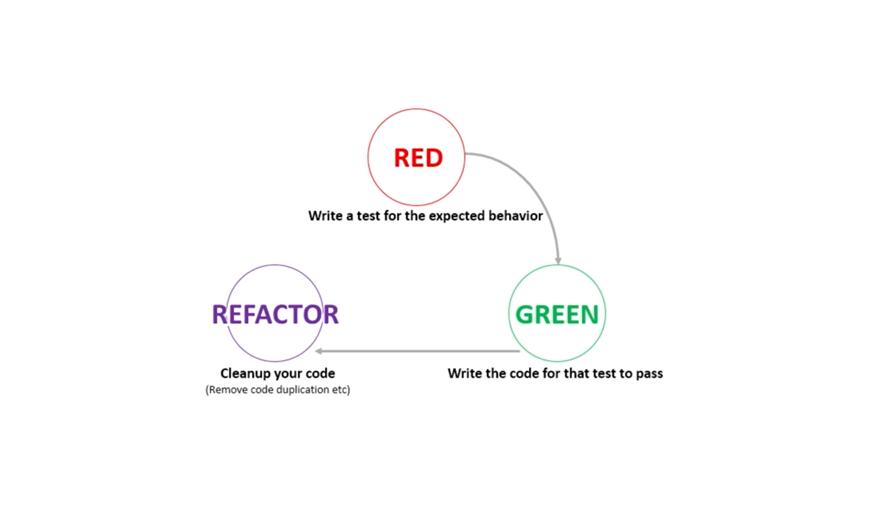
**Test Driven Development Methodology**

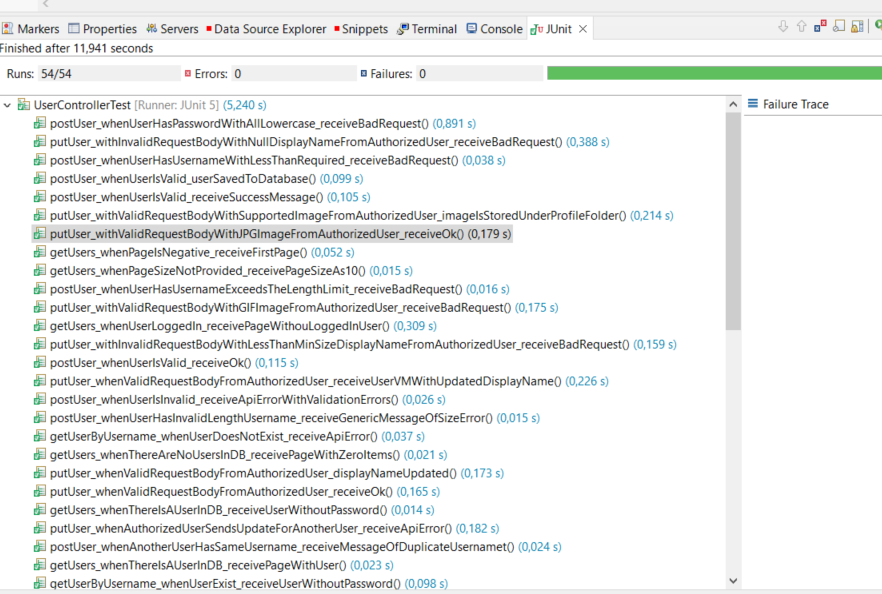
* Based on testing code before implenting requirements
* Allows for clean and reusable code

