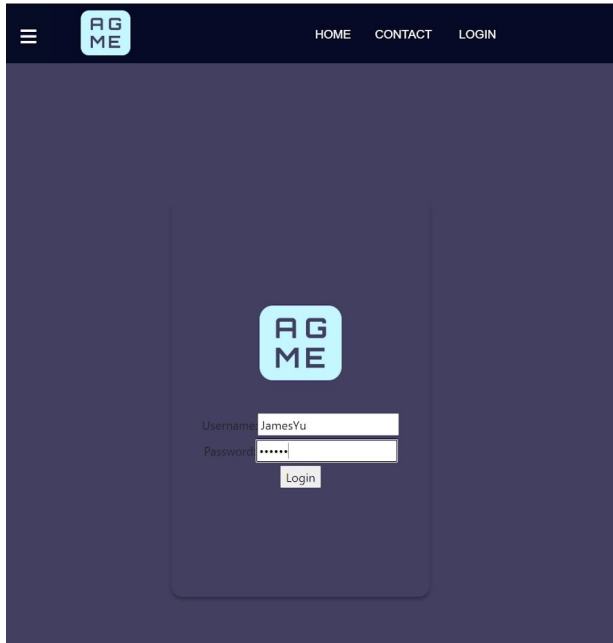
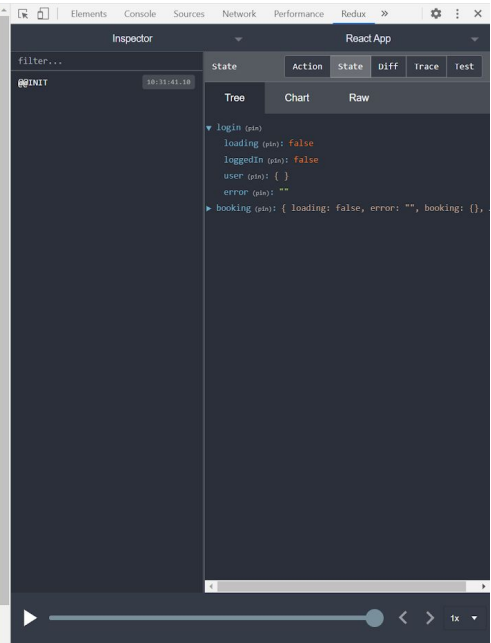
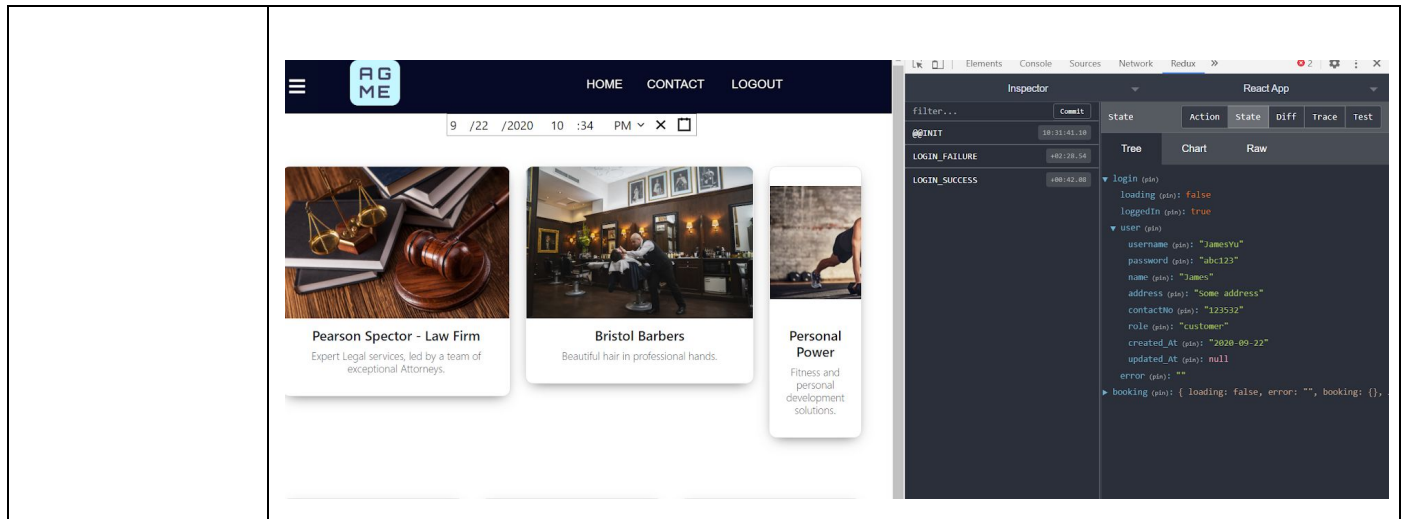


Acceptance Testing for logging in

Test Case #1: Customer User Logs in with Valid Credentials	
Test Case Description:	Testing logging in functionality with an existing customer user and correct credentials
Date/Time of Testing:	21/09/2020 - 22:20
Setup:	<ul style="list-style-type: none"> A customer user called "JamesYu" already exists in the database <ul style="list-style-type: none"> Username: "JamesYu" Password: "abc123!" Existing user has basic information just for bare requirements of a user Starting at initial home page of the service
Steps/Procedure:	<ol style="list-style-type: none"> Select and Navigate to Log In Page Input credentials (Username + Password) Select the 'Submit' option
Expected Result:	User is able to log in and is taken to the customer dashboard
Actual Result:	User is logged in and redirected to the home/dashboard page
Bugs Found:	None
Screenshots	<p>Before logging in</p>  <p>After logging in</p> 



Test Case #2: Customer User Logs in with Invalid Credentials	
Test Case Description:	Testing logging in functionality with a customer user inputting incorrect credentials
Date/Time of Testing:	21/09/2020 - 22:30
Setup:	<ul style="list-style-type: none"> Starting at initial home page of the service The following credentials combination DOES NOT exist in database (hashed or otherwise) <ul style="list-style-type: none"> Username: "JamesYu" Password: "hello"
Steps/Procedure:	<ol style="list-style-type: none"> Select and Navigate to Log In Page Input credentials (Username + Password) Select the 'Submit' option
Expected Result:	User is refused login, prompted with an incorrect username and password message, and can try again at inputting credentials.
Actual Result:	User is unable to log in
Bugs Found:	Login failure message is not displayed to user
Screenshots	Failing to log in

AGME

Username

Password

Login

```
next state ▶ {login: [-], booking: [-]} redux-logger.js:1
▼ action undefined @ 22:39:27.148 redux-logger.js:1
  prev state ▶ {login: [-], booking: [-]} redux-logger.js:1
  action () => { redux-logger.js:1
    return {
      type: _loginTypes__WEBPACK_IMPORTED_MODULE_1["LOGIN_REQUEST"]
    }
  }
  next state ▶ {login: [-], booking: [-]} redux-logger.js:1
  ▶ POST http://localhost:8888/api/login 400 xhr.js:178
  ▼ action LOGIN_FAILURE @ 22:39:27.186 redux-logger.js:1
    prev state ▶ {login: [-], booking: [-]} redux-logger.js:1
    action {type: "LOGIN_FAILURE", payload: "Request failed with status code 400"} redux-logger.js:1
    next state ▶ {login: [-], booking: [-]} redux-logger.js:1
    Request failed with status code 400 loginActions.js:48
    Login failed loginActions.js:49
  \
```