CS 731 - Software Testing

Client-Side Web Application Testing (Bypass Testing)

towards partial fulfillment

of the requirements for the course of

CS 731 Software Testing

Source Project: Brain Health Initiative

Jayprakash Ray
[MT2021060]

Khushal Abrol
[MT2021063]

Tools Used:
IDE: IntelliJ IDEA
Development Framework:
-Spring Boot (Backend) and Angular (Frontend)
Testing Tool: Selenium

Testing Strategy Used: **Client-side web applications testing (bypass testing)**

Repository Links:

- Frontend
- Backend

Bypass Testing:

[1] Web software applications are increasingly being deployed in sensitive situations. Web applications are used to transmit, accept and store data that is personal, company confidential and sensitive. Input validation testing (IVT) checks user inputs to ensure that they conform to the program's requirements, which is particularly important for software that relies on user inputs, including Web applications. A common technique in Web applications is to perform input validation on the client with scripting languages such as JavaScript. An insidious problem with client-side input validation is that end users can bypass this validation. Bypassing validation can cause failures in the software, and can also break the security on Web applications, leading to unauthorized access to data, system failures, invalid purchases and entry of bogus data. We are developing a strategy called bypass testing to create client-side tests for Web applications that intentionally violate explicit and implicit checks on user inputs. This paper describes the strategy, defines specific rules and adequacy criteria for tests, describes a proof-of-concept automated tool, and presents initial empirical results from applying bypass testing.

Source Code Description:

A web platform for assisting doctors in screening and diagnosis of neurological disorders built using Angular in frontend and Spring Boot in Backend with security features such as JWT Authorization and Role Based Access Control using AUTH Guard.

Feature Testing:

- For testing purpose, we have implemented each feature by two ways one with normal flow, to illustrate how it supposed to perform in normal execution
- For the other part we have tried to bypass the form by adding some invalid inputs into the fields and checking whether out functionality detects the error of perform normal execution.

1. Admin

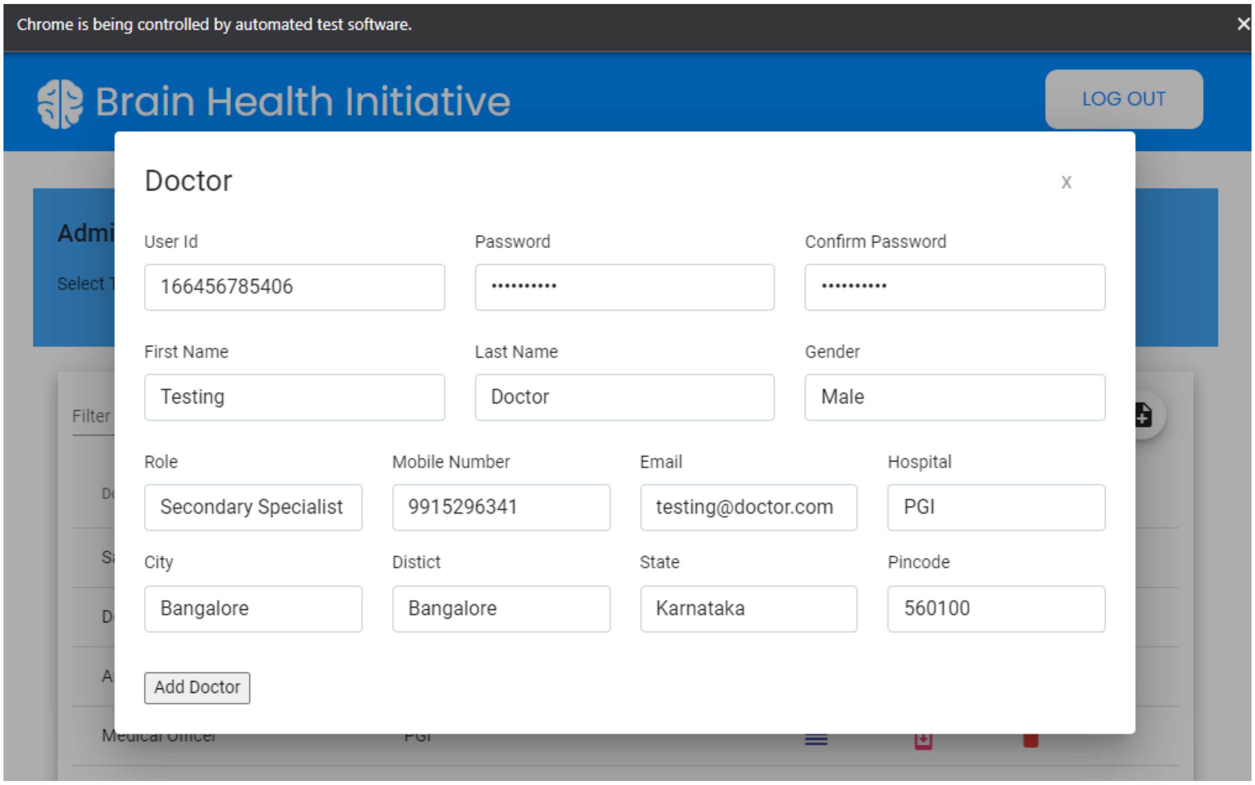
Testing Admin Feature:

- We have various forms and each form contains multiple fields. We took one form to perform bypass testing which is “Add Doctor” form.
- Add doctor form has following fields:

Field Name	**Type**	**Constrains**
UserId	String	12 Digits and unique
Password	String	None
Confirm Password	String	Should be same as Password
First Name	String	None
Last Name	String	None
Gender	Drop Down	None
Role	Drop Down	None
Mobile Number	String	10 Digits
Email	String	Should be in the format: username@mailServer.domain
Hospital	Drop Down	None
City	String	None
Distict	String	None
State	String	None

Pincode	String	6 Digits

- We have created two class to check feature with valid inputs (AdminValidInput) and invalid inputs (AdminInvalidInput).
- In **AdminValidInput** class we have added all the valid inputs and executed the test
- Test execution:.
  - Filled fields with valid inputs.

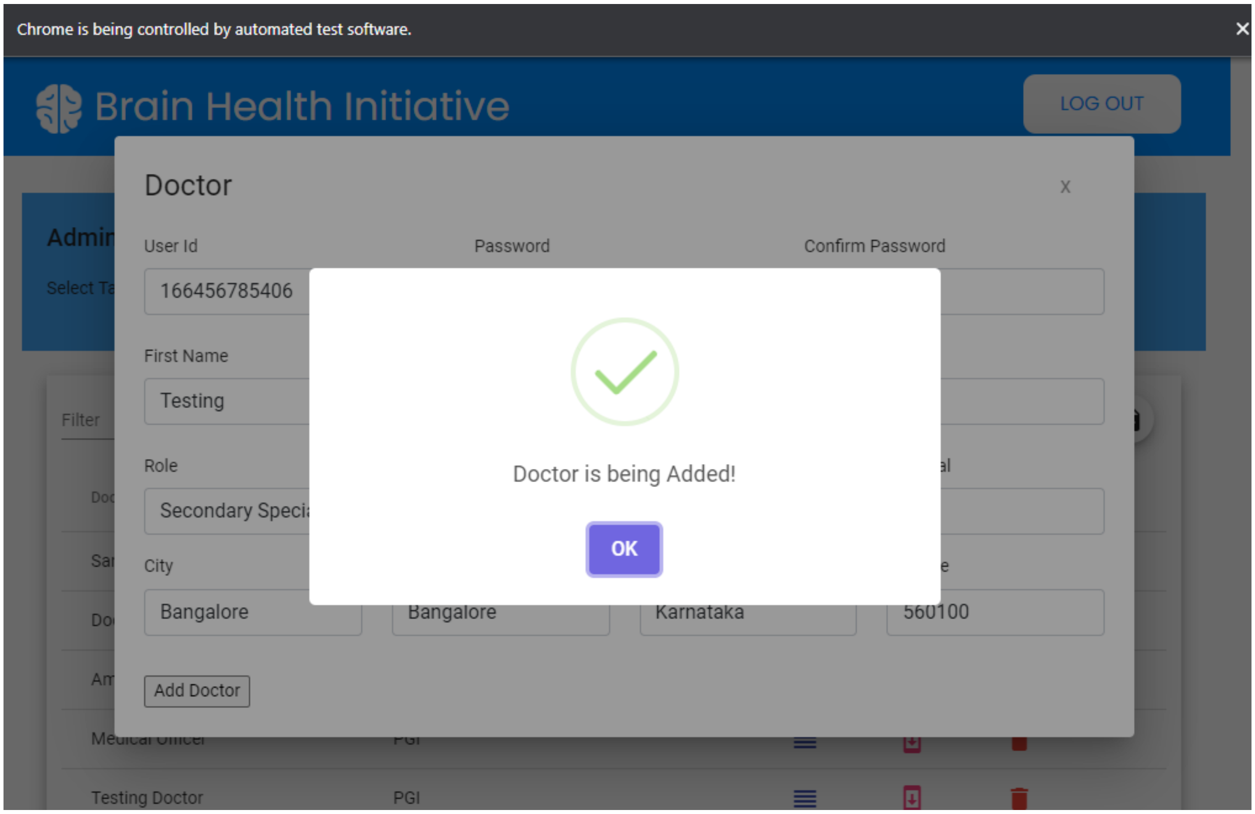


Doctor

User Id	Password	Confirm Password
166456785406	••••••••••	••••••••••
First Name	Last Name	Gender
Testing	Doctor	Male