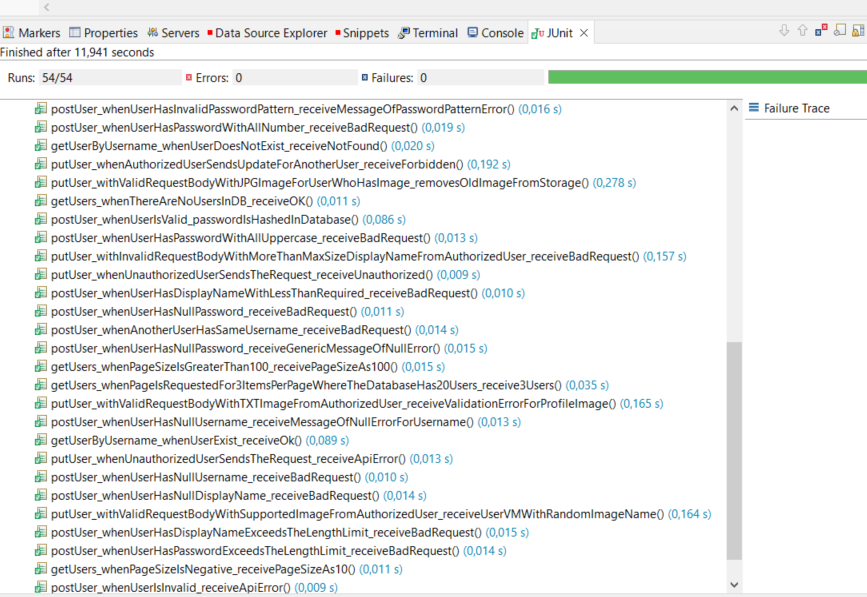


**Backend**

**Integration test: UserControllerTest.java**

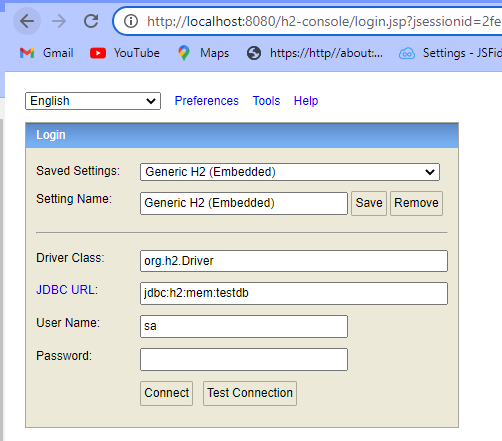
**Run as JUnit Test**





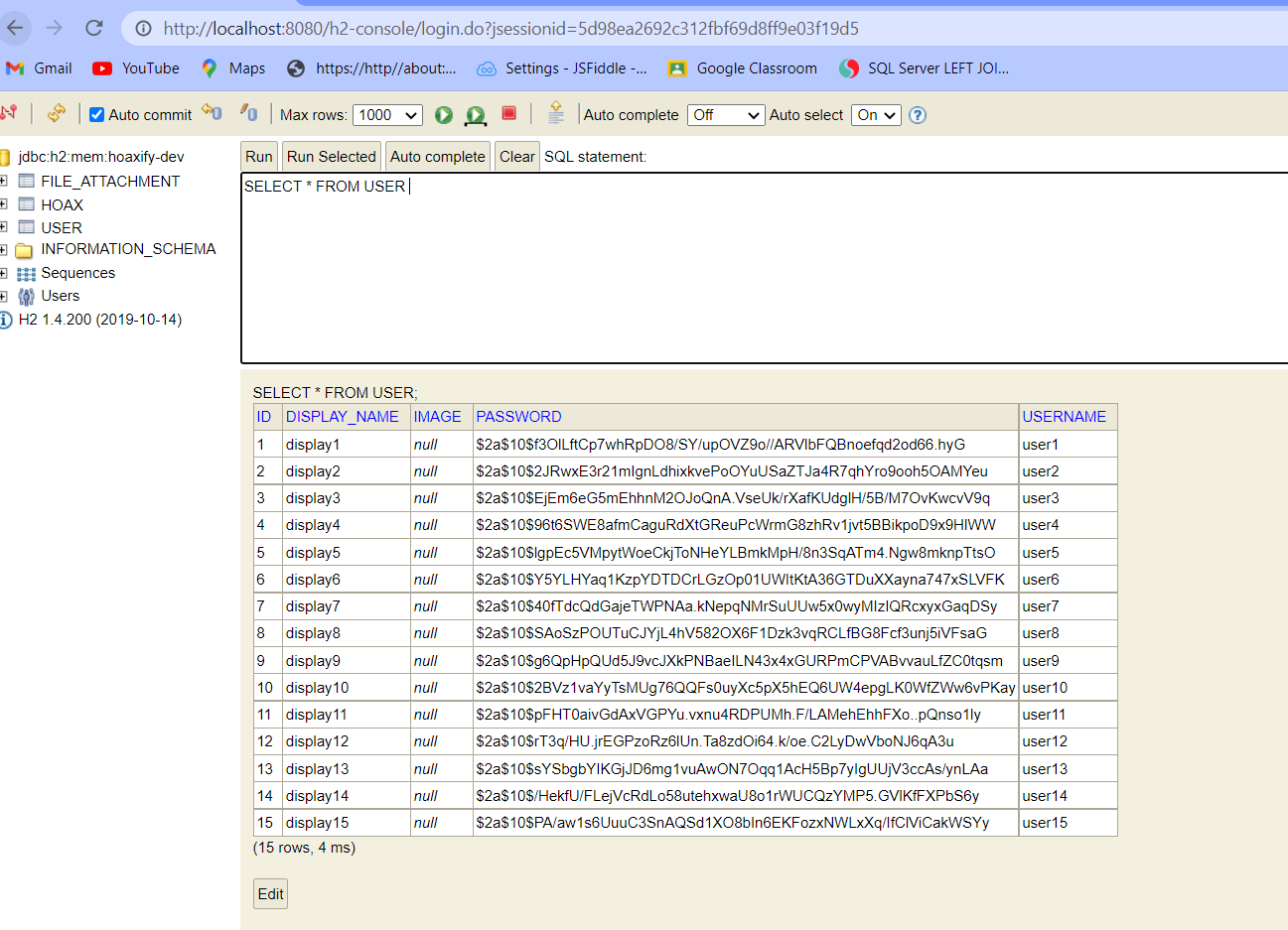


**H2 Console/Database**

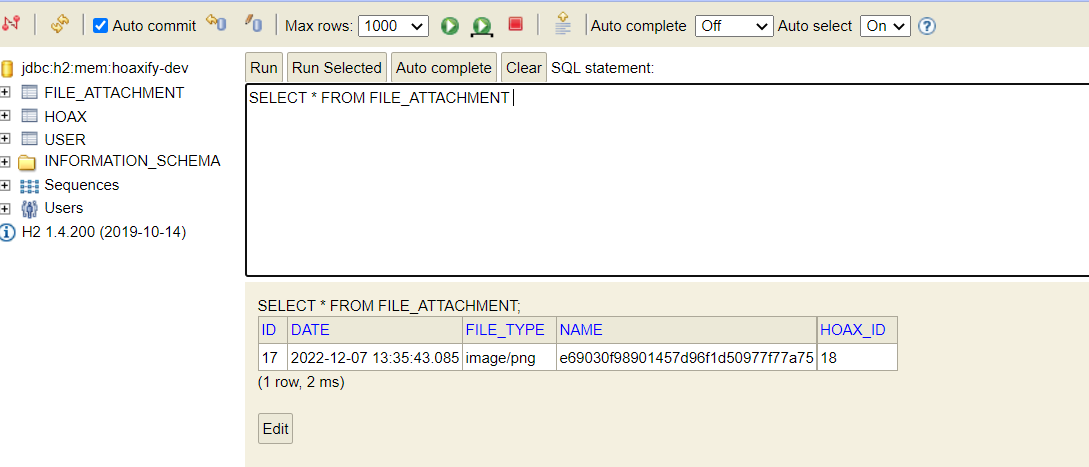


* JDBC URL---Mem: memory
* **H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:hoaxify-dev'**
  + Repalace JDBC URL with this

