Changes made in code:

Login:
Lock out after too many wrong attempts:

```
    if (!user) {
      res.status(400).json({ error: 'Invalid username or password' });
      return;
    }

    if (user.lockoutUntil && new Date(user.lockoutUntil) > new Date()) {
      res.status(400).json({ error: 'Account is locked. Try again later.' });
      return;
    }

    const isPasswordValid = await bcrypt.compare(sanitizedPassword, user.password);

    if (!isPasswordValid) {
      const failedLoginAttempts = user.failedLoginAttempts + 1;
      const lockoutUntil = failedLoginAttempts >= 3 ? new Date(new Date().getTime() + 15 *
60000) : null; // Lock password for 15mins

      await usersCollection.updateOne({ _id: user._id }, {
        $set: { failedLoginAttempts, lockoutUntil }
      });

      res.status(400).json({ error: 'Invalid username or password' });
      return;
    }
```

Valid input length and prevent null/blank:

```
  // Validate input - Ensure fields are not blank and match expected format
  if (!username || !password) {
    res.status(400).json({ error: 'All fields are required' });
    return;
  }

  // Validate input length
  if (username.length > 50 || password.length > 50) {
    res.status(400).json({ error: 'Fields must be less than 50 characters' });
    return; }
```

Sanitize input:

```
const sanitizedData = {
  username: sanitizeHtml(formData.username),
  password: sanitizeHtml(formData.password),
};
const JSONdata = JSON.stringify(sanitizedData);
const endpoint = '/api/login';
const options = {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSONdata,
};
```

Register:

Length + validation checks:

```
if (!formData.username || formData.username.length > 50) {
  setError('Username is required and must be less than 50 characters.');
  return;
}
if (!formData.password || formData.password.length > 50) {
  setError('Password is required and must be less than 50 characters.');
  return;
}

// Ensure fields are not blank and apply validation
if (!formData.username) {
  setError('Username is required.');
  return;
}
if (!formData.password) {
  setError('Password is required.');
  return;
}
if (!formData.email) {
  setError('Email is required.');
  return;
}
```

Make sure its a valid email using email-validator

```
import validator from 'email-validator';

// Validate email format using email validator
if (!validator.validate(formData.email)) {
```

```
    setError('Invalid email format!');
    return;
  }
```

Sanitize user data;

```
  const sanitizedData = {
    username: sanitizeHtml(formData.username),
    password: sanitizeHtml(formData.password),
    email: sanitizeHtml(formData.email),
    type: 'customer' // default
  };
```

Manager.js

Prevent login for non manager user types:

```
  if (!user || user.type !== 'manager') {
    res.status(401).json({ error: 'Unauthorized' });
    return;
  }
```

Submit_order.js

Validate input - Ensure fields are not blank and match expected format

```
  if (!cartItems || !userDetails || !total) {
    res.status(400).json({ error: 'All fields are required' });
    return;
  }

  const {
    fullName, email, address, city, county, eircode, cardNumber, expirationDate, cvc,
  } = userDetails;

  // Validate email format
  if (!validator.validate(email)) {
    res.status(400).json({ error: 'Invalid email format' });
    return;
  }

  // Ensure no fields are empty
  if (!fullName || !email || !address || !city || !county || !eircode || !cardNumber || !expirationDate ||
!cvc) {
    res.status(400).json({ error: 'All fields are required' });
    return;
```

```
  }
```

Sanitize input and validate length

```
  const sanitizedCartItems = cartItems.map(item => ({
    ...item,
    productid: sanitizeHtml(item.productid).substring(0, 50),
    title: sanitizeHtml(item.title).substring(0, 50),
    description: sanitizeHtml(item.description).substring(0, 50),
    price: parseFloat(sanitizeHtml(item.price)),
    images: sanitizeHtml(item.images).substring(0, 50),
    quantity: parseInt(sanitizeHtml(item.quantity), 10),
  }));

  const sanitizedUserDetails = {
    fullName: sanitizeHtml(fullName).substring(0, 50),
    email: sanitizeHtml(email).substring(0, 50),
    address: sanitizeHtml(address).substring(0, 50),
    city: sanitizeHtml(city).substring(0, 50),
    county: sanitizeHtml(county).substring(0, 50),
    eircode: sanitizeHtml(eircode).substring(0, 50),
    cardNumber: sanitizeHtml(cardNumber).substring(0, 50),
    expirationDate: sanitizeHtml(expirationDate).substring(0, 50),
    cvc: sanitizeHtml(cvc).substring(0, 50),
  };
```

