### **User Story 1**

**Requirement:**

Implement Log In functionality.

| **Epic** | **User Story** | **Acceptance Criteria** | |
| --- | --- | --- | --- |
| As an **Employee, I Want** to be able to securely login to the system **so that the I am only authorized** to access my account to book the seat. |  | |
| As a **registered Employee**, **I want** to log in with my username and password **so that** the system can authenticate me, and I can trust it. | 1. Given that I am a registered employee and logged out, if I go to the log in page and enter my username and password and click on Log in, then the profile associated to my account should be accessible. 2. Given that I am a registered employee and logged out, if I go to the log in page and enter my username but an incorrect password and click on Log in, then log in fails with an error message that specifies that the username or password was wrong. | |
|  | As a **registered Employee**, **I want** to login using credentials and if I forget my password, Ishould beable to go to forget password and change. | 1. Given that I am a registered employee and logged out, if I go to forget password, I can change my password by providing correct OTP sent to their registered email. | |

### **Functional Validations for Login**

1. Verify that cursor is focused on the “Username” text box on the page load (login page).
2. Verify that tab functionality is working properly or not.
3. Verify that Enter/Tab key works as a substitute for the Sign-in button.
4. Verify that the User is able to Login with Valid Credentials.
5. Verify that the User is not able to Login with an invalid Username and invalid Password.
6. Verify that the User is not able to Login with a Valid Username and invalid Password.
7. Verify that the User is not able to log in with an invalid Username and Valid Password.
8. Verify that the User is not able to log in with a blank Username or Password.
9. Verify that the User is not able to Login with inactive credentials.
10. Verify that the reset button clears the data from all the text boxes in the login form.
11. Verify that the login credentials, mainly password stores in a database in an encrypted format.
12. Verify that clicking on the browser back button after successful login should not take the User to log out mode.
13. Verify that validation message is displayed in the case when User leaves Username or Password as blank.
14. Verify that validation message is displayed in case of exceeding the character limit of the Username and Password fields.
15. Verify that validation message is displayed in case of entering special character in the Username and password fields.
16. Verify that the “Keep me logged in” checkbox is unselected by default (depends on business logic, it may be selected or unselected).
17. Verify that the timeout of the login session (Session Timeout).
18. Verify that the logout link is redirected to login/home page.
19. Verify that User is redirected to appropriate page after successful login.
20. Verify that the User is redirected to the Forgot password page when clicking on the Forgot Password link.
21. Verify that the User is redirected to the Create an account page when clicking on the Signup / Create an account link.
22. Verify that spaces should not be allowed before any password characters attempted.

### **Security Test Cases for Login Page**

1. Verify that clicking on the browser back button after successful logout should not take the User to a logged-in mode.
2. Verify that there is a limit on the total number of unsuccessful login attempts (No. of invalid attempts should be based on business logic. Based on the business logic, User will be asked to enter the captcha and try again or user will be blocked).
3. Verify that the password is in encrypted form (masked format) when entered in the password field.
4. Verify the password can be copy-pasted. System should not allow users to copy paste password.
5. Verify that encrypted characters in the “Password” field should not allow deciphering if copied.
6. Verify whether the login form is revealing any security information by viewing the page source.
7. Verify that the login page is vulnerable to SQL injection.
8. Verify whether Cross-site scripting (XSS ) vulnerability works on a login page. XSS vulnerability may be used by hackers to bypass access controls.

### **Performance Test Cases for Login Page**

1. Verify that how much time the application is taking to load the home page after entering the valid username and password in the login page.

