# Customer Meeting

Customer meeting date to demo this iteration.

Our next meeting with the client is on 12-April-2022 at 4:45 pm

### Meeting notes from the previous client meeting:

* Discussion on the idea of showing comparison per response graph to participants
  + Dr. Thomas said this graph needs a lot of cognitive effort to understand
  + No need to have this graph as the information shown is not much use for the participant
  + Instead, focus on the bar graph showing the summary of responses of participants. Rainbow colors? Shapes? Min-max labels?
* Can we convert the bar graph to a pie chart?
  + Dr. Thomas: Yes but the linear relationship is not shown in a pie chart
* Rescheduled the meeting to some other time between 12:30 pm to 3:30 pm

## User Stories Implemented

User stories are implemented in this iteration.

User stories we implemented in this iteration are:

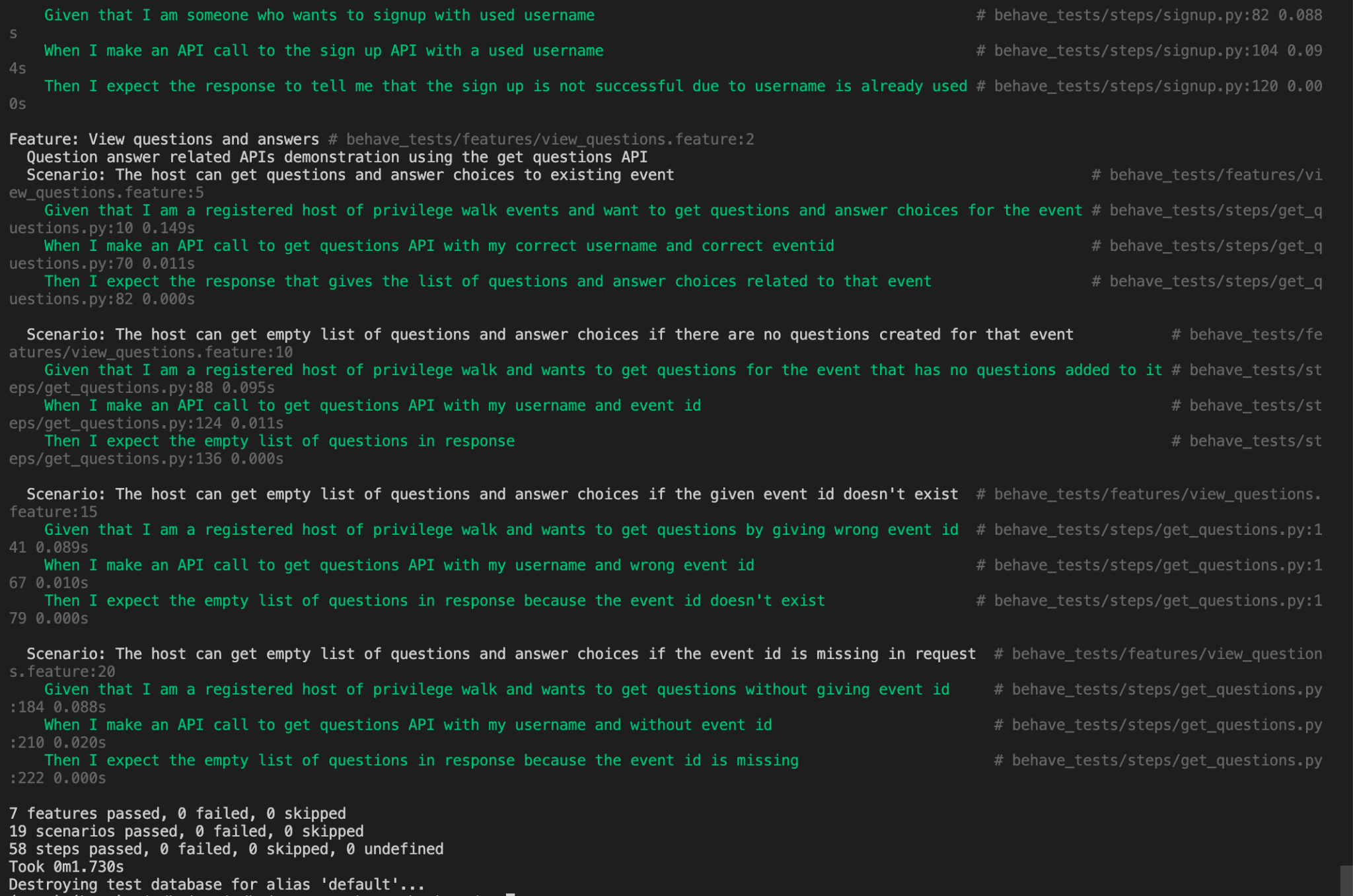
1. User Story: As a host, before the event, I wish to be able to click start to allow participants to now start an event.
2. User Story: As a host, before the event, I wish to see how many people have joined the event
3. User Story: As a host, during the event, I wish to see the current questions users are answering.
4. User Story: As a host, before the event, I wish to display a QR code that people can use to join
5. User Story: As a participant, I wish to see the questions answered so far and how many are left, so that I can know how many questions do I still need to answer
6. User story: As a participant, I want to be shown a QR code, that I can scan to join a live event
7. User Story: As a host, I want to host the event live
8. User story: As a participant, I want to see a question and possible answer choices, so that I can answer a question and participate in the study.
9. User Story: As a host, before the event, I wish to give participants a link to join the event.
10. User Story: As a host, during the event, I wish to be able to click next to show the next question.
11. User story: As a participant, I want to be shown a link, that I can use to join a live event.