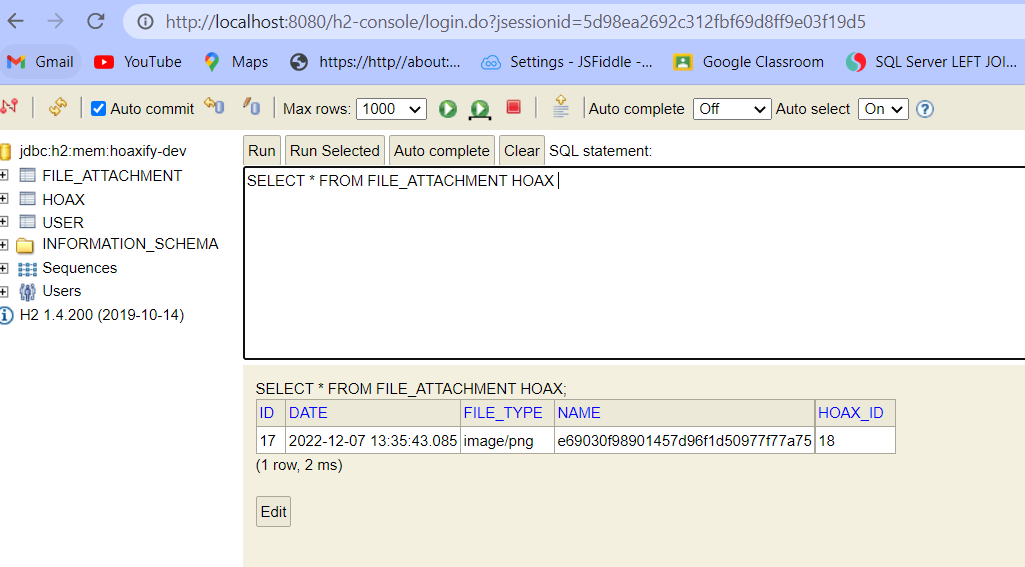
**User table**



**File attachments Table**

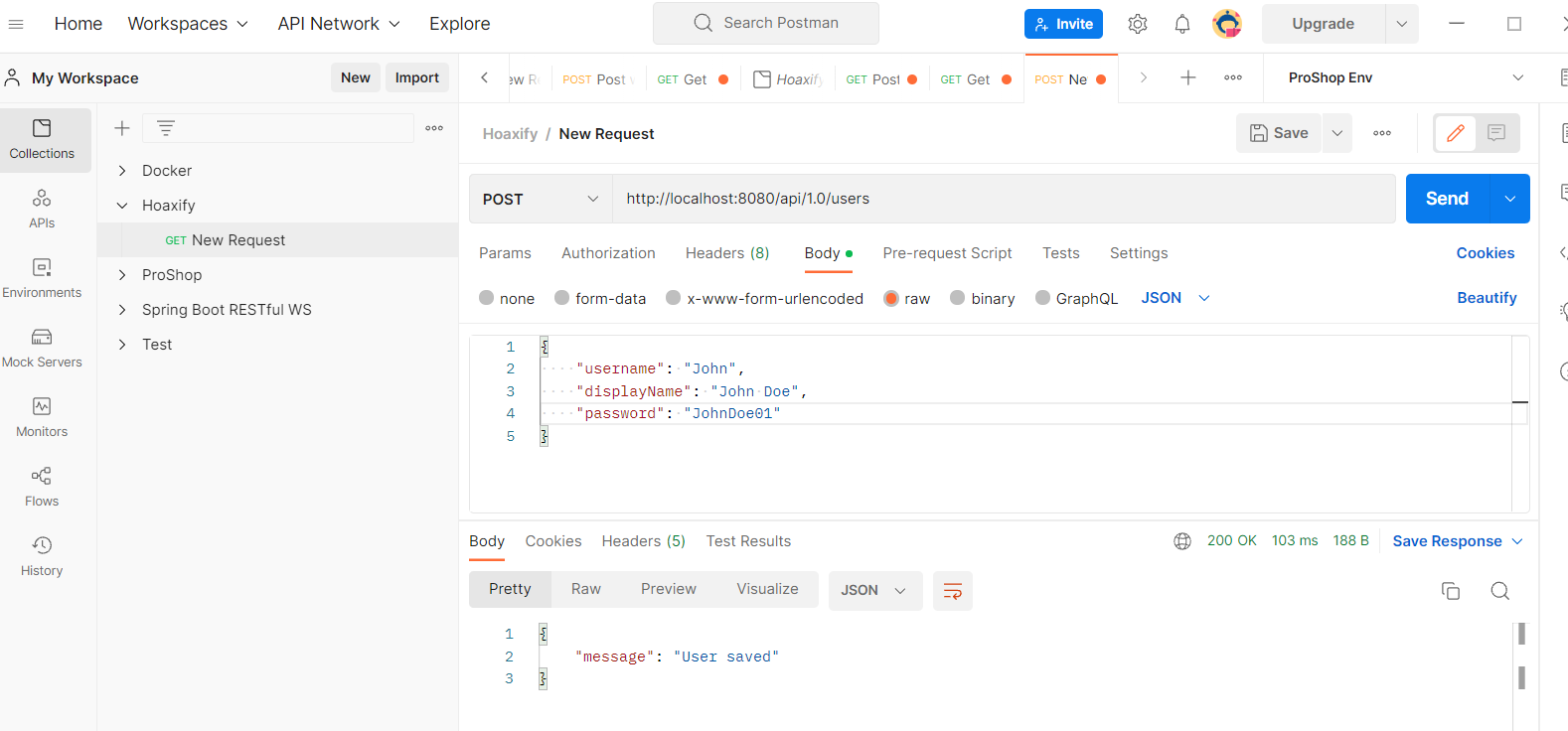


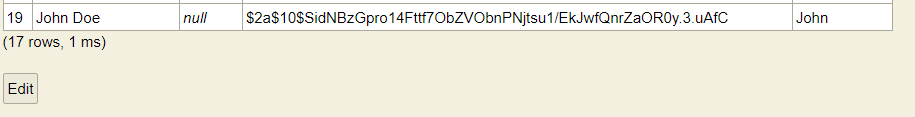
**Hoax Table**



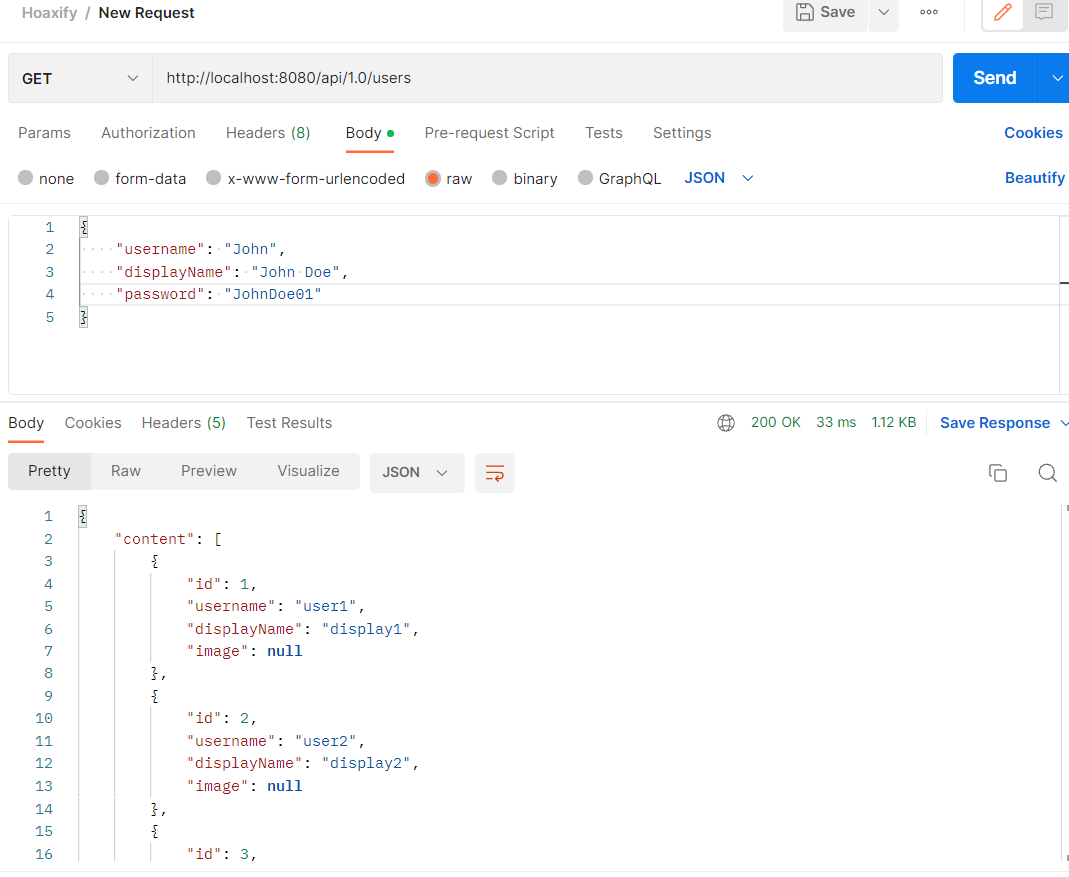
**Postman tests**

**password hashing**



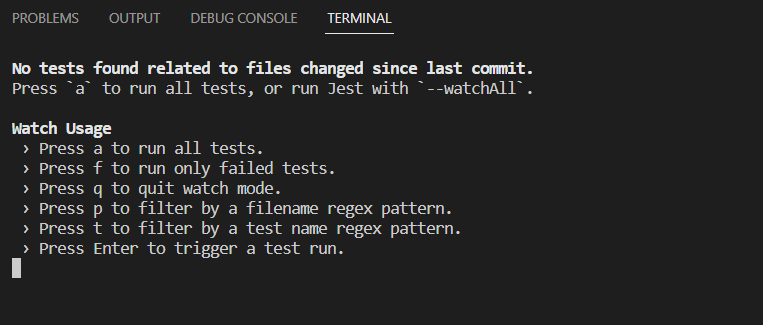