### **User Story 2**

**Requirement:**

Implement Home Page.

| **Epic** | **User Story** | **Acceptance Criteria** |
| --- | --- | --- |
| As a registered **Employee, I Want** to be able to view my home page after login. | As a registered **Employee**, **I want** to see my booking information if already booked. | Given that I am a registered employee, I can see my name and the seat number that I have booked, or it should show me no seats booked. |
| As a registered **Employee, I want** to be able to book a seat from home page itself. | Given that I am a registered employee, I should be able to navigate to book a seat with my details already stored. |
| As a registered **Employee**, **I want** to see complete details after booking my seat. | Given that I am a registered employee, I can see my booking details such as shift time, floor, booking ID, seat number and the date for which I have booked. |
|  | As a registered **Employee**, **I want** to be able to see my profile from home page. | Given that I am a registered employee, I should be able to navigate to my profile with my details. |

### **Functional Validations for Home Page**

1. Verify that user details are displayed.
2. Verify that view profile and book a seat is visible.
3. Verify that on click of book seat button redirects to the booking page.
4. Verify that on click of profile button redirects to the profile page.
5. Verify that the booked seat number is showing.
6. Verify that on click of view pass the complete details of booked seat is displayed.

### **Security Test Cases for Home Page**

1. Verify that the user is not in login mode after he is logged out.

### **Performance Test Cases for Home Page**

1. Verify that all the buttons on home page are working on a single click.

### **User Story 3**

**Requirement:**

Implement view Profile functionality.

| **Epic** | **User Story** | **Acceptance Criteria** |
| --- | --- | --- |
| As a registered **Employee, I Want** to be able to view my profile. | As a registered **Employee**, **I want** to see my details. | Given that I am a registered employee, I can see my employee Id and email ID. |
| As a **registered Employee**, **I want** to change my password. | Given that I am a registered employee, I can change my password by providing old password and I can use new password to login to the system. |

### **Functional Validations for View Profile**

1. Verify that on click of change password the user is re-directed to change password page.
2. Verify that user can change password only by providing old password.
3. Verify that user can login using changed password.
4. Verify that user can view his details.
5. Verify that user gets a notification on successful change of password.

### **Security Test Cases for View Profile**

1. Verify that new password meets the requirements of strong password.

### **Performance Test Cases for View Profile**

1. Verify that how much time the application is taking to load the complete details of the user.
2. Verify that how much time the application takes to send the notification of the successful update password.

### **User Story 4**

**Requirement:**

Implement seat booking functionality.

| **Epic** | **User Story** | **Acceptance Criteria** |
| --- | --- | --- |
| As a registered **Employee, I Want** to be able to book a seat. | As a registered **Employee**, **I want** to click on book a seat to navigate to booking page. | Given that I am a registered employee, I can book a seat by navigate to booking page. |
| As a **registered Employee**, **I want** to book a seat on a particular day or a week in booking page. | Given that I am a registered employee, I should be able to select the date both for single day or on weekly basis for any future date or for the present day. |
|  | As a **registered Employee**, **I want** to book a seat on a particular shift in booking page. | Given that I am a registered employee, I should be able to view all the shifts available in the office and select. |
|  | As a **registered Employee**, **I want** to book a seat on any floor of the office in booking page. | Given that I am a registered employee, I should be able to view all the available seats and also booked seats in that floor I have chosen. |
|  | As a **registered Employee**, **I want** to opt for food in booking page. | Given that I am a registered employee, I want the option for both opting for office lunch or not. |

### **Functional Validations for Seat Booking**

1. Verify that user can click on book seat and navigate to booking page.
2. Verify that user can book a seat on any floor.
3. Verify that user can book only available seats.
4. Verify that user cannot book already booked seat.
5. Verify that user can book a seat daily or weekly basis.
6. Verify that user can select present or future dates.
7. Verify that user cannot select past dates for booking.
8. Verify when user is booking a seat all the available shifts are displayed.
9. Verify the user can select only one shift for booking a seat.
10. Verify that user cannot select a shift after a certain time.
11. Verify that user can opt for lunch.
12. Verify that user can deny lunch.

### **Security Test Cases for Seat Booking**

1. Verify that booked seat is not displayed as an available seat to the user.

### **Performance Test Cases for Seat Booking**

1. Verify that how much time the application is taking to load the floor blueprint to book a seat.
2. Verify that how much time the application takes to display booked seat information.