Password encryption:

(register + login api)

```
import bcrypt from 'bcrypt';
```

Hashing password on registration:

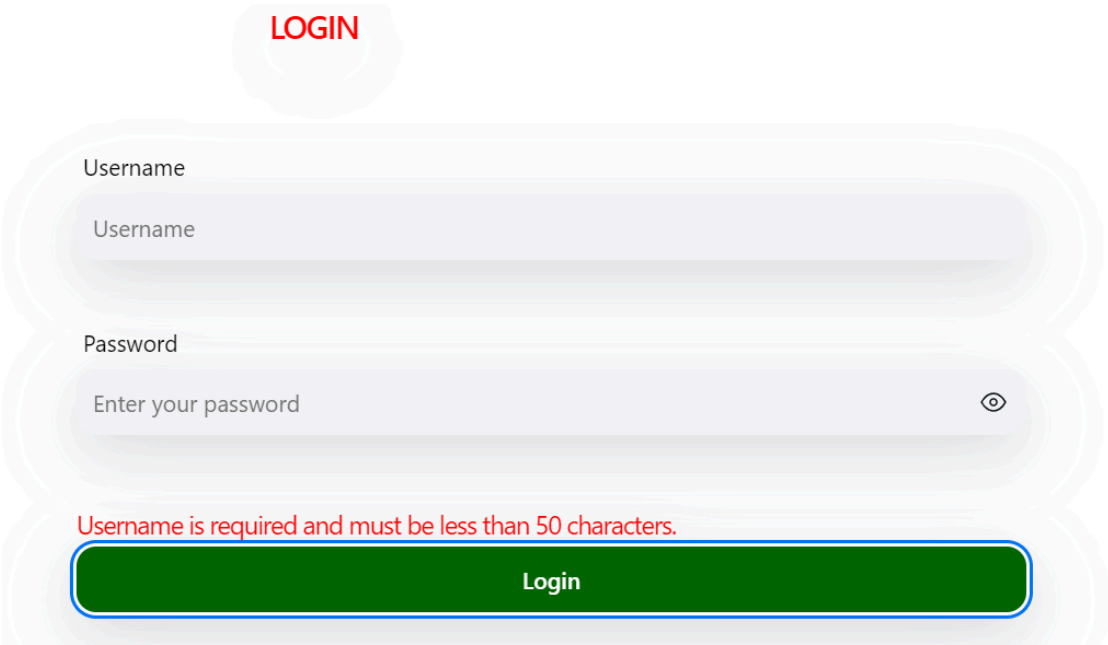
```
    const hashedPassword = await bcrypt.hash(sanitizedPassword, 10);
```

Comparing password on login to the user input and the encrypted password on the database:

```
    const isPasswordValid = await bcrypt.compare(sanitizedPassword, user.password);
```