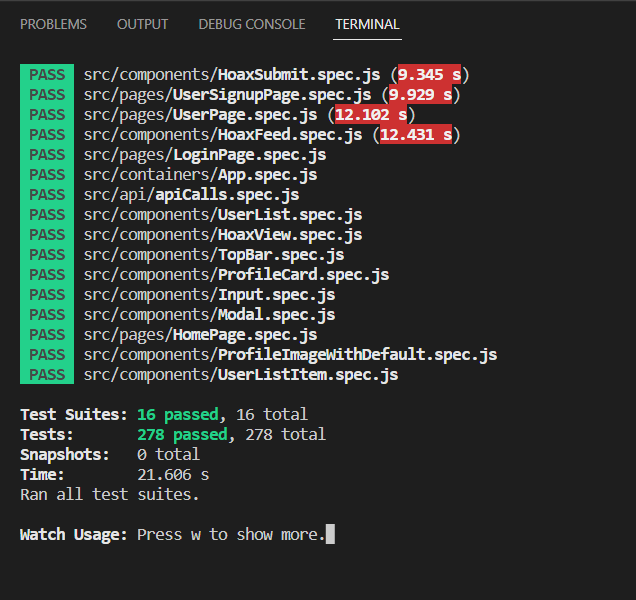
**Get Users**

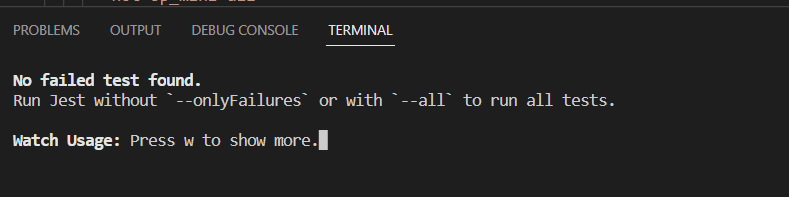


**Running tests on Frontend**

* Run npm test
  + Tests run in watchmode, meaning whenever we update code and build out tests, our application will be refreshed.

