Story titile: User Registration

Story Description: As a User I should able to register myself by providing below inputs.

1. Name (text only).
2. email id (email format)
3. moblie number (only nubmers).
4. DOB (date - DD/MON/YYYY)
5. Gender (drop down or radio button).
6. Photo (file selection)
7. Password (Alpha numberic with 6-15 length)
8. Confirm password (Alpha numberic).

Message Scenarios:

|  |  |  |  |
| --- | --- | --- | --- |
| Conditions | Level | Message | Message code |
| Name is empty | Error | Name can’t be empty | MM0001 |
| Invalid DOB format | Error | DOB should be in DD/Mon/YYYY format |  |
|  |  |  |  |
| After saved | Info | The user details saved successfully |  |

Error response:

For validation failure: HTTP status code 501

Error response: {

“isError”:true,

“messages”: [{“messageCode”:””,

“message”:””,

“level”:”error”},

{“messageCode”:””,

“message”:””},

{“messageCode”:””,

“message”:””}

]

}

Please show the below error message when for the respective negative scenarions

* Name:

If it's empty then show ***Name can't be empty***.

If non-aplhabet characters ***Only characters are allowed***.

* Email id:

If email id is empty then show

if the emila is not as per format(email**@**email**.**com) - ***Please enter valid email id***

* DOB:

if it's empty show ***DOB can't be empty.***

* Password:

if it's empty then show ***Password can't be empty.***

if the password is not as per required length then show password ***should be in between 6-15 charecter.***

* Confirm Password:

if it's empty then show ***Confirm Password can't be empty.***

if the confirm password field value is not matching with password field value then show ***Password and confirm password is not same.***

Exit Scenario:

After registration, I should able to login and access the application.

Task List:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Assigned with | Story Point | Aprox. Start Date | End date |
| Crate DB table |  | 1 |  |  |
| Create REST resources |  | 5 | 11-Apr-2019 | 16-Apr-2019 |
| Create the UI screens |  |  |  |  |
|  |  |  |  |  |

Task details:

1. Create below REST API resources –
   1. API to save the user details - /user Http Method Post.

Payload: {

“name”:””,

}

* 1. Get the user details - /user/{userId} Http Method – GET.
  2. Update the user details - /user HttpMethod - PUT

1. Create the client UI screen -