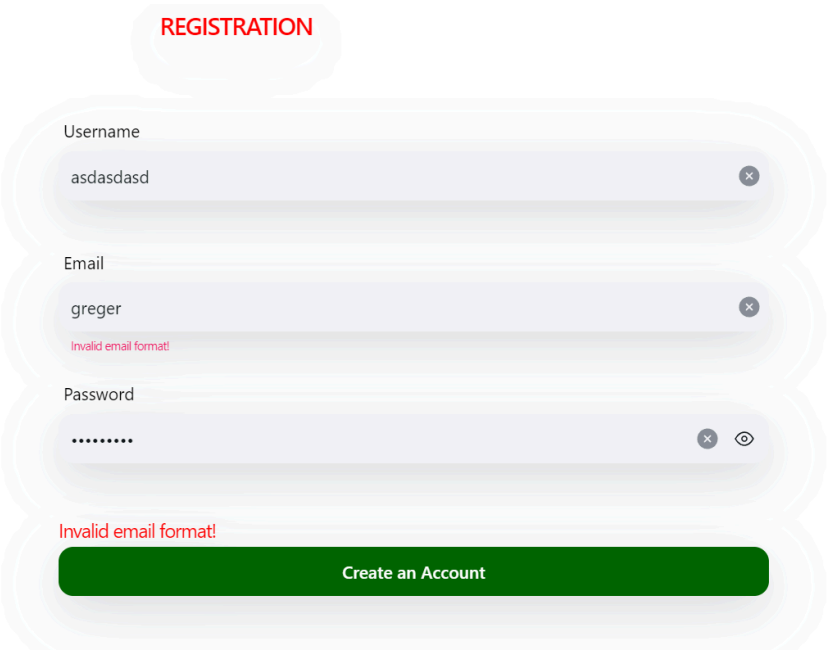
VALIDATION

1. **Limit the number of characters that can be inserted into any text field (client and server-side)**



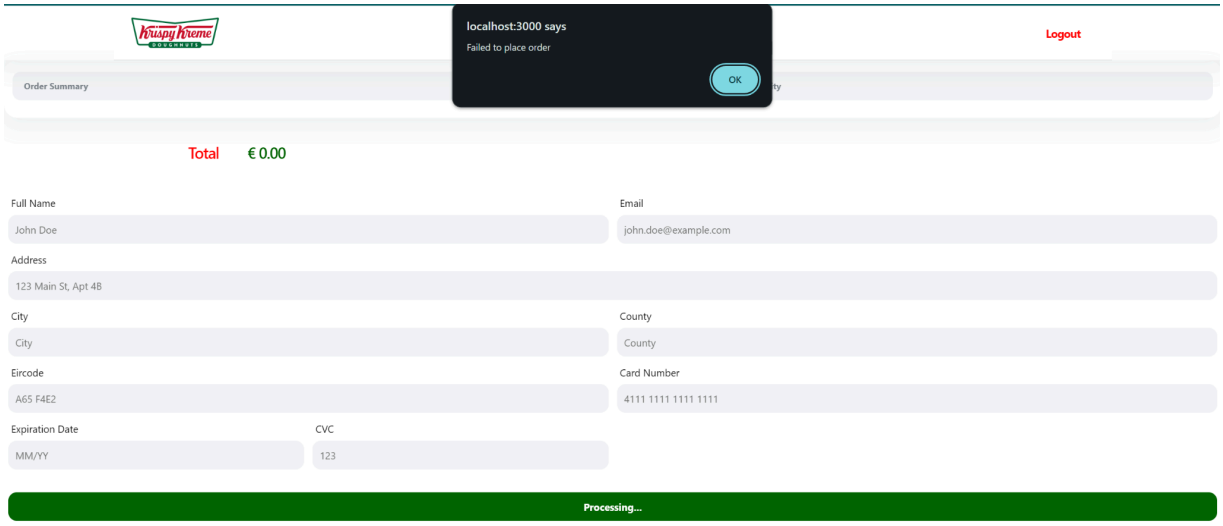
(Added to all other major text fields too)

2. **Check to ensure the content that has been entered into the field matches what is expected, e.g., an email field should only allow for an email address to be entered.**