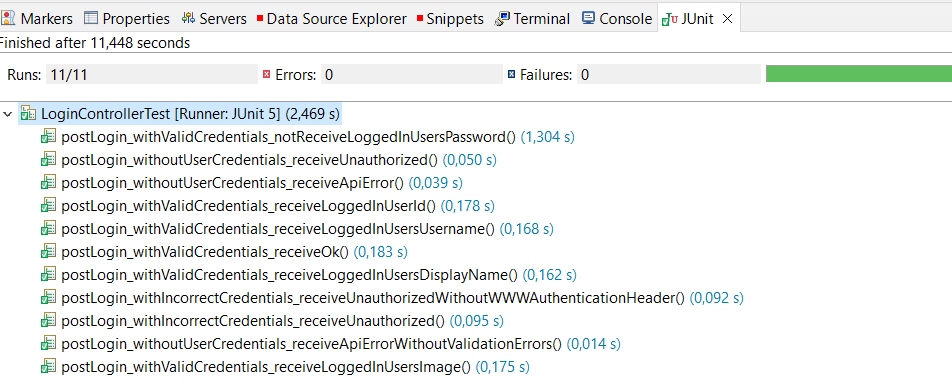




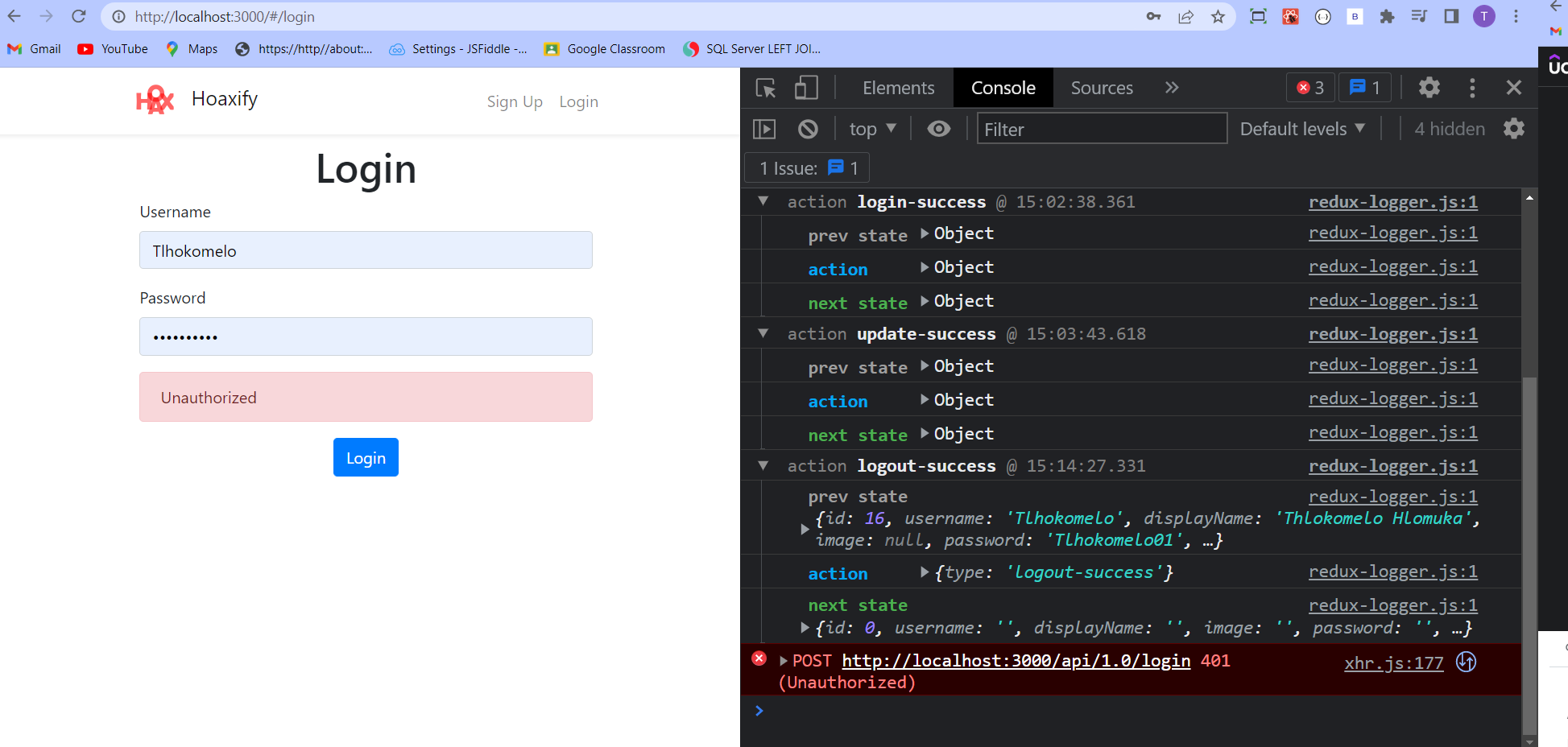


**Integration test: LoginControllerTest.Java**

**Run as JUnit Test**

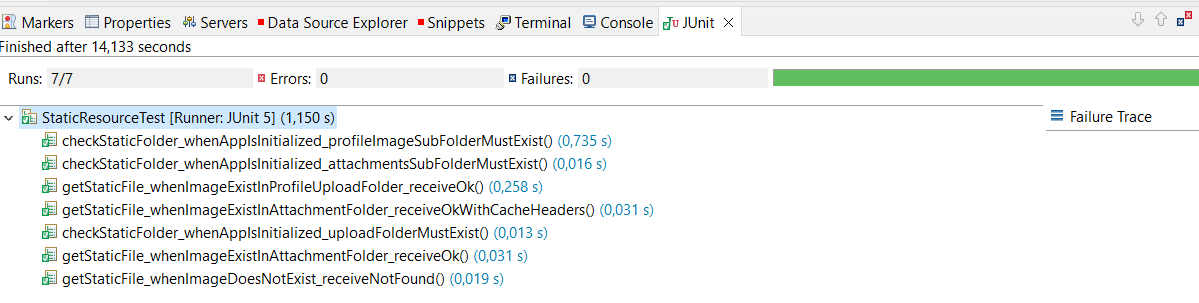


**Redux Connection Test**



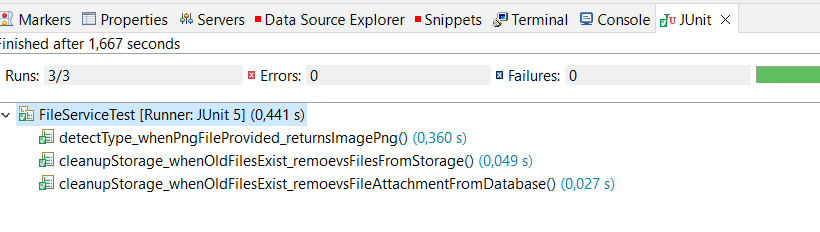
**Integration Test: StaticResourceTest.java**

