1.1 User Name Validation

Test case Id	**Test Case**	**Test Function**	**Result pass/Fail**

TC01	ValidateUsername(snehal);
	
	O/P :
		{
		
		}
			Fail

TC02	vaildateUsername(snehal);
	
	O/P: username is accepted.
		validateUsername(name)
		{
		If(name contains alphabets)
		{
		Print “Accepted”;
		}
		
		}
			Pass

TC03	vaildateUsername(snehal B);
	
	O/P:
	Error :username should not contain space
		validateUsername(name)
		{
		If(name contains alphabets)
		{
		Print “Accepted”;
		}
		
		}
			Fail

TC04	ValidateUsername(snehal B);
	
	O/P :
	Error :username should not contain space.
		validateUsername(name)
		{
		if(name contains space)
		{
		Error: username should not contain space.
		}
		elseIf(name contains alphabets)
		{
		Print “Accepted”;
		}
		
		}
			Pass

TC05	ValidateUsername(snehal123);	validateUsername(name)
		{
		if(name contains space)
		{
		Error: “username should not
			Fail

	O/P: username is accepted	contain space.”
		}
		
		elseIf(name contains alphabets)
		{
		Print “Accepted”;
		}
		
		}	

TC06	ValidateUsername(snehal123);
	
	O/P :username is accepted
		validateUsername(name)
		{
		if(name contains space)
		{
		Error:” username should not contain space”.
		}
		elseif(name contains alphanumeric)
		{
		Print “Accepted”;
		}
		elseIf(name contains alphabets)
		{
		Print “Accepted”;
		}
		
		}
			Pass

TC07	ValidateUsername(snehal*123);
	
	O/P :
	Error :username should not contain special symbols.
		validateUsername(name)
		{
		if(name contains space)
		{
		Error:” username should not contain space”.
		}
		elseif(name contains alphanumeric)
		{
		Print “Accepted”;
		}
		elseIf(name contains alphabets)
		{
		Print “Accepted”;
		}
		
		}
			Fail

TC08	ValidateUsername(snehal*123);
	
	O/P :
	Error :username should not contain special symbols.
		validateUsername(name)
		{
		if(name contains space)
		{
		Error:" username should not contain space".
		}
		
		elseif(name contains special char){
		
		Error: "username should not contain special symbols. "
		}
		elseif(name contains alphanumeric)
		{
		Print "Accepted";
		}
		elseIf(name contains alphabets)
		{
		Print "Accepted";
		}
		
		}
			Pass

TC08	ValidateUsername( );
	
	O/P :
	Error :username should not empty
		validateUsername(name)
		{
		if(name contains space)
		{
		Error:" username should not contain space".
		}
		
		elseif(name contains special char){
		
		Error: "username should not contain special symbols. "
		}
		elseif(name contains alphanumeric)
		{
		Print "Accepted";
		}
		elseIf(name contains alphabets)
		{
		Print "Accepted";
		}
		
		}
			Fail

TC09	ValidateUsername( );
	
	O/P :
	Error :username should not empty
		validateUsername(name)
		{
		if(name is empty){
		Error:" username should not empty".
		}
		else{
		}
		if(name contains space)
		{
		Error:" username should not contain space".
		}
		
		elseif(name contains special char){
		
		Error: "username should not contain special symbols. "
		}
		elseif(name contains alphanumeric)
		{
		Print "Accepted";
		}
		elseIf(name contains alphabets)
		{
		Print "Accepted";
		}
		}}
			Pass

1.2 email validation

Test case Id	**Test Case**	**Test Function**	**Result pass/Fail**

TC01	ValidateEmail(snehal);	{
		
		}
			Fail

TC02	validateEmail(snehal);	validateEmail(email){
		
		If (email !contain "@"){
		Error :"email should contain @"
		}
			Pass

	O/P : Error : email should contain @	}	

TC03	validateEmail(snehal@);
	
	O/P:
	Error:Email should contain domain name
		validateEmail(email){
		
		If (email !contain “@”){
		Error :”email should contain @”
		}
		
		}
			Fail

TC04	validateEmail(snehal@gmail);
	
	O/P:
	Error:email should contain domain name
		validateEmail(email){
		If (email !contain “@”){
		Error :”email should contain @”
		}
		if(email !contain domain name){
		Error:”email should contain domain name”
		}
		
		}
			Fail

TC05	validateEmail(snehal@gmail);
	
	O/P:
	Error:email should contain domain name
		validateEmail(email){
		If (email !contain “@”){
		Error :”email should contain @”
		}
		if(email !contain domain name){
		Error:”email should contain domain name”
		}
		if(email !contain .com){
		Error::”email should contain .com”
		}
		}
			Pass

TC06	validateEmail(@gmail.com);	validateEmail(email){
		If (email !contain “@”){
		Error :”email should contain @”
		}
		if(email !contain domain name){
		Error:”email should contain domain name”
		}
		if(email !contain .com){
			Fail

	O/P: Error: email should contain username	Error::"email should contain .com" } }	

TC07	validateEmail(@gmail.com);
	
	O/P:
	Error: email should contain username
		validateEmail(email){
		If (email !contain "@"){
		Error :"email should contain @"
		}
		if(email !contain domain name){
		Error:"email should contain domain name"
		}
		if(email !contain .com){
		Error::"email should contain .com"
		}
		if(email !contain username){
		Error: "email should contain username"
		}
		}
			Pass