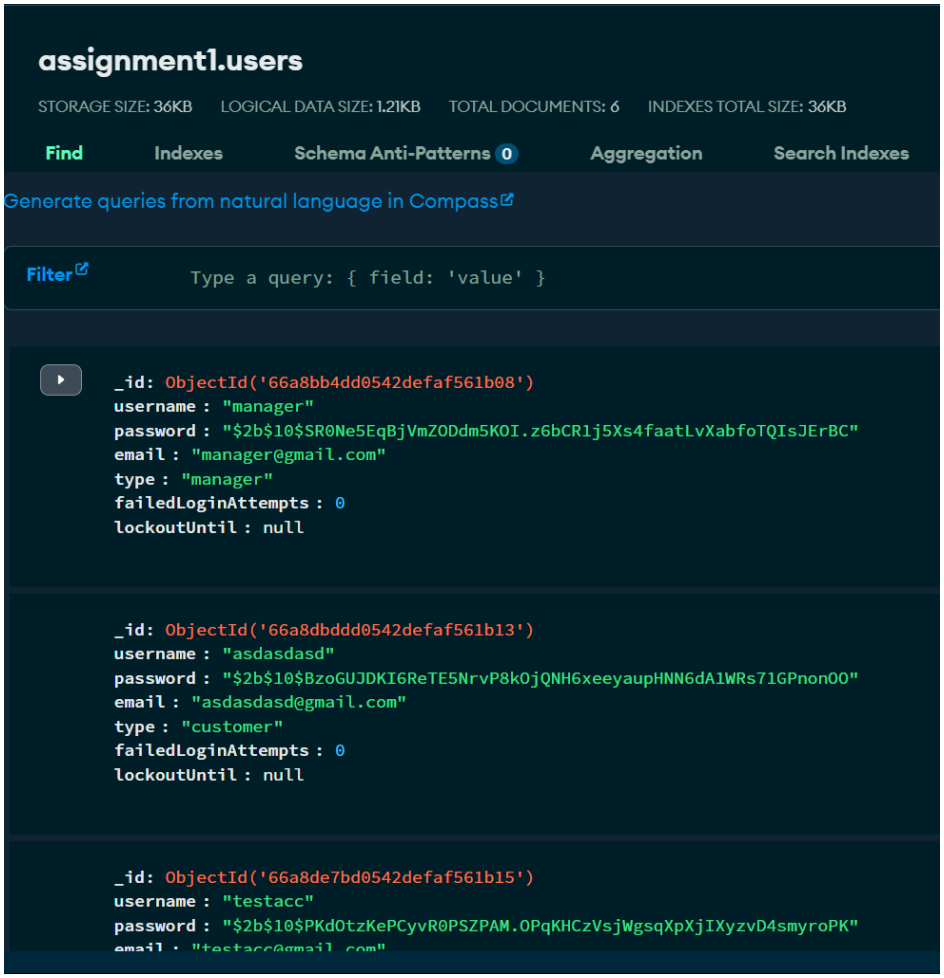


(Added to all other text fields too, like fields that only accept numbers)

3. **The user should never be able to enter blank values including NULL.**



4. **Although it is possible to store plain text in a database, this is never the best solution. When a user submits a password, it should be hashed using bcrypt and checked against a hash for the password that is stored in the database.**



5. **In any area where user input is added, validation should be added to ensure it is escaped.**

6. **To prevent malicious injection, a combination of sanitization approaches should be used on the user input.**

```
const sanitizedUsername = sanitizeHtml(username);
const sanitizedEmail = sanitizeHtml(email);
const sanitizedPassword = sanitizeHtml(password);
const sanitizedType = sanitizeHtml(type || 'customer');
```

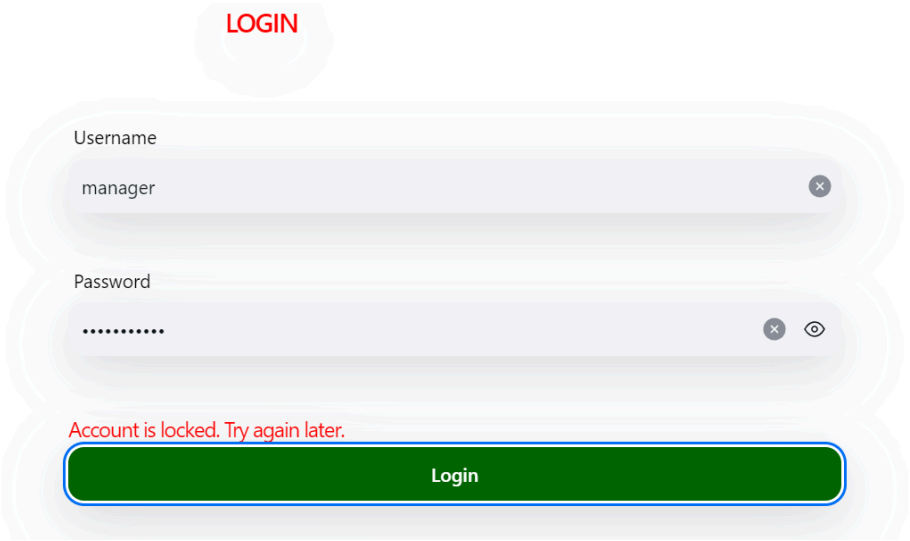
(Applied everywhere with user input)