Role	Mobile Number	Email	Hospital
Secondary Specialist	9915296341	testing@doctor.com	PGI
City	Distict	State	Pincode
Bangalore	Bangalore	Karnataka	560100

  - Form submitted successfully.



Doctor

User Id: 166456785406

Password

Confirm Password

First Name: Testing

Role: Secondary Speci

City: Bangalore

Bangalore

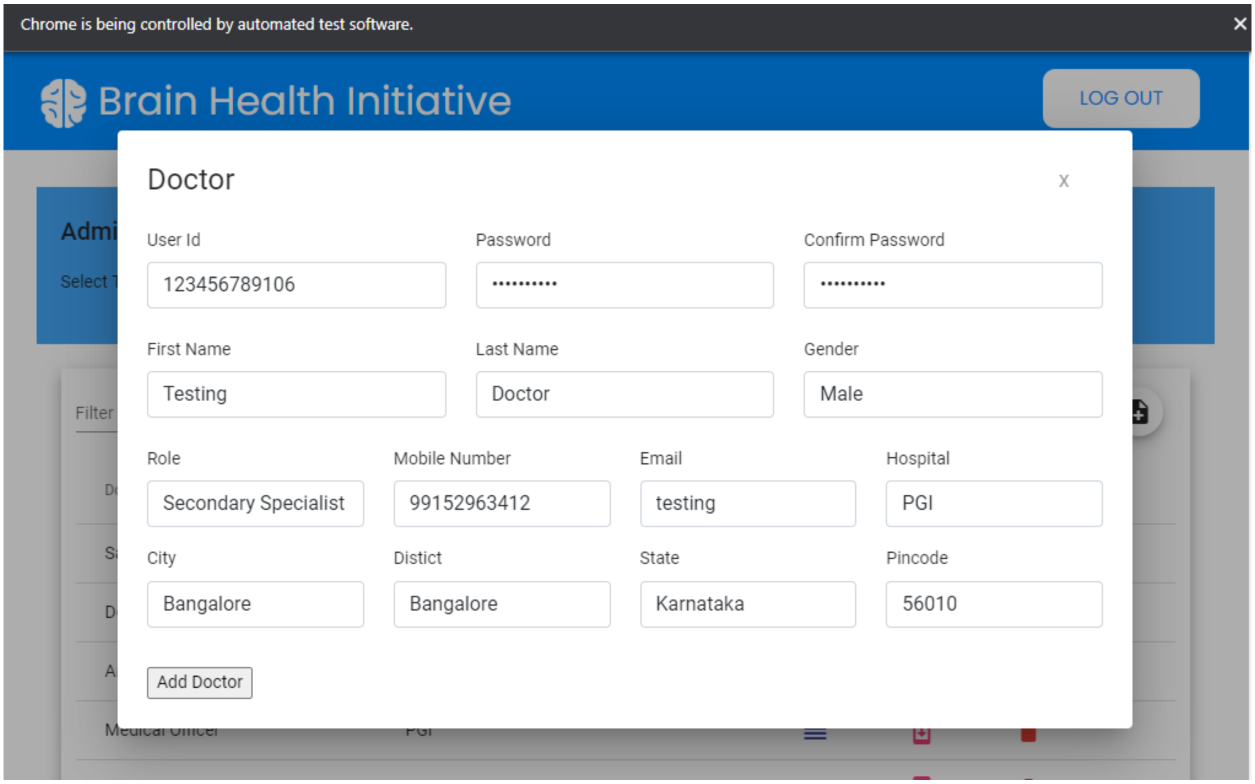
Karnataka

560100

Doctor is being Added!

Conclusion:
When following normal flow and added valid inputs in form fields, form submitted successfully.

- In **AdminInvalidInput** class we have added some invalid inputs and executed the test
- Test execution:
  - Filling some fields with invalid inputs.