TC08	validateEmail(snehal@gmail.com);
	
	O/P: email Accepted
		validateEmail(email){
		If (email !contain "@"){
		Error :"email should contain @"
		}
		if(email !contain domain name){
		Error:"email should contain domain name"
		}
		if(email !contain .com){
		Error::"email should contain .com"
		}
		if(email !contain username){
		Error: "email should contain username"
		}
		}
			Fail

TC09	validateEmail(snehal@gmail.com)	validateEmail(email){
		If (email !contain "@"){
		Error :"email should contain @"	

	O/P: email Accepted	}
		if(email !contain domain name){
		Error:”email should contain domain name”
		}
		if(email !contain .com){
		Error::”email should contain .com”
		}
		if(email !contain username){
		Error: “email should contain username”
		}
		
		if(email contains username,”@”,domain Name,”.com”)
		{
		Print “ email Accepted”
		}
		}
			Pass

1.3 Password validation

Test case Id	**Test Case**	**Test Function**	**Result pass/Fail**
TC01	validatePassword(abc); O/P: Error :password too short	{ }	Fail

TC02	validatePassword(abc);	validatePassword(password){
		
		if(password < 6){
		Error :password too short
		}
		}
			Pass

TC03	validatePassword(abc);
	
	O/P
	Error:password too short
	Password should contain at least
		validatePassword(password){
		
		if(password < 6){
		Error :password too short
		}
		}
			Fail

	1 number		

TC04	validatePassword(abc);
	
	O/P
	Error:password too short
	Password should contain at least 1 number
		validatePassword(password){
		
		if(password < 6){
		Error :password too short
		}
		if(password !contain number){
		Error:Password should contain at least 1 number
		}
		}
			Pass

TC05	validatePassword(abc);
	
	O/P
	Error:
	1.password too short
	2.Password should contain at least 1 number
	3.Password should contain at least 1 special char
		validatePassword(password){
		
		if(password < 6){
		Error :password too short
		}
		if(password !contain number){
		Error:Password should contain at least 1 number
		}
		}
			Fail

TC06	validatePassword(abcde1);
	
	O/P
	Error:
	Password should contain at least 1 special char
		validatePassword(password){
		
		if(password < 6){
		Error :password too short
		}
		if(password !contain number){
		Error:Password should contain at least 1 number
		}
		
		if(password ! contains special char){
		Error: Password should contain at least 1 special char
		}
		}
			Pass

TC07	validatePassword(abcde1);	validatePassword(password){
		
		if(password < 6){
		Error :password too short
		}
		if(password !contain number){
			Fail

	O/P:
	Error:
	Password should contain at least 1 Upper case letter
		Error:Password should contain at least 1 number
		}
		
		if(password ! contains special char){
		Error: Password should contain at least 1 special char
		}
		}	

TC08	validatePassword(abcde@1);
	
	O/P:
	Error:
	Password should contain at least 1 Upper case letter
		validatePassword(psswd){
		if(password < 6){
		Error :password too short
		}
		if(password !contain number){
		Error:Password should contain at least 1 number
		}
		if(password ! contains special char){
		Error: Password should contain at least 1 special char
		}
		if(password !contain upper case letter){
		Error:Password should contain at least 1 Upper case letter
		}
		}
			Pass

TC09	validatePassword( );
	
	O/P
		validatePassword(password){
		if(password < 6){
		Error :password too short
		}
		if(password !contain number){
		Error:Password should contain at least 1 number
		}
		if(password ! contains special char){
		Error: Password should contain at least 1 special char
		}
		if(password !contain uppercase letter){
		Error:Password should contain at
			Fail

	Error: Password should not be empty.	least 1 Upper case letter } }	

TC10	validatePassword( );
	
	O/P
	Error:
	Password should not be empty.
		validatePassword(password){
		if(password is empty){
		Error:
		Password should not be empty
		}
		else{
		if(password < 6){
		Error :password too short
		}
		if(password !contain number){
		Error:Password should contain at least 1 number
		}
		if(password ! contains special char){
		Error: Password should contain at least 1 special char
		}
		if(password !contain uppercase letter){
		Error:Password should contain at least 1 Upper case letter
		}
		}
		}
			Pass

TC11	validatePassword(Abc@123);
	
	O/P:
	Password Accepted
		validatePassword(password){
		if(password is empty){
		Error:
		Password should not be empty
		}
		else{
		if(pswd < 6){
		Error :password too short
		}
		if(pswd !contain number){
		Error:Password should contain at least 1 number
		}
		if(pswd ! contains special char){
		Error: Password should contain at least 1 special char
		}
			Pass

		if(pswd !contain uppercase letter){
		Error:Password should contain at least 1 Upper case letter
		}
		}	

1.4 Re-enter Password validation

Test case Id	**Test Case**	**Test Function**	**Result pass/Fail**

TC01	Re_passwordValidation(abc);	{
		
		}
		
			Fail

TC02	Re_passwordValidation(abc);
	
	O/P:
	Error:
	Password Not Match
		Re_passwordValidation (Rpswd){
		if(Rpswd == pswd){
		Print “password match”
		}
		Else{
		Password Not Match
		}
		}
			Pass

TC03	Re_passwordValidation(abc);
	
	O/P:
	Password Match
		Re_passwordValidation (Rpswd){
		if(Rpswd == pswd){
		Print “password match”
		}
		Else{
		Password Not Match
		}
		}
			Pass