7. **Users should not be able to access areas of the application without logging in.**

```
if (!user || user.type !== 'manager') {
  res.status(401).json({ error: 'Unauthorized' });
  return;
}
```

(Same with menu page, will redirect them to login page)

8. **A user may attempt to log in multiple times with the incorrect password. Implement a time-based approach to prevent brute-force attacks.**

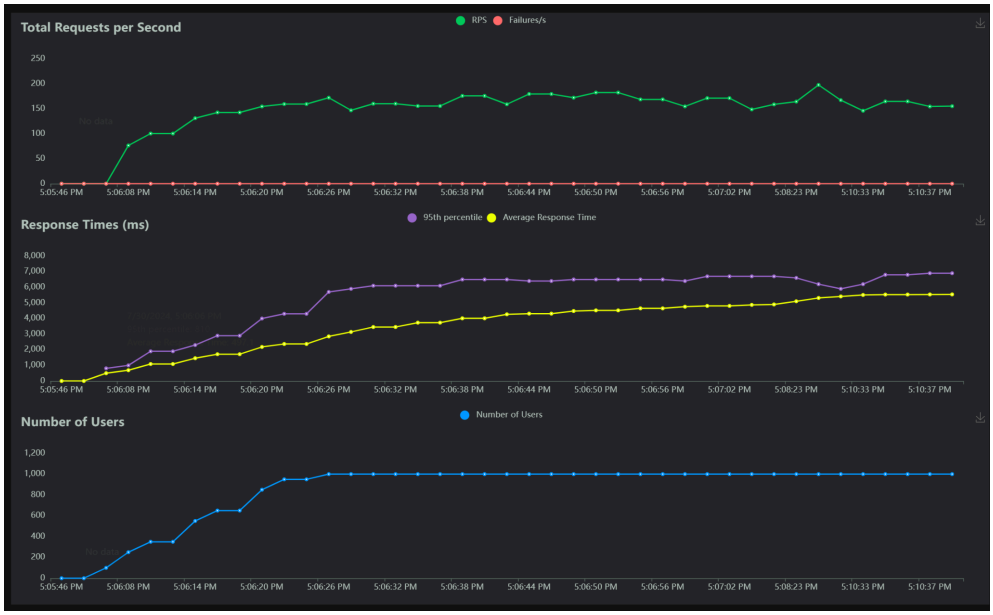


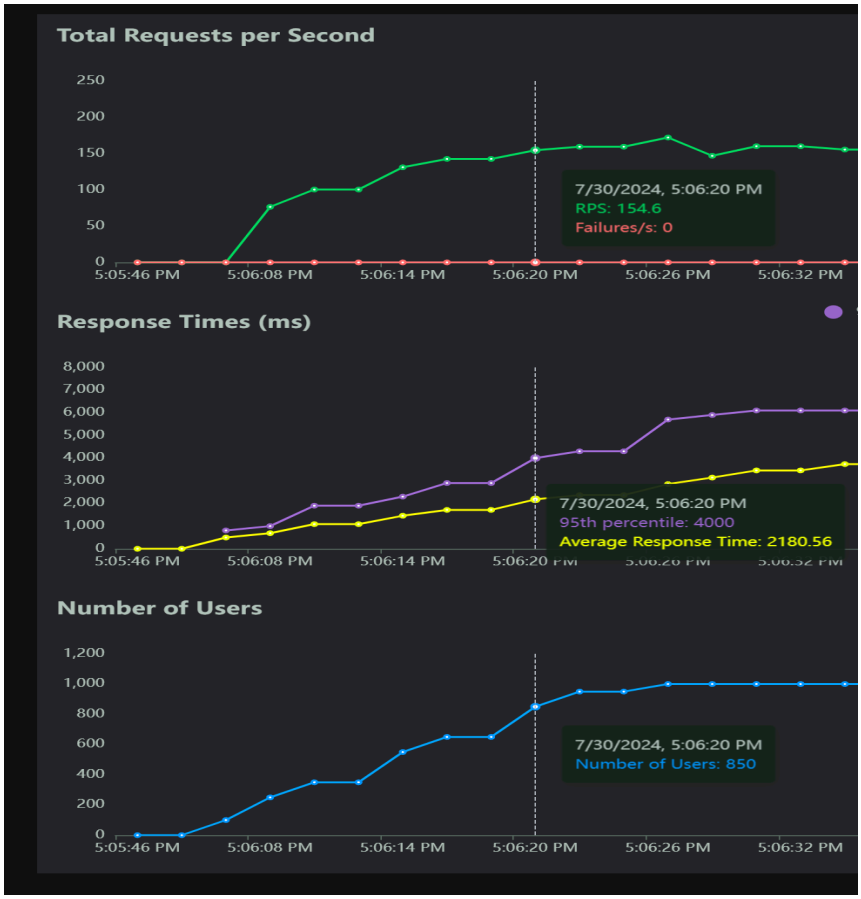

```
if (user.lockoutUntil && new Date(user.lockoutUntil) > new Date()) {
  res.status(400).json({ error: 'Account is locked. Try again later.' });
  return;
}

const isPasswordValid = await bcrypt.compare(sanitizedPassword, user.password);

if (!isPasswordValid) {
  const failedLoginAttempts = user.failedLoginAttempts + 1;
  const lockoutUntil = failedLoginAttempts >= 3 ? new Date(new Date().getTime() + 15 *
60000) : null; // Lock password for 15mins
```

STRESS TEST





As we can see from the graphs, the page started to take around 2 seconds to load at around 850 users

REGISTER

Project: http://localhost:3000/login

Tests

Search tests...

- ✓ 1_login
- ✗ 2_menu page
- ✓ 3_continue to checkout
- ✓ 4_checkout
- ✓ register

http://localhost:3000/register

	Command	Target	Value
1	✓ *open*	http://localhost:3000/register	
2	✓ *set window size*	1708x1068	
3	✓ *click*	id=username	
4	✓ *type*	id=username	testacc
5	✓ *type*	id=email	testacc@gmail.com
6	✓ *type*	id=password	45678
7	✓ *click*	css=.nextui-button--hovered	

Command

Target

Value

Description

Log Reference

2. setWindowSize on 1708x1068 OK
3. click on id=username OK
4. type on id=username with value testacc OK
