**UI Automation (E2E Tests)**

**Approach**

1. I have used a hybrid framework which includes Cypress + TypeScript + Mocha.

2. Benefits of this is it is very easy to read Scenario flow and easy to maintain.

**Library used to automate Front End**

1. I have chosen Cypress as a JS/TS based framework to automate UI actions.

2. Cypress is an open source framework to automate user actions.

3. Cypress is compatible with a lot of different npm modules.

4. Cypress directly injects HTML for execution so it’s fast in terms of performance.

**Programming language**

1. I have chosen TypeScript as a programming language.

2. Typescript is a modern world programming language widely used for modern web application development.

3. We can use the same programming language for UI as well as API Automation.

**Unit testing framework**

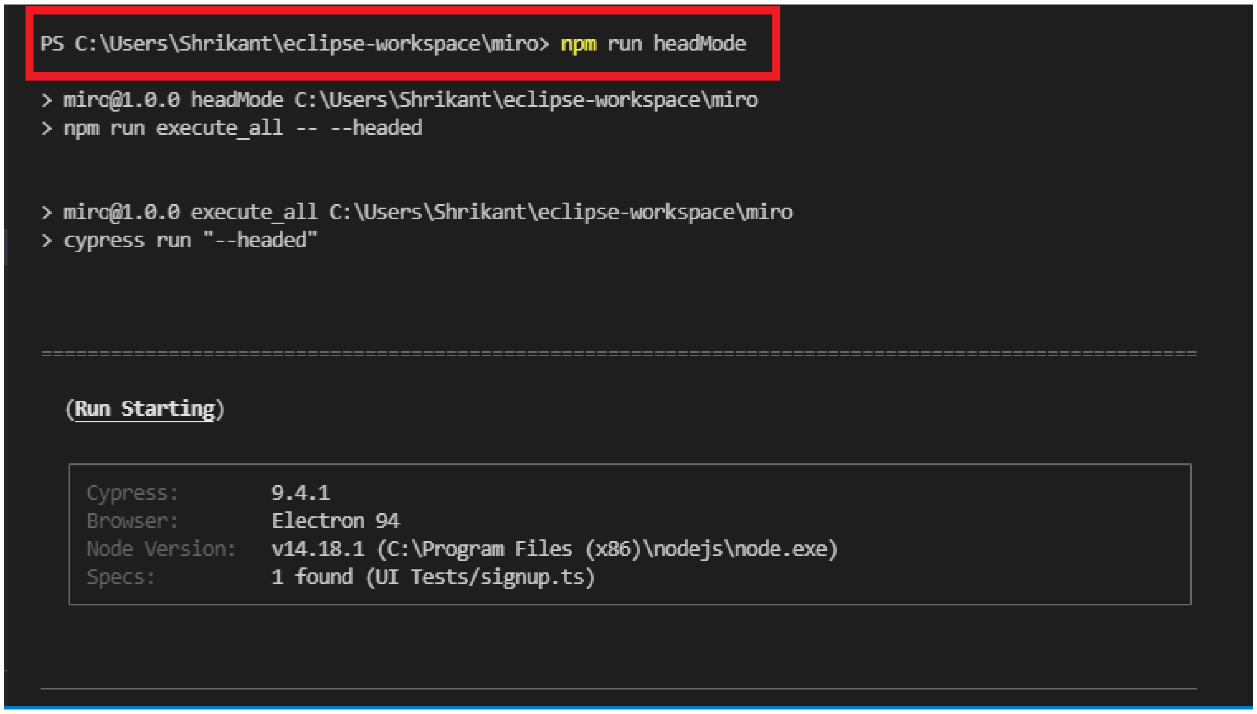
I have chosen Mocha as a unit-testing framework because it is most compatible with JS/TS.