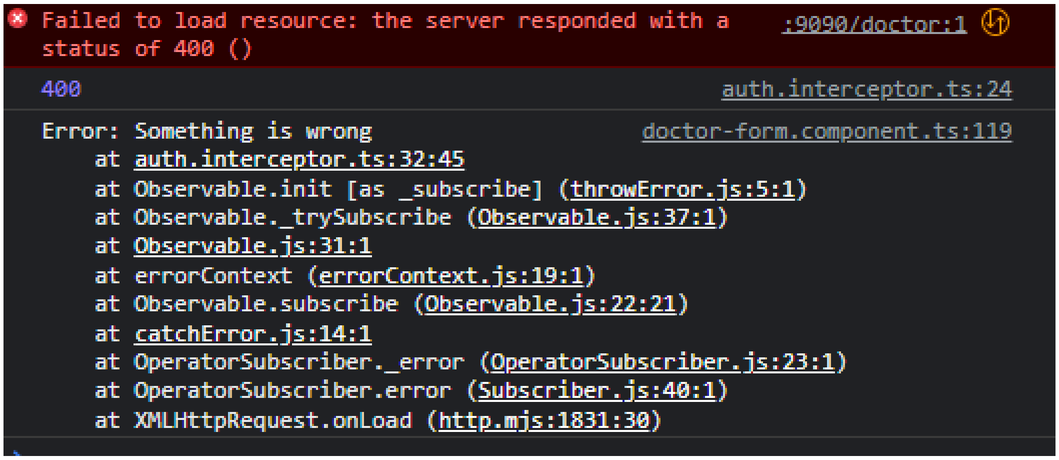


Doctor

User Id	Password	Confirm Password
123456789106	••••••••••	••••••••••
First Name	Last Name	Gender
Testing	Doctor	Male

Role	Mobile Number	Email	Hospital
Secondary Specialist	99152963412	testing	PGI
City	Distict	State	Pincode
Bangalore	Bangalore	Karnataka	56010

- ○ Form not submitted, on UI form remained opened and got submitted but form get rejected at server and server responded with Server error 400 and form is not accepted by the backend



```
Failed to load resource: the server responded with a     :9090/doctor:1
status of 400 ()
400                                                  auth.interceptor.ts:24
Error: Something is wrong                   doctor-form.component.ts:119
    at auth.interceptor.ts:32:45
    at Observable.init [as _subscribe] (throwError.js:5:1)
    at Observable._trySubscribe (Observable.js:37:1)
    at Observable.js:31:1
    at errorContext (errorContext.js:19:1)
    at Observable.subscribe (Observable.js:22:21)
    at catchError.js:14:1
    at OperatorSubscriber._error (OperatorSubscriber.js:23:1)
    at OperatorSubscriber.error (Subscriber.js:40:1)
    at XMLHttpRequest.onLoad (http.mjs:1831:30)
```

Test Result:: Test Passed

- Error is not handled at the frontend, but handled at the backend. So it gives error in the console instead of a pop-up.

2. Consultation Form

Bypass testing of **Submit Consultation Form**:

- This feature lets doctors add a consultation form on a patient's medical visit by entering details of his current medical condition and also can prescribe medicine by filling and submitting the form.