We have implemented all the user stories we planned to implement for this iteration.

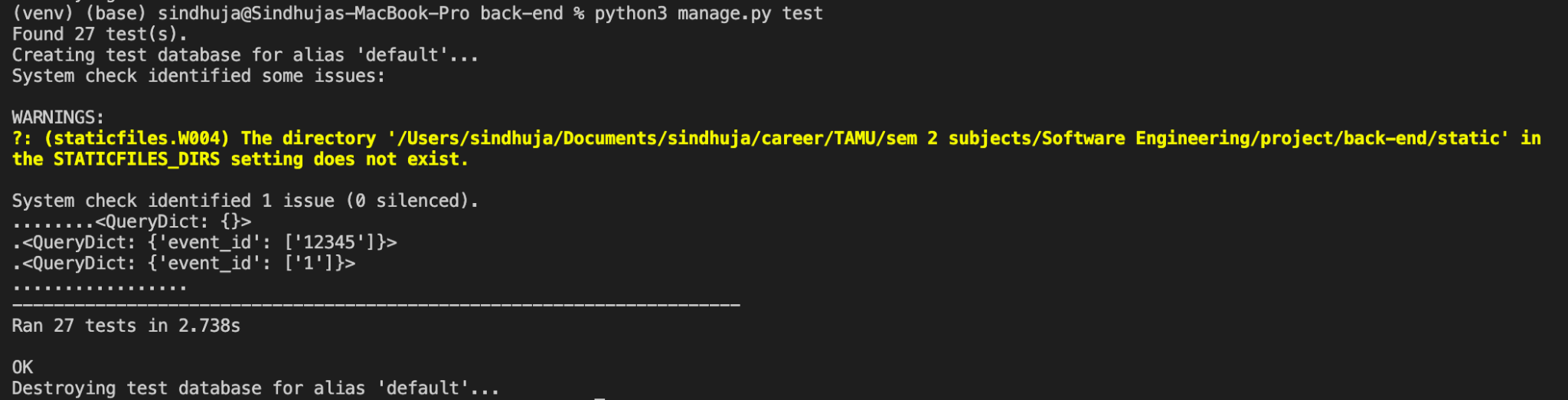
# TESTS

## Backend

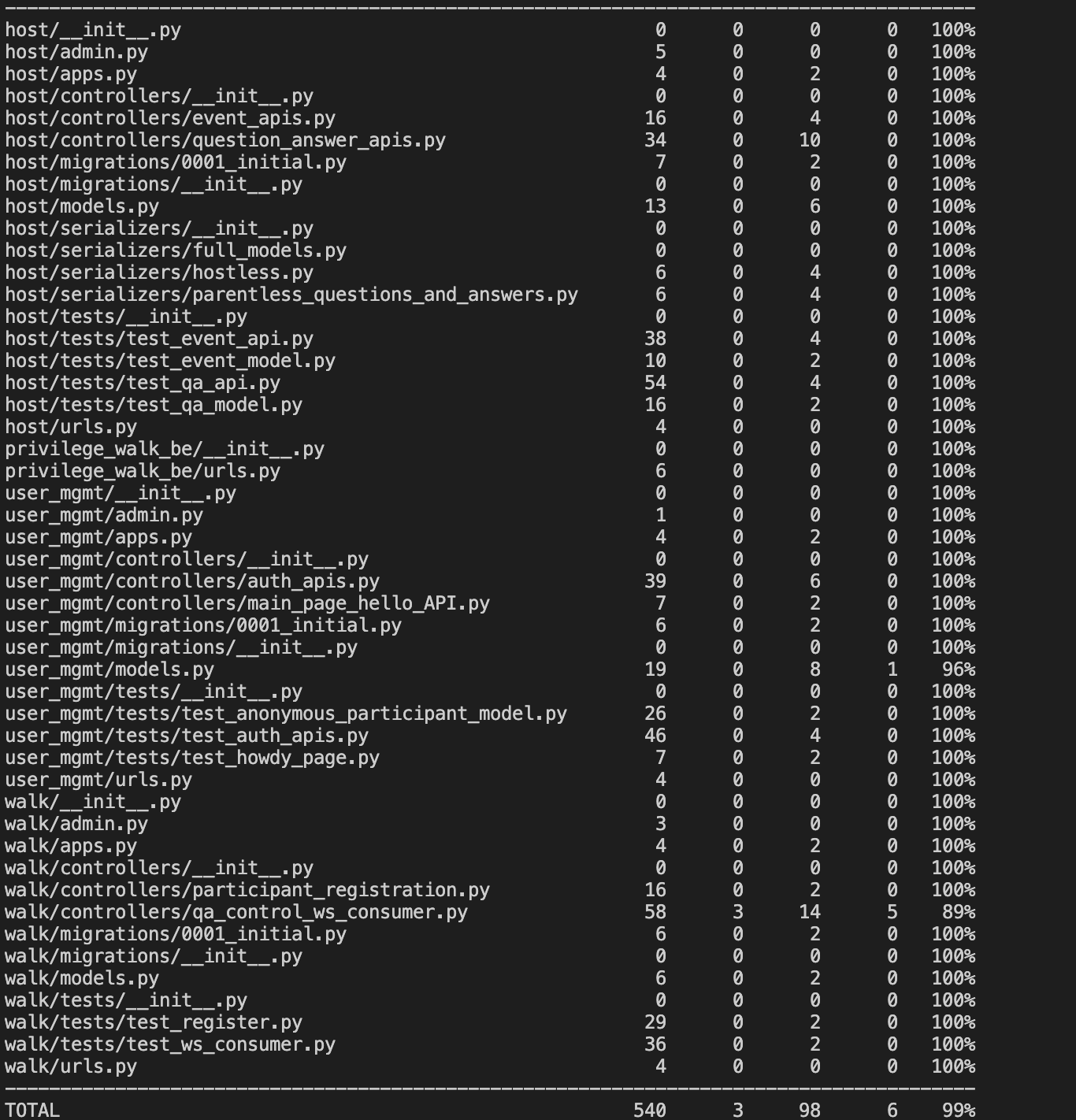
**Results:** BDDs test results for backend apis



Backend TDDs



**Coverage:** Backend tests with 99% coverage.



The instructions to get coverage report is mentioned in ReadMe of repo: <https://github.com/Privilege-walk/back-end.git>

Steps to Run backend tests:

