Solution for QA challenge

Introduction:

As a solution of QA tasks. I have created the Test Automation framework using which we can capable to write automated test scripts for trello.com

Pre-requisite:

- **NPM** as node package manager is installed
- **NodeJS** as runtime environment is installed
- **WebdriverIO** as base test automation framework
- **MochaJS** as test runner
- **Babel-JS** as JavaScript compatibility
- **Page object model** with layered architecture

Steps to execute:

1. Clone the code from GitHub: https://github.com/hardikdiyora/coding_task.git
2. Import the code into IDE for better visibility. (preferably, IntelliJ IDEA)
3. Edit the file credential.js and put email and password accordingly.
4. Run command: **npm install**
5. Run command: **npm test**

Folder Structure:

```
|-- wdio.conf.js                    (WebdriverIO configuration file)
|
|-- package. json                   (NodeJS manifest/metadata file)
|
|-- src
|     |--config                     (Common configurations like credentials)
|     |     |--credential.js
|     |
|     |--pages.                     (Page classes and objects)
|     |     |--boards
|     |               |--page.js        (Page class with behaviors)
|     |               |--selector.js    (Web elements locators for that page)
|     |
|     |--tests                      (Tests directory)
|     |     |--boards               (Feature-wise directory)
|     |          |--board.test.js   (test file containing test scripts)
|     |
|     |--utils
|           |--general.js           (Common utilities like randomText, takeScreenshots etc.)
|
|-- babel.config.js                 (Babel-JS configuration file)
|
|-- results_logs                    (Execution output logs & error screenshots)
          |-screenshots
```

Tests Automated:

1. Verify user can able to create a private board successfully
2. Verify user can able to open a private board successfully
3. Verify user can able to add list inside a board successfully

Possible next-step:

1. For standard: include ESlint into framework.
2. Add more re-usable constructs with adding more test scripts.
3. Build test pyramid safety net with component tests, integration tests, and E2E system integration tests.
4. Reduce duplication code and optimize using mocha hooks.