### **User Story 5**

**Requirement:**

Implement admin functionality.

| **Epic** | **User Story** | **Acceptance Criteria** |
| --- | --- | --- |
| As an **Admin**, I should be able to check and maintain the records of the booking details. | As an **Admin**, **I want** tologin as an admin. | Given that I am an admin, I should be able to successfully login as an admin. |
| As an **Admin**, **I want** to send to the credentials to the user through email. |  |
| As an A**dmin**, **I want** to see the number of seats booked. | Given that I am an admin, I should be able to successfully send the credentials through mail. |
| As an **Admin**, **I want**  to see the number of people who have opted for food. | Given that I am an admin, I should be able to see the seat that have been booked with employee ID and for which floor and the shift time. |
|  | As an **Admin**, **I want** to generate weekly and monthly reports. | Given that I am an admin, I should be able to see the employee who has opted for food. |
|  | As an **Admin**, **I want** to confirm the employee coming to the office. | Given that I am an admin, I should be able to see the weekly and monthly reports. |
|  |  | Given that I am an admin, I should be able to verify the booking and then approve. |

### **Functional Validations for Admin Page**

1. Verify that cursor is focused on the “Username” text box on the page load (login page).
2. Verify that tab functionality is working properly or not.
3. Verify that the Admin is able to Login with Valid Credentials.
4. Verify that the Admin is not able to Login with an invalid Username and invalid Password.
5. Verify that the Admin is not able to Login with a Valid Username and invalid Password.
6. Verify that the Admin is not able to log in with an invalid Username and Valid Password.
7. Verify that the Admin is not able to log in with a blank Username or Password.
8. Verify that the Admin is not able to Login with inactive credentials.
9. Verify that the reset button clears the data from all the text boxes in the login form.
10. Verify that the login credentials, mainly password stores in a database in an encrypted format.
11. Verify that clicking on the browser back button after successful login should not take the Admin to log out mode.
12. Verify that validation message is displayed in the case when Admin leaves Username or Password as blank.
13. Verify that validation message is displayed in case of exceeding the character limit of the Username and Password fields.
14. Verify that validation message is displayed in case of entering special character in the Username and password fields.
15. Verify that the “Keep me logged in” checkbox is unselected by default (depends on business logic, it may be selected or unselected).
16. Verify that the timeout of the login session (Session Timeout).
17. Verify that the logout link is redirected to login/home page.
18. Verify that User is redirected to appropriate page after successful login.
19. Verify that the User is redirected to the Forgot password page when clicking on the Forgot Password link.
20. Verify that the User is redirected to the Create an account page when clicking on the Signup / Create an account link.
21. Verify that spaces should not be allowed before any password characters attempted.
22. Verify that Admin can generate reports on daily or weekly basis.
23. Verify that Admin can approve incoming request of an employee.
24. Verify that the Admin can send the credentials to that particular user through mail.

### **Security Test Cases for Admin Page**

1. Verify that clicking on the browser back button after successful logout should not take the User to a logged-in mode.
2. Verify that there is a limit on the total number of unsuccessful login attempts (No. of invalid attempts should be based on business logic. Based on the business logic, User will be asked to enter the captcha and try again, or user will be blocked).
3. Verify that the password is in encrypted form (masked format) when entered in the password field.
4. Verify the password can be copy-pasted. System should not allow users to copy paste password.
5. Verify that encrypted characters in the “Password” field should not allow deciphering if copied.
6. Verify whether the login form is revealing any security information by viewing the page source.
7. Verify that the login page is vulnerable to SQL injection.
8. Verify whether Cross-site scripting (XSS ) vulnerability works on a login page. XSS vulnerability may be used by hackers to bypass access controls.

### **Performance Test Cases for Admin Page**

1. Verify that how much time the application is taking to load the home page after entering the valid username and password in the login page.