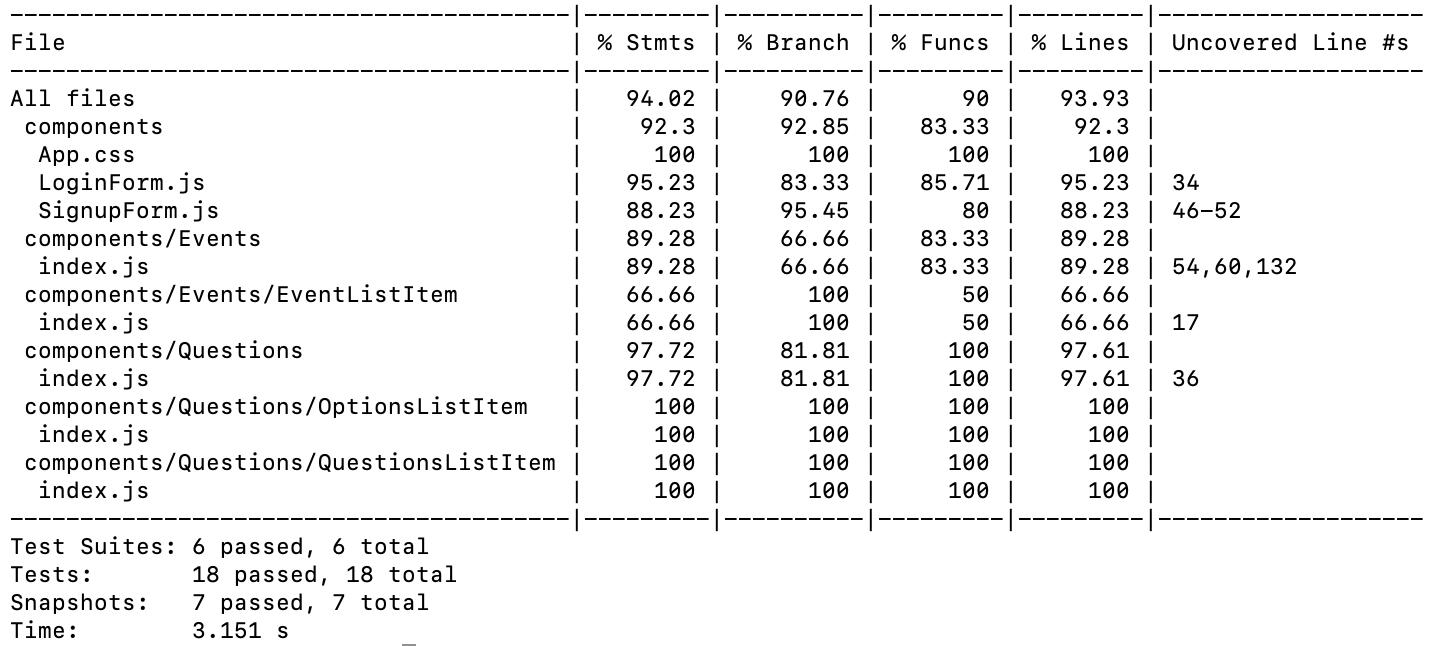
1. Git clone the backend repo
   1. git clone https://github.com/Privilege-walk/back-end.git
2. Install requirements
   1. Look at the Running instructions mentioned in ReadMe at <https://github.com/Privilege-walk/back-end.git>
   2. Once you create and activate virtualenv install the requirements using command
      1. pip3 install -r requirements.txt
3. Run BDDs
   1. We used ‘behave’ to create BDDs for the features and scenarios as we are developing backend in Django
   2. The BDDs can be run at once by using below command:
      1. python manage.py behave
4. Run TDDs
   1. All the TDDs for all apis scenarios can be tested using the command
      1. python manage.py test

For more detailed explanation on instructions to run tests for backend are mentioned in ReadMe of the repo: <https://github.com/Privilege-walk/back-end.git>

