the time constraint. so, just remove or comment out the notice in the webpage, and don't send the email-address verification email.) (FINAL DECISION [5/1/2023 11:41 AM CST] due to the present time constraint, I'm making this a future update, along with creating a new database table for keeping all the old user info,

including old email addresses. Replacing "Your email address will not be updated until you click the link in the email-address verification link." with "Your email address will be updated instantly without the email-address verification. Please make sure that the new email address you enter is correct.")

[5/1/2023 12:26 PM CST] Implement 'My Account | Logout' in the right-top corner in the top bar of the receipt page.

implement sending Telegram message (instead of, or in addition to, email) when Stockmarket homepage and checkout page are visited, with a module const variable (likely named `notifyOnTelegram`) for turning this on and off. [4/27/2023 2:49 PM CST]

[5/7/2023 5:31 PM CST] For GET routes or end points for webpage return, pretty error page return in HTML, instead of returning plain-text error, such as at /Checkout.

[5/10/2023 2:37 PM CST] Use Mock Service Worker library to declaratively mock API communication in the Jest DOM unit and integration tests instead of stubbing `fetch()` wrapper functions, so that the fetch code gets tested.

<https://testing-library.com/docs/react-testing-library/example-intro>

<https://testing-library.com/docs/react-testing-library/faq>

[5/12/2023 4:25 PM CST]

Write and execute `__tests__/EmailAddressVerificationNotice-test.js` with the backend mocking to test all the frontend code including the fetch code.

[5/13/2023 12:01 PM CST]

On the user account page, update the UI for enabling updating each user account setting separately with a separate button (e.g. 'Update' button next to email input field, and old email address being displayed below the email input field.)

Also, in the backend, update the database and database code for keeping track of all the changes.

[5/15/2023 11:37 AM CST]

Update /ResetPassword end-point operation to PATCH, since that is more proper than POST.

Search "put vs. patch" online.

PUT is a technique of altering resources when the client transmits data that revamps the whole resource. PATCH is a technique for transforming the resources when the client transmits partial data that will be updated without changing the whole data.

<https://byjus.com/gate/difference-between-put-and-patch-request/>

[5/17/2023 1:17 PM CST]

Make email addresses case-insensitive during ~~registration and~~ login.

[5/18/2023 5:18 PM EST] Update the code to use email-address case-insensitive match during login.

[5/19/2023 3:26 PM CST]

AWS Fargate blue/green deployment implementation in Jenkins pipeline.

XVI. References

XVII. Software License

1. [4/19/2023 5:48 PM CST]

This software, Allen Young's Stockmarket Demo, has been created by Allen Young to demonstrate his skills in full-stack web app development and deployment using the most prevalent software development and deployment libraries, frameworks, and tools.

This software is made available publicly by Allen Young under the xx license.

You may freely distribute and modify this software under the following terms.

...

XVIII. =====

=====

XIX. =====

=====

XX. =====

=====

XXI. =====

=====

XXII. =====

=====