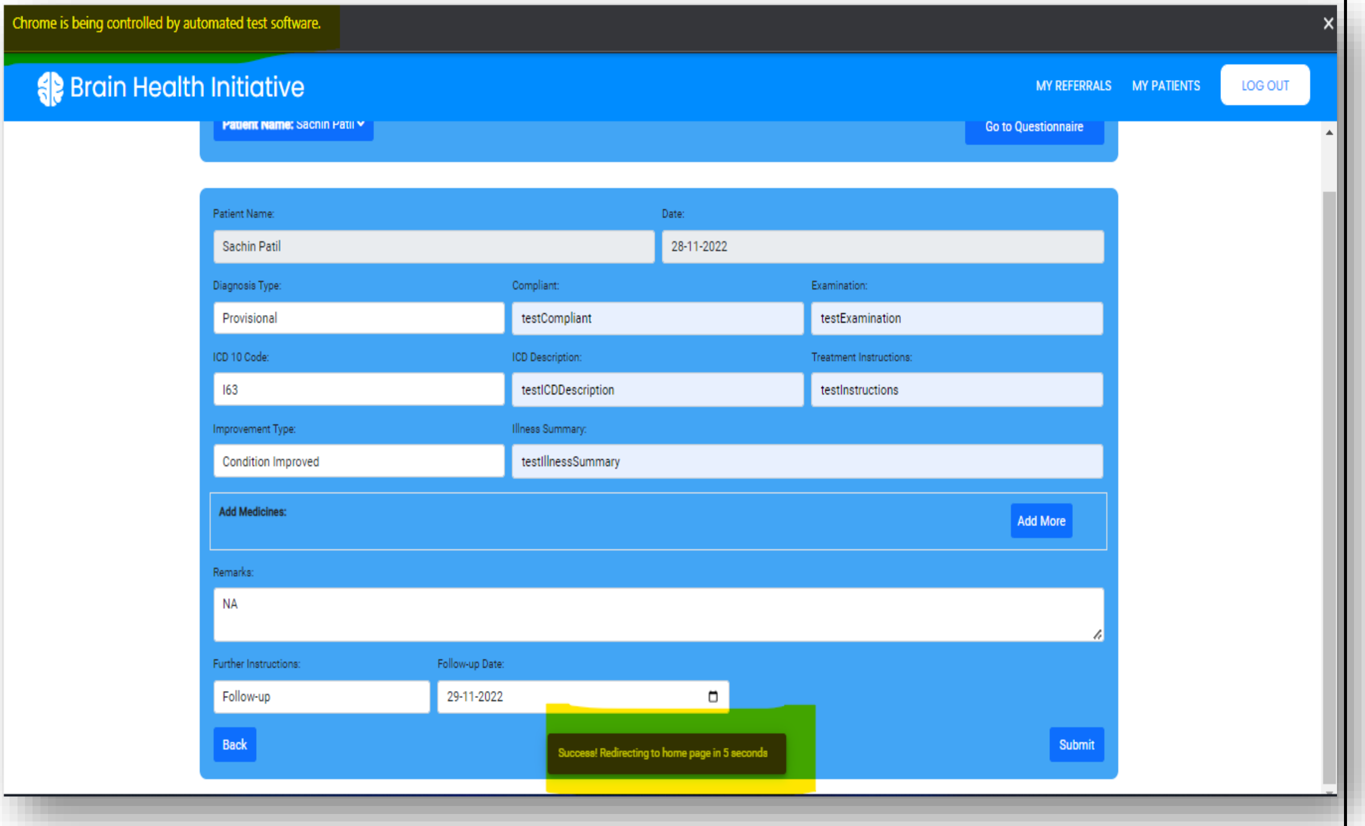
- Consultation form has following fields:

Field Name	**Type**	**Constrains**
Patient Name	String	Read Only
Date	Date	Read Only [Current Date]
Diagnosis Type	Drop Down	None
Compliant	String	Required
Examination	String	None
ICD 10 Code	Drop Down	Required
ICD Description	String	None
Treatment Instruction	String	None
Improvement Type	Drop Down	Required
Illness Summary	Drop Down	None
Medicines	String	None
Remarks	String	None
Further Instructions	String	None
Follow Up Date	Date	Future Date

Testing on Valid Input:

- Add Consultation Form Feature has following Client End Validations:

Attribute	Constraint
Date	Current Date Only [Read Only Field]
Compliant	Required Field
ICD Description	Required Field



Chrome is being controlled by automated test software.

Patient Name: Sachin Patil

Date: 28-11-2022

Diagnosis Type: Provisional

Compliant: testCompliant

Examination: testExamination

ICD 10 Code: I63

ICD Description: testICDDescription

Treatment Instructions: testInstructions

Improvement Type: Condition Improved

Illness Summary: testIllnessSummary

Add Medicines:

Remarks: NA

Further Instructions: Follow-up

Follow-up Date: 29-11-2022

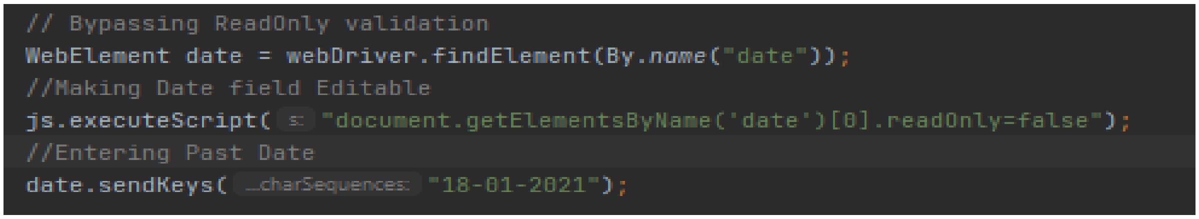
Success! Redirecting to home page in 5 seconds

Here We entered Valid Input which satisfies all Constraint [Refer Above Screenshot]

Bypassing the Constraint to test on Invalid Inputs:

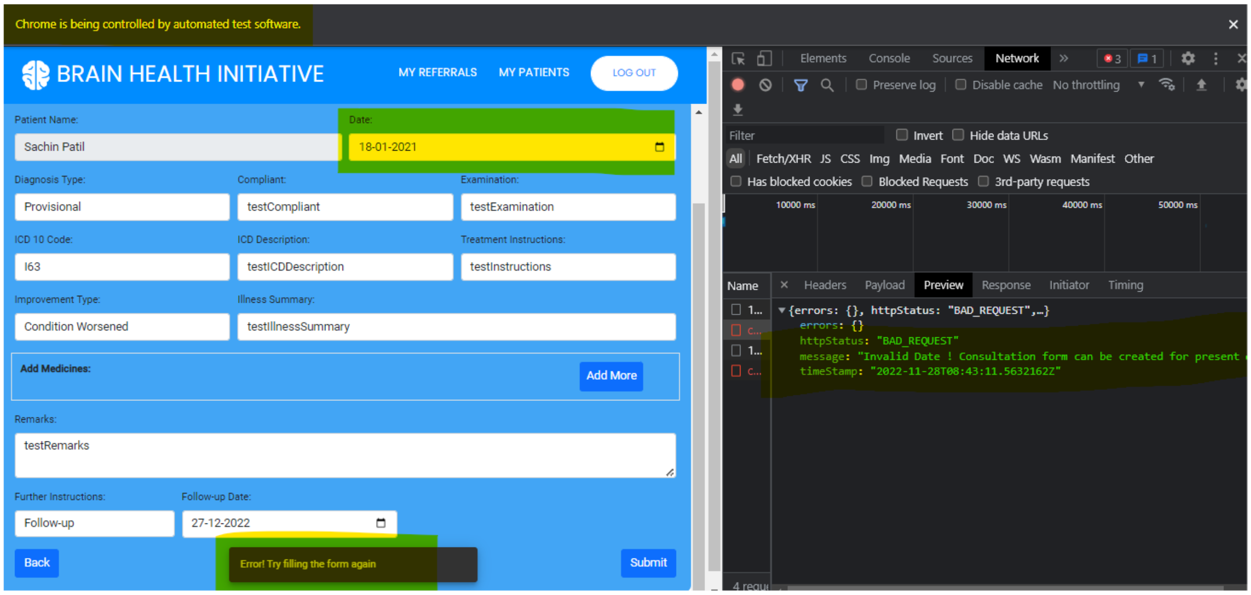
1. Constraint: Form Date Field is auto populated with current date and is read only.

Bypassing the Constraint:



```
// Bypassing ReadOnly validation
WebElement date = webDriver.findElement(By.name("date"));
//Making Date field Editable
js.executeScript( s: "document.getElementsByName('date')[0].readOnly=false");
//Entering Past Date
date.sendKeys( ...charSequences: "18-01-2021");
```

Here By using JavaScript and selenium we change the attribute property of date field to editable And inserted previous date (18-01-2021) which is invalid and submitted the form



Patient Name: Sachin Patil
Date: 18-01-2021
Diagnosis Type: Provisional
Compliant: testCompliant
Examination: testExamination
ICD 10 Code: I63
ICD Description: testICDDescription
Treatment Instructions: testInstructions
Improvement Type: Condition Worsened
Illness Summary: testIllnessSummary
Add Medicines:
Remarks: testRemarks
Further Instructions: Follow-up
Follow-up Date: 27-12-2022

Error! Try filling the form again

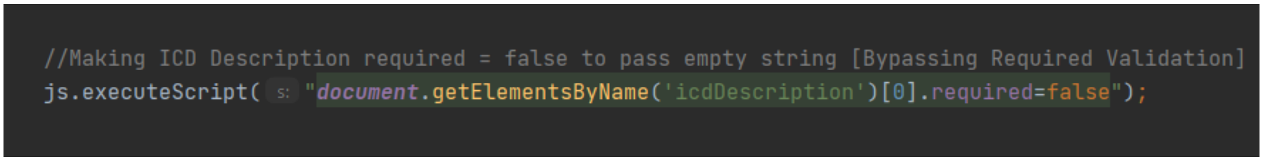
```
▼{errors: {}, httpStatus: "BAD_REQUEST",…}
   errors: {}
   httpStatus: "BAD_REQUEST"
   message: "Invalid Date ! Consultation form can be created for present
   timeStamp: "2022-11-28T08:43:11.5632162Z"
```

Test Result: Passed

Above Screenshot shows server responded with error code and Message : *Invalid Date! Consultation Form can only be created for present date.*