## FRONTEND

**TDDs:**

**Coverage**: TDDs frontend test coverage

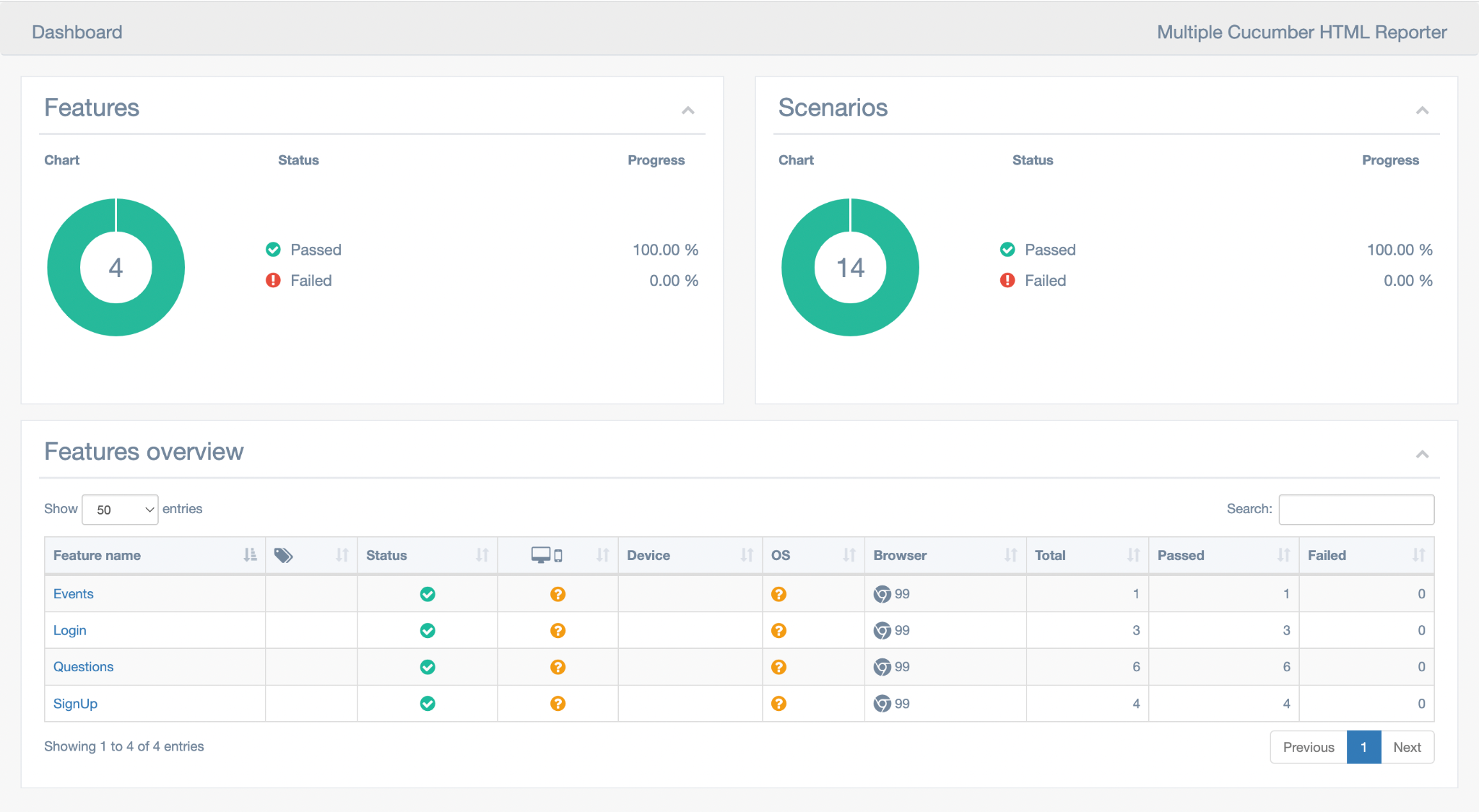


**Results:** for TDDs



**BDDs**

**Results:**

****

**STEPS**

Steps to Run frontend tests:

1. Clone the repository and switch to the root of the repository.
   1. git clone <https://github.com/Privilege-walk/front-end.git>
   2. cd front-end
2. Install npm packages.
   1. npm install
3. RUN TDDs.
   1. npm run test
4. RUN BDDs
   1. Make sure frontend is up by running.
      1. npm start
   2. Make sure the back-end is running by (I’m assuming you’ve cloned the backend repository and installed requirements. See steps 1 and 2 below in “Steps to Run backend tests”):
      1. python manage.py runserver
   3. Run BDD tests by running.
      1. npm run test-bdd

For more detailed steps please checkout README for frontend: https://github.com/Privilege-walk/front-end/blob/main/README.md

# LINKS

Pivotal Tracker: <https://www.pivotaltracker.com/n/projects/2556087>

GitHub Repo Backend: <https://github.com/Privilege-walk/back-end>

GitHub Repo Frontend: <https://github.com/Privilege-walk/front-end>  
GitHub Repo Project: <https://github.com/Privilege-walk/privilege-walk>  
Heroku Deployment backend(old): <https://privilegewalkbe.herokuapp.com/>  
Heroku Deployment project: <https://privilegewalk.herokuapp.com/>

Backend deployed in AWS and is available at: <http://54.157.248.16:8000/>  
Code Climate Frontend: <https://codeclimate.com/github/Privilege-walk/front-end>

Code Climate Backend: <https://codeclimate.com/github/Privilege-walk/back-end>