**How to run**

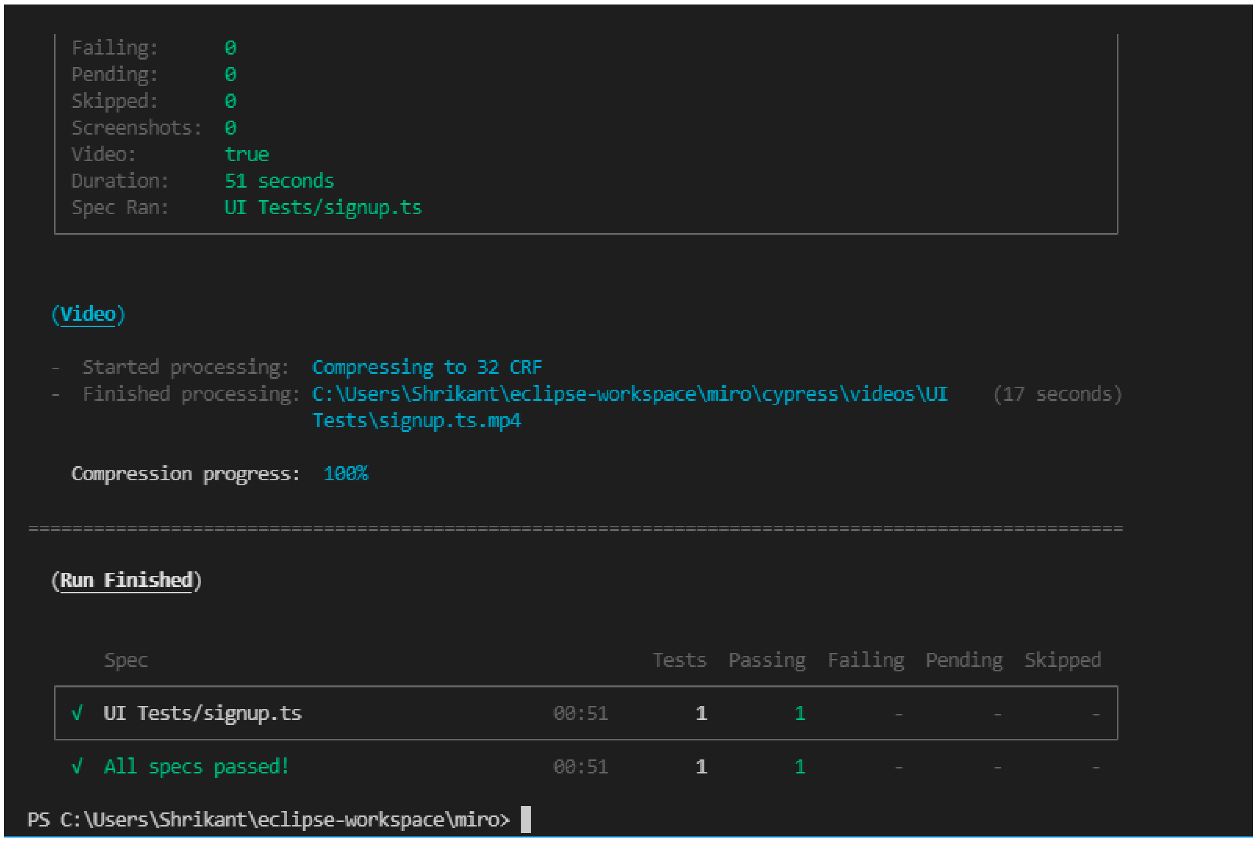
1. Clone the entire project on a local machine from GitHub.

2. Execute **npm install** to install all required packages from package.json

2. Execute npm command from command-line as per below screen-shot.



```
PS C:\Users\Shrikant\eclipse-workspace\miro> npm run headMode

> miro@1.0.0 headMode C:\Users\Shrikant\eclipse-workspace\miro
> npm run execute_all -- --headed


> miro@1.0.0 execute_all C:\Users\Shrikant\eclipse-workspace\miro
> cypress run "--headed"



====================================================================================================

  (Run Starting)

  ┌────────────────────────────────────────────────────────────────────────────────────────────────┐
  │ Cypress:        9.4.1                                                                          │
  │ Browser:        Electron 94                                                                    │
  │ Node Version:   v14.18.1 (C:\Program Files (x86)\nodejs\node.exe)                              │
  │ Specs:          1 found (UI Tests/signup.ts)                                                   │
  └────────────────────────────────────────────────────────────────────────────────────────────────┘


────────────────────────────────────────────────────────────────────────────────────────────────────
```



```
  │ Failing:      0                                                                          │
  │ Pending:      0                                                                          │
  │ Skipped:      0                                                                          │
  │ Screenshots:  0                                                                          │
  │ Video:        true                                                                       │
  │ Duration:     51 seconds                                                                 │
  │ Spec Ran:     UI Tests/signup.ts                                                         │
  └──────────────────────────────────────────────────────────────────────────────────────────┘


  (Video)

  -  Started processing:  Compressing to 32 CRF
  -  Finished processing: C:\Users\Shrikant\eclipse-workspace\miro\cypress\videos\UI    (17 seconds)
                          Tests\signup.ts.mp4

    Compression progress:  100%

====================================================================================================

  (Run Finished)


       Spec                                              Tests  Passing  Failing  Pending  Skipped
  ┌──────────────────────────────────────────────────────────────────────────────────────────┐
  │ √  UI Tests/signup.ts                       00:51        1        1        -        -        - │
  └──────────────────────────────────────────────────────────────────────────────────────────┘
    √  All specs passed!                        00:51        1        1        -        -        -

PS C:\Users\Shrikant\eclipse-workspace\miro> █
```

**Reusable NPM commands -**

```
"scripts": {

   "execute_all": "node_modules\\.bin\\cypress run",

   "headMode": "npm run execute_all -- --headed",

   "chromeTest": "npm run execute_all -- --browser chrome",

   "parallelRun": "npm run execute_all -- --browser chrome --parallel",

   "recordDashboardTest": "npm run execute_all -- --record --key
af58abc1-3d54-4480-8988-03783009b593"

  },
```

**Reporting**

1. Dashboard reporting - [Cypress Dashboard]()

2. HTML Reporting

**Debugging -**

1. Assertions / Video recording / Screen shots on failures.

**<u>What features we can add to framework</u>**

1. Execute Automation scripts in headless mode for faster execution.

2. Implement parallel execution to reduce overall execution time.

3. Integrate Automation with Jenkins for continuous integration.

4. Pull code at runtime from GitHub by Jenkins, execute, and send emailable reports to relevant audiences.

5. Create Automation dashboard to monitor 24*7 Jenkins jobs status.

6. Store automation reports on cloud ex – S3 bucket in AWS.

7. Manage most of Automation configuration as command-line arguments.

8. Separate Object Repository/Test Data on Cloud or any other Third Party tool for less maintenance.

9. Add Browser Compatibility feature to execute scripts on different browsers/OS etc.

10. Execute same scripts on different environments ex – Integration/Staging/Preview/Production etc.

11. Integrate Automation with different Third party tools ex – Slack for notification purpose / Integrate with JIRA to update automation results.