5. type on id=email with value testacc@gmail.com OK
6. type on id=password with value 45678 OK
7. Trying to find css=.nextui-button--hovered... OK

Warning Element found with secondary locator xpath=//button[@type='submit']. To use it by default, update the test step to use it as the primary locator.

'register' completed successfully

LOGIN

Project: http://localhost:3000/login

Tests

Search tests...

- ✓ 1_login
- ✗ 2_menu page
- ✓ 3_continue to checkout
- ✓ 4_checkout
- ✓ register

http://localhost:3000/register

	Command	Target	Value
		extui-c-eFfoBo	
4	✓ *click*	id=username	
5	✓ *type*	id=username	asdasdasd
6	✓ *type*	id=password	po'$"xti92txpit949
7	✓ *mouse over*	css=.nextui-c-eFfoBo > .nextui-c-iWjDFM	
8	✓ *click*	css=.nextui-button--hovered	

Command

Target

Value

Description

Log Reference

6. type on id=password with value 45078 OK	13:39:13
7. Trying to find css=.nextui-button--hovered... OK	13:39:13
'register' completed successfully	13:39:22
Running '1_login'	15:12:36
1. open on /login OK	15:12:37
2. setWindowSize on 1708x1068 OK	15:12:37
3. click on css=.nextui-c-dfbgCO:nth-child(2) .nextui-c-eFfoBo OK	15:12:37
4. click on id=username OK	15:12:38
5. type on id=username with value asdasdasd OK	15:12:38
6. type on id=password with value po'$"xti92txpit949 OK	15:12:38
7. mouseOver on css=.nextui-c-eFfoBo > .nextui-c-iWjDFM OK	15:12:38
8. click on css=.nextui-button--hovered OK	15:12:38
9. mouseOut on css=.nextui-button--hovered OK	15:12:38
'1_login' completed successfully	15:12:38

MENU PAGE

Selenium IDE - http://localhost:3000/login*

Project: http://localhost:3000/login*

Tests

Search tests...