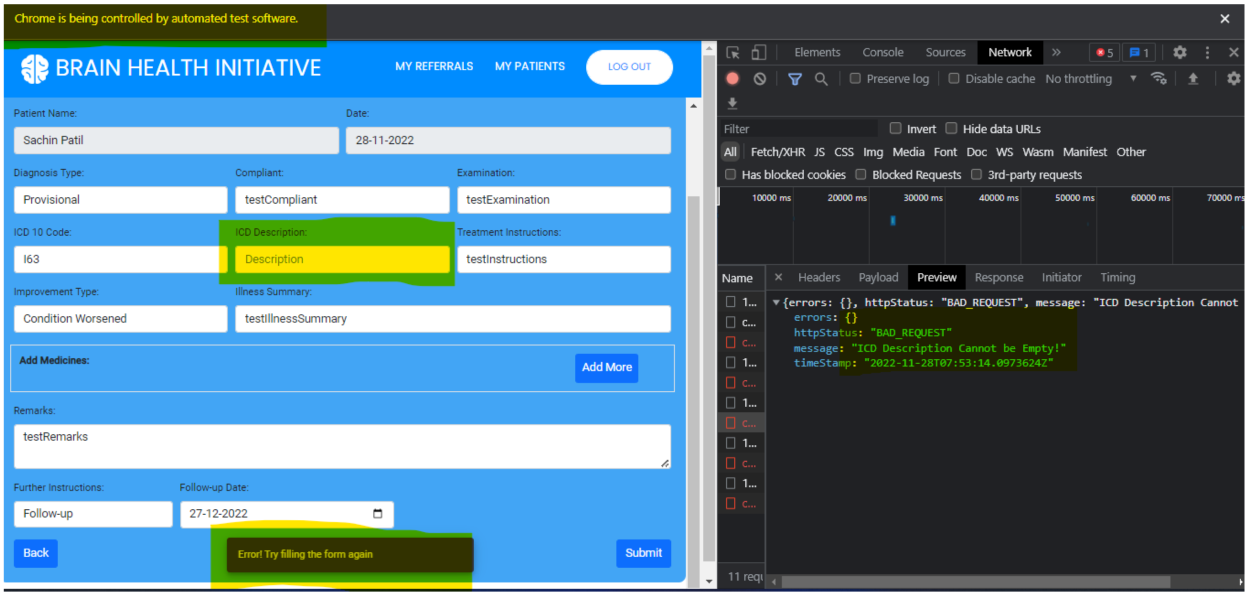
2. Constraint: ICD Description is required

Bypassing the Constraint:



```
//Making ICD Description required = false to pass empty string [Bypassing Required Validation]
js.executeScript( s: "document.getElementsByName('icdDescription')[0].required=false");
```

Here By using JavaScript and selenium we change the attribute property of date field to non-required, kept it blank and submitted the form



Chrome is being controlled by automated test software.

Patient Name: Sachin Patil
Date: 28-11-2022
Diagnosis Type: Provisional
Compliant: testCompliant
Examination: testExamination
ICD 10 Code: I63
ICD Description: Description
Treatment Instructions: testInstructions
Improvement Type: Condition Worsened
Illness Summary: testIllnessSummary
Add Medicines:
Remarks: testRemarks
Further Instructions: Follow-up
Follow-up Date: 27-12-2022

Error! Try filling the form again

```
▼{errors: {}, httpStatus: "BAD_REQUEST", message: "ICD Description Cannot
   errors: {}
   httpStatus: "BAD_REQUEST"
   message: "ICD Description Cannot be Empty!"
   timeStamp: "2022-11-28T07:53:14.0973624Z"
```

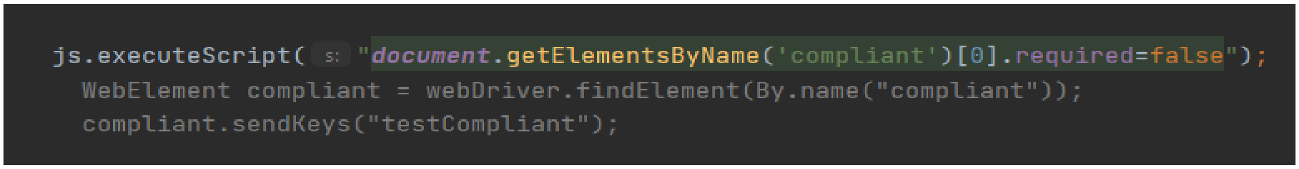
Test Result: Passed

Above Screenshot shows server responded with error code and Message: *ICD Description cannot be empty.*