**Run as JUnit Test**



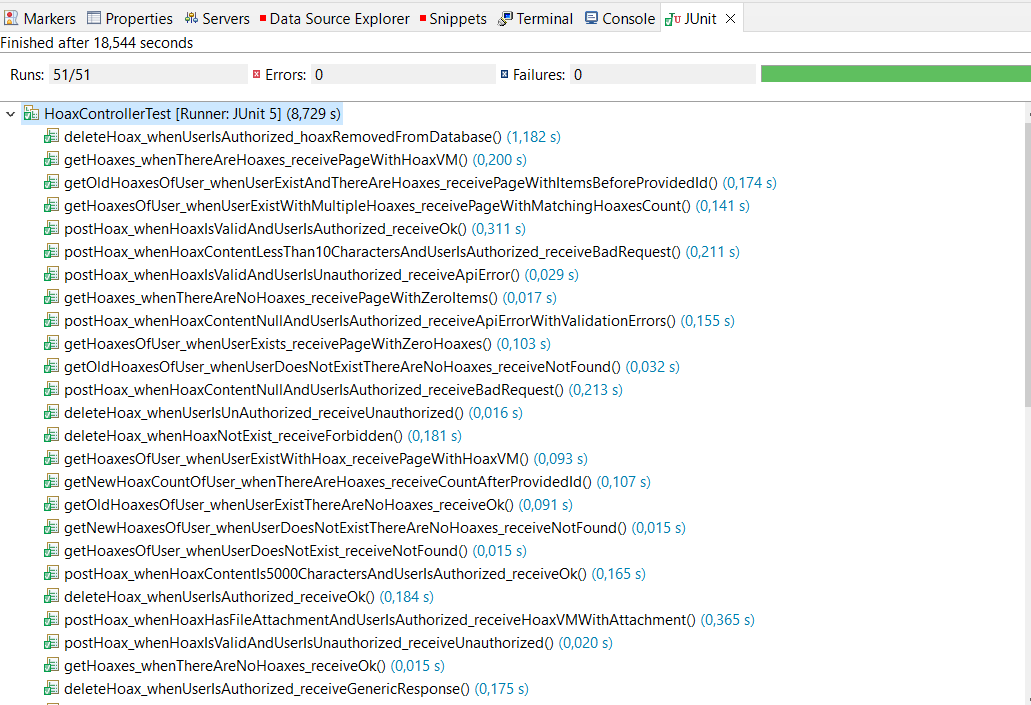
**Integration Test: FileServiceTest.java**