- ✓ 1_login
- ✓ 2_menu_page*
- ✓ 3_continue to checkout
- ✓ 4_checkout
- ✓ register

http://localhost:3000/customer

	Command	Target	Value
1	✓ open	http://localhost:3000/customer	
2	✓ set window size	1708x1068	
3	✓ click	css=.nextui-button--hovered	
4	✓ click	css=.nextui-c-kRHeuF:nth-child(1) .nextui-c-cakKrd	
5	✓ click	css=.nextui-button--hovered	
6	✓ click	css=.nextui-c-lklfrF > .nextui-c-hnqGNA	

Command

Target

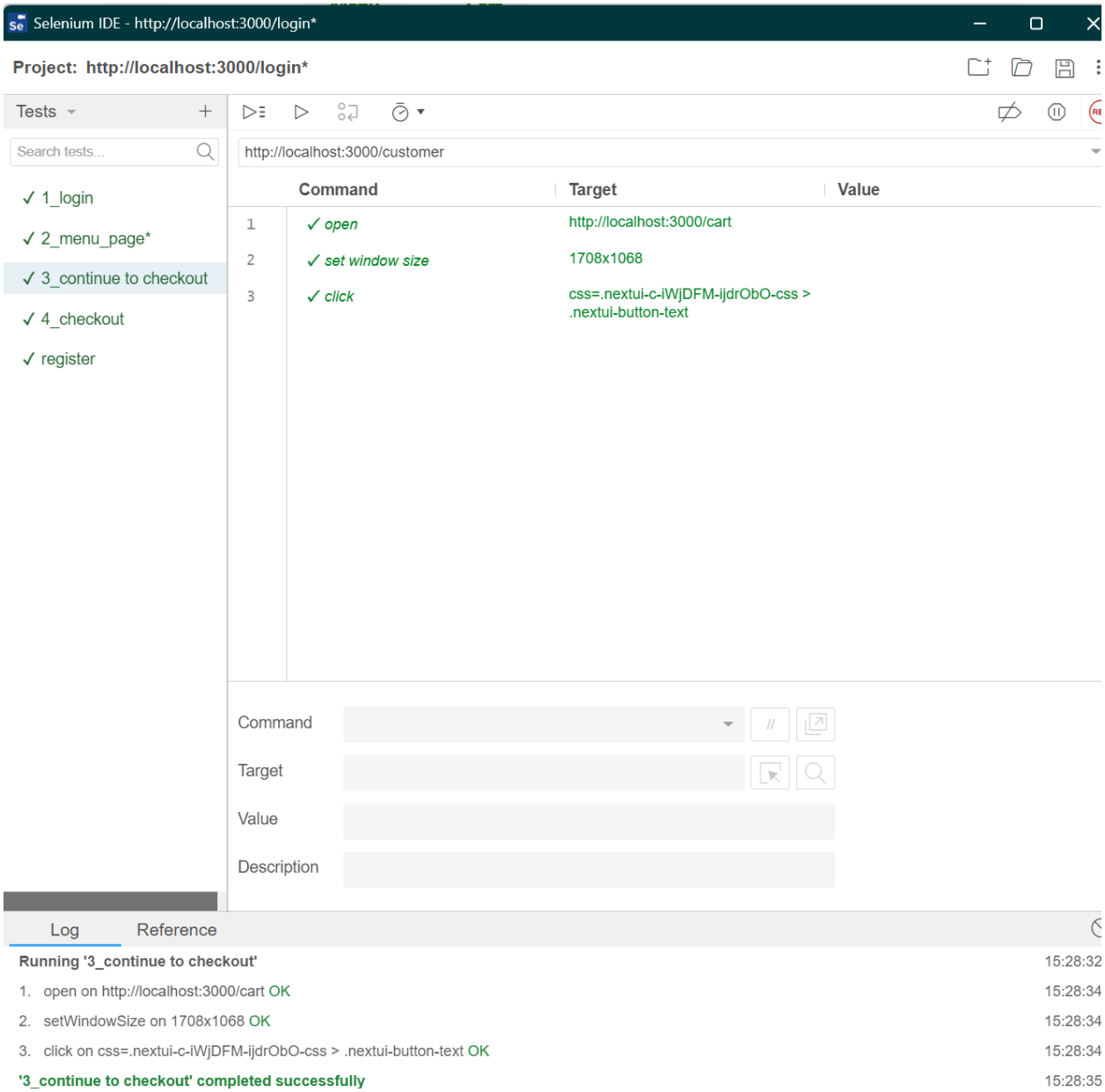
Value

Description

Log Reference

Running '2_menu_page'	15:25:03
1. open on http://localhost:3000/customer OK	15:25:04
2. setWindowSize on 1708x1068 OK	15:25:04
3. Trying to find css=.nextui-button--hovered... OK	15:25:04
Warning Element found with secondary locator xpath=//button[@type='submit']. To use it by default, update the test step to use it as the primary locator.	15:25:34
4. click on css=.nextui-c-kRHeuF:nth-child(1) .nextui-c-cakKrd OK	15:25:34
5. Trying to find css=.nextui-button--hovered... OK	15:25:35
Warning Element found with secondary locator xpath=//button[@type='submit']. To use it by default, update the test step to use it as the primary locator.	15:26:05
6. click on css=.nextui-c-lklfrF > .nextui-c-hnqGNA OK	15:26:05
'2_menu_page' completed successfully	15:26:05

ADD TO CHECKOUT



CHECKOUT

Project: http://localhost:3000/login*

Tests

Search tests...

- ✓ 1_login
- ✓ 2_menu_page*
- ✓ 3_continue to checkout
- ✓ 41_checkout*
- ✓ register

http://localhost:3000/checkout

	Command	Target	Value
1	✓ *open*	http://localhost:3000/checkout	
2	✓ *set window size*	1708x1068	
3	✓ *click*	id=fullName	
4	✓ *type*	id=fullName	John Doe
