

Documentation of Test Results

1. Summary of encountered bugs

a. Bugs from Test_Case_id001

Step where bug was Encountered	Test Steps	Errors/bugs	Root cause	How it was resolved
1	Register Patient	Userentry app not found (django error)	Missing app in <code>INSTALLED_APPS</code> in <code>Settings.py</code>	Fixed adding app "Userentry" to <code>setting.py</code>
2	Register TP	Django templates error; not redirecting after registering	Views not added in <code>Views.py</code> Incorrect Url in <code>Urls.py</code>	Fixed in <code>Views.py/ Urls.py</code>
3	Register Patient	User unable to login/ no redirection	Redirect function not added in <code>registerPatient.html</code>	Fixed html form part (test after test 1)
4	Register with a previously used email address and username	Registered successfully/ redirect to login page causing errors with searchtp	Email Field not unique in <code>Models.py</code>	Fixed <code>models.py</code> for username and email
5	Register Patient after newly added model types	Certain model names not found/ duplicate models found.	Duplicate names in models call of <code>Forms.py</code> and Model names not added in models call in <code>Forms.py</code>	Fixed the imported models in <code>forms.py</code>
6	Register to after newly added CSS/template	CSS not working	Static loading tag not added in HTML	fixed by loading static file into HTML
7	Register User	Register forms not working	Views and urls don't match	Issue fixed with <code>views.py</code> and <code>urls.py</code>
8	Register User (any)	User not registering as certain user (TP/Patient)	One form trying to register both users causing clash	Fixed by adding separate forms for Patient/TP
10	Register TP	Form not saving into the models/table	Save function was not used to save user and forms. <code>user.save()/ forms.save()</code> not added	fixed html file

b. Bugs from Test_Case_id002

Step where bug was Encountered	Test Steps	Errors/bugs	Root cause	How it was resolved
5	<ol style="list-style-type: none"> 1. Pass an invalid origin address 2. Select one of the options of specialty from the dropdown menu 3. toggle the insurance match field 4. press search 	IndexError: list index out of range	Since the origin was invalid (for example origin = 'asdfghjkl'), the google maps api wasn't able to identify it as a location. Hence there was no route, and since the distance is retrieved by accessing an element of the routes list, that's what gave an IndexError.	Encapsulated the cause of the issue in a try except statement. If there is an invalid origin passed, an exception will be called, then the code in the except block will set the distance variable to '-1' indicating that the address was not valid.
6	<ol style="list-style-type: none"> 1. Pass an invalid origin address 2. Don't explicitly select any option from the dropdown menu 3. toggle the insurance match field 4. press search 5. 	IndexError: list index out of range	Since the origin was invalid (for example origin = 'asdfghjkl'), the google maps api wasn't able to identify it as a location. Hence there was no route, and since the distance is retrieved by accessing an element of the routes list, that's what gave an IndexError.	Encapsulated the cause of the issue in a try except statement. If there is an invalid origin passed, an exception will be called, then the code in the except block will set the distance variable to '-1' indicating that the address was not valid.
7	<ol style="list-style-type: none"> 1. Pass an invalid origin address 2. Select any option from the dropdown menu 3. Don't toggle the insurance 	IndexError: list index out of range	Since the origin was invalid (for example origin = 'asdfghjkl'), the google maps api wasn't able to identify it as a location. Hence there was no route, and since the distance is retrieved by	Encapsulated the cause of the issue in a try except statement. If there is an invalid origin passed, an exception will be called, then the code in the except block will set the distance variable to '-1' indicating that

	match field 4. Press search		accessing an element of the routes list, that's what gave an IndexError.	the address was not valid.
8	1. Pass an invalid origin address 2. Don't explicitly select any option from the dropdown menu 3. Don't toggle the insurance match field 4. press search	IndexError: list index out of range	Since the origin was invalid (for example origin = 'asdfghjkl'), the google maps api wasn't able to identify it as a location. Hence there was no route, and since the distance is retrieved by accessing an element of the routes list, that's what gave an IndexError.	Encapsulated the cause of the issue in a try except statement. If there is an invalid origin passed, an exception will be called, then the code in the except block will set the distance variable to '-1' indicating that the address was not valid.

c. Bugs from Test_Case_id003

There were no encountered bugs while testing Book appointments, and this is mainly due to the fact that there is not much that can go wrong as the user input itself is simply a series of button clicks, unlike 'Find Nearest treatment providers' (tested in TestCaseid002), where there was a text field open to many types of input.