**Run as JUnit Test**

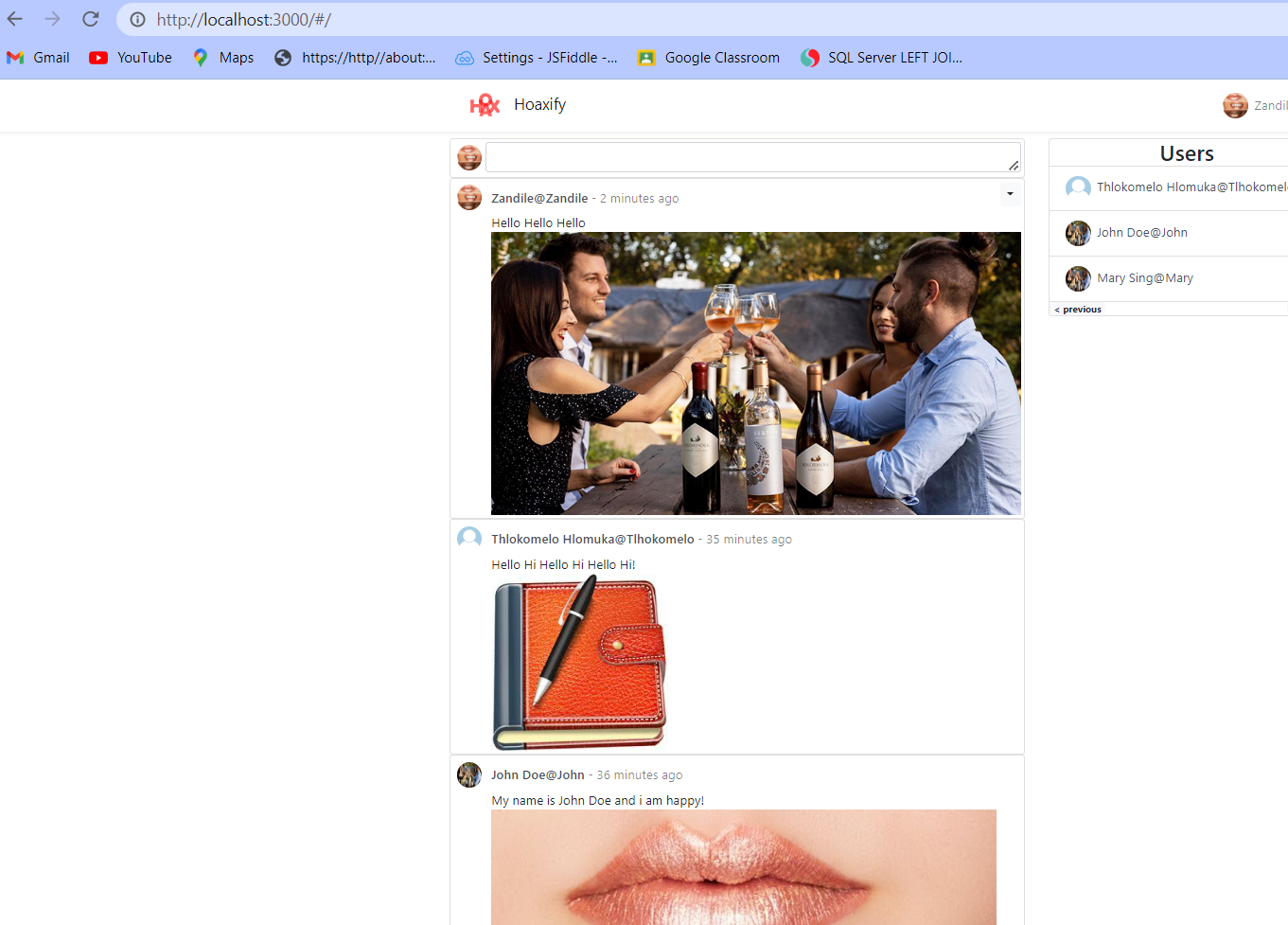


**Integration Test: HoaxControllerTest.java**

**Run as JUnit Test**

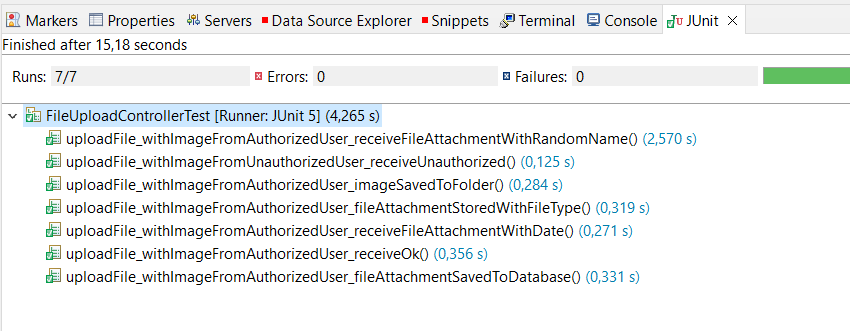


**Loaded Hoaxes**



**Integration Test: FileUploadControllerTest.java**

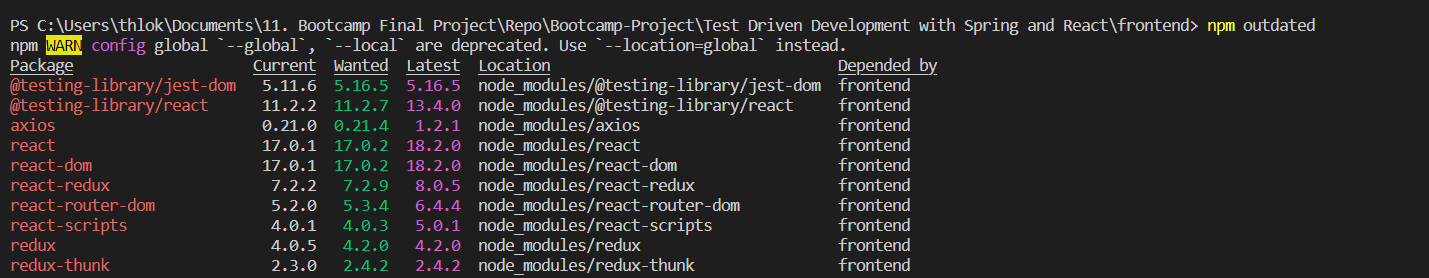
**Run as JUnit Test**



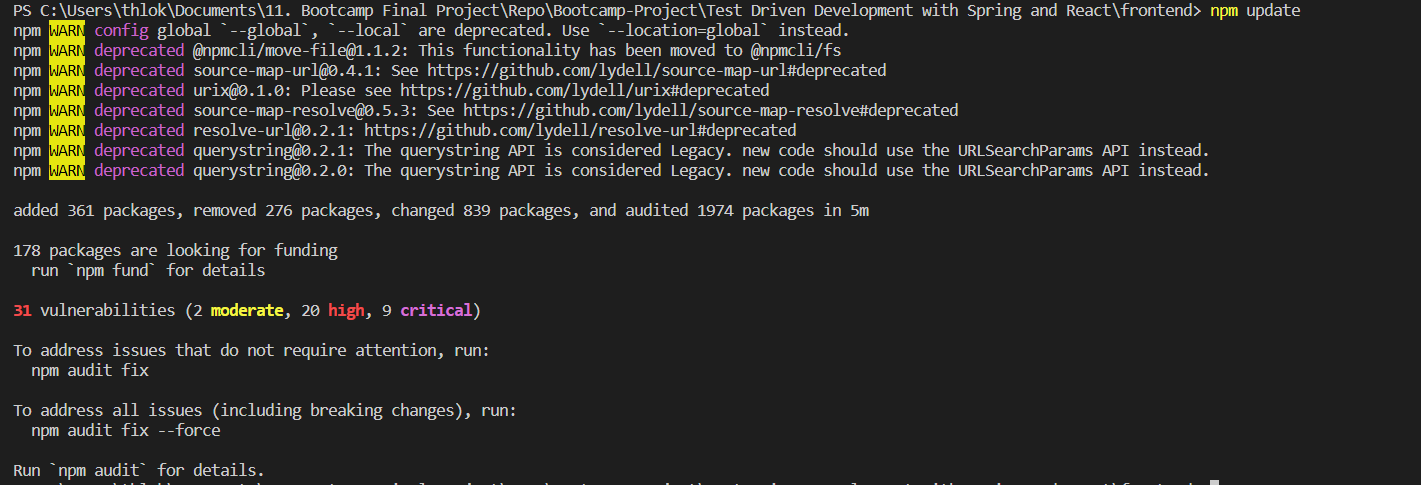
**Upgrades**

**Frontend**

* **To see which dependencies are outdated**
  + Run npm outdated



* **To update outdated depndencies**
  + Run npm update



**Running tests in Coverage mode**

* Makes it possible to spot untested paths in our code. It is an important metric for determining the health of a project.
* Run npm test -- --coverage --watchAll=false