5	✓ *type*	id=email	johndoe@gmail.com

Command: click

Target: css=.nextui-c-kRHeuF:nth-child(10)

Value:

Description:

Log | Reference

Running '41_checkout' 15:40:14

1. open on http://localhost:3000/checkout OK 15:40:16
2. setWindowSize on 1708x1068 OK 15:40:16
3. click on id=fullName OK 15:40:16
4. type on id=fullName with value John Doe OK 15:40:17
5. type on id=email with value johndoe@gmail.com OK 15:40:18
6. type on id=address with value 30 Street OK 15:40:19
7. type on id=city with value Blanchardstown OK 15:40:19
8. type on id=county with value Dublin 15 OK 15:40:19
9. type on id=eircode with value d19b2cg OK 15:40:19
10. type on id=cardNumber with value 41111 1111 1111 OK 15:40:20
11. type on id=expirationDate with value 08/27 OK 15:40:20
12. type on id=cvc with value 213 OK 15:40:20
13. click on css=.nextui-c-kRHeuF:nth-child(10) OK 15:40:20

'41_checkout' completed successfully 15:40:20

REMOVE DONUT FROM CART

Project: http://localhost:3000/login*

Tests

Search tests...

- ✓ 1_login
- ✓ 2_menu_page*
- ✓ 3_continue to checkout
- ✓ 5_remove_cart*
- ✓ 41_checkout*
- ✓ register

http://localhost:3000/cart

	Command	Target	Value
1	✓ *open*	http://localhost:3000/cart	
2	✓ *set window size*	1708x1068	
3	✓ *mouse over*	css=svg	
4	✓ *mouse out*	css=svg	
5	✓ *click*	css=svg	

Command

Target

Value

Description

Log | Reference

Running '5_remove_cart'	15:50:54
1. open on http://localhost:3000/cart OK	15:50:55
2. setWindowSize on 1708x1068 OK	15:50:55
3. mouseOver on css=svg OK	15:50:55
4. mouseOut on css=svg OK	15:50:56
5. click on css=svg OK	15:50:57
'5_remove_cart' completed successfully	15:50:57