Valid Input

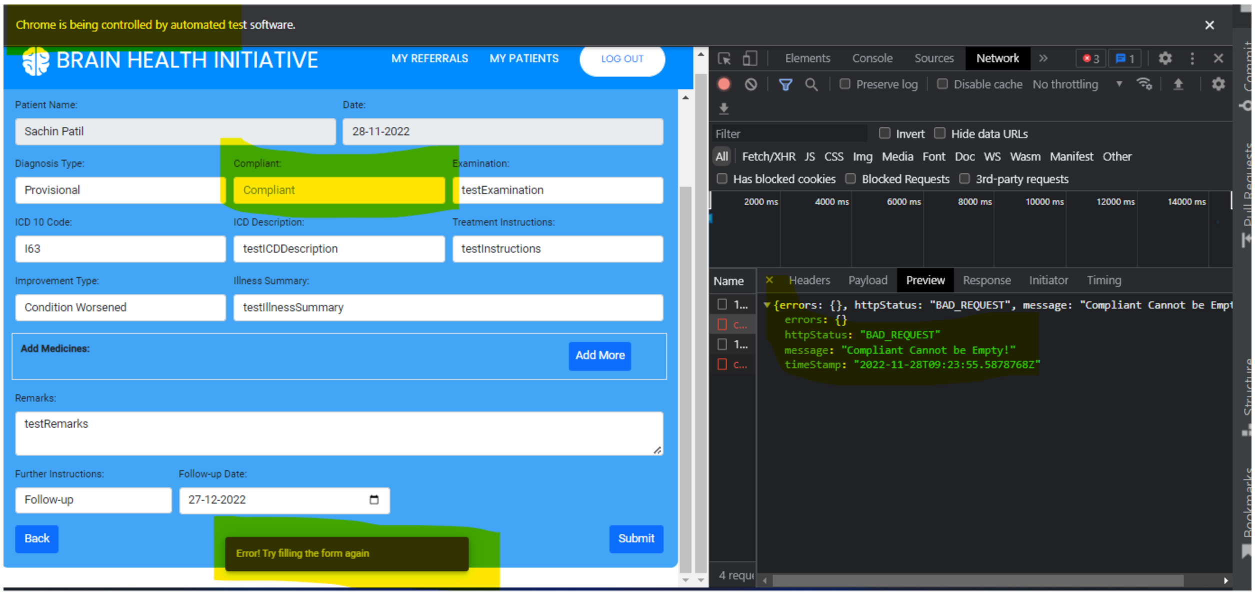
3. Constraint: Compliant is required

Bypassing the Constraint:



```
js.executeScript( s: "document.getElementsByName('compliant')[0].required=false");
  WebElement compliant = webDriver.findElement(By.name("compliant"));
  compliant.sendKeys("testCompliant");
```

Here By using JavaScript and selenium we change the attribute property of date field to non-required, kept it blank and submitted the form



Chrome is being controlled by automated test software.

Patient Name: Sachin Patil

Date: 28-11-2022

Diagnosis Type: Provisional

Compliant: Compliant

Examination: testExamination

ICD 10 Code: I63

ICD Description: testICDDescription

Treatment Instructions: testInstructions

Improvement Type: Condition Worsened

Illness Summary: testIllnessSummary

Add Medicines:

Remarks: testRemarks

Further Instructions: Follow-up

Follow-up Date: 27-12-2022

Error! Try filling the form again

```
▼{errors: {}, httpStatus: "BAD_REQUEST", message: "Compliant Cannot be Empt
   errors: {}
   httpStatus: "BAD_REQUEST"
   message: "Compliant Cannot be Empty!"
   timeStamp: "2022-11-28T09:23:55.5878768Z"
```

Test Result: Passed

Above Screenshot shows server responded with error code and Message: *Compliant cannot be empty.*

Attribute	**Constraint**	**Test Case**	**Test Result**
Date	Current Date Only [Read Only Field]	18-1-2021	Passed
Compliant	Required Field	Empty	Passed
ICD Description	Required Field	Empty	Passed

Project Contributions:

Jayprakash Ray [MT2021060]	End to End Bypass Testing of Consultation Form Module with Documentation.
Khushal Abrol [MT2021063]	End to End Bypass Testing of Admin Module with Documentation.

References:

[1] J. Offutt, Y. Wu, X. Du and H. Huang, "Bypass testing of Web applications," 15th International Symposium on Software Reliability Engineering, 2004, pp. 187-197, doi: 10.1109/ISSRE.2004.13.