

Testing Overview: After the user runs the Burnout as an application on Visual Studio, the user can browse to <http://localhost:3000>.

Test Data Summary: There should be some data that stores information about meals, workout schedules.

Jing Huang(jhuang42) Create test cases TC0-TC20.

Test ID	Description	Expected Results	Actual Results
TC0: Navigate Workout Page 01	Precondition: The Burnout application is running, and the user is logged in. <ol style="list-style-type: none">1. Open the application in a web browser.2. Navigate to the “MY WORKOUTS” section from the navigation bar.	The “MY WORKOUTS” page is displayed, showing the available table of weekly workout plans.	The “MY WORKOUTS” page is displayed, showing available workouts schedule.
TC1: View the ADD button view	Precondition: TC0 Input Data: N/A Action: Click the “ADD” button.	Pop-up window shows: “New Workout Form.” Pop-up window	Pop-up window shows: “New Workout Form.”
TC2: Close the “New Workout Form” view without adding data	Precondition: TC1 Input Data: N/A Action: Click the “CANCEL” button.	Pop-up window is closed	Pop-up window is closed
TC3: Submit the “New Workout Form” view without adding data	Precondition: TC1 Input Data: N/A Action: Click the “Submit” button.	Pop-up window is not closed	Pop-up window is not closed. Error message pop out: <ol style="list-style-type: none">1. “Please select a day.”2. “Workout title is required.”3. “Please enter a valid duration in minutes.”
TC4: View the Edit button view without data	Precondition: TC0 Input Data: N/A Action: Click the “Submit” button.	Nothing pop-up	Nothing pop-up
TC5: Create Schedule Valid 1	Precondition: TC1 Input Data: { day: "Monday",	Pop-up window is closed	Pop-up window is closed

	<p>workoutTitle: "Morning Yoga", duration: 30, videoLink: "https://example.com/morning-yoga", description: "A refreshing yoga session to start the week." },</p> <p>Action: Click the "Submit" button.</p>		
TC6: Create Schedule Valid 2	<p>Precondition: TC1</p> <p>Input Data: { day: "Wednesday", workoutTitle: "HIIT Cardio", duration: 45, videoLink: "https://example.com/hiit-cardio", description: "High-intensity interval training for maximum fat burn." },</p> <p>Action: Click the "Submit" button.</p>	Pop-up window is closed	Pop-up window is closed
TC7: Create Schedule Valid 3	<p>Precondition: TC1</p> <p>Input Data: { day: "Friday", workoutTitle: "Strength Training", duration: 60, videoLink: "https://example.com/strength-training", description: "Build muscle with this comprehensive strength workout." },</p> <p>Action: Click the "Submit" button.</p>	Pop-up window is closed	Pop-up window is closed
TC8: View result of created schedules1	<p>Precondition: TC5, TC6, TC7</p> <p>Input Data: N/A</p> <p>Action1: Click the "MONDAY" tab button. Action2: Click the "WEDNESDAY" tab button. Action3: Click the "FRIDAY" tab</p>	<p>The updated workout plans are displayed by weekday.</p> <p>When the "MONDAY" tab is clicked, a card is displayed showing the updated workout plan: { workoutTitle: "Morning</p>	Same

	button.	<p>Yoga", duration: 30, videoLink: "https://example.com/morning-yoga", description: "A refreshing yoga session to start the week." },</p> <p>When the "WEDNESDAY" tab is clicked, a card is displayed showing the updated workout plan:</p> <pre>{ workoutTitle: "HIIT Cardio", duration: 45, videoLink: "https://example.com/hiit-cardio", description: "High-intensity interval training for maximum fat burn." }</pre> <p>When the "FRIDAY" tab is clicked, a card is displayed showing the updated workout plan:</p> <pre>{ "Strength Training", duration: 60, videoLink: "https://example.com/strength-training", description: "Build muscle with this comprehensive strength workout." },</pre>	
TC9: Navigate the Video Link in Card	<p>Precondition: TC8</p> <p>Input Data: N/A</p> <p>Action:</p> <ol style="list-style-type: none"> Click the "MONDAY" tab to display the workout plan 	<p>When the "MONDAY" tab is clicked, a card is displayed showing the updated workout plan.</p> <p>The user is redirected to the</p>	Same

	<p>card.</p> <ol style="list-style-type: none"> 2. Locate "Morning Yoga" Workout Plan card 3. Click on the "WATCH VIDEO". 	specified video link for the workout, opening in the same window.	
TC10: Create Schedule missing select day field	<p>Precondition: TC1</p> <p>Input Data: { workoutTitle: "Morning Yoga 1", duration: 30, videoLink: "https://example.com/morning-yoga", description: "A refreshing yoga session to start the week." },</p> <p>Action: Click the "Submit" button.</p>	Error message displayed "Please select a day."	same
TC11: Create Schedule missing workoutTitle field	<p>Precondition: TC1</p> <p>Input Data: { day: "Monday", duration: 30, videoLink: "https://example.com/morning-yoga", description: "A refreshing yoga session to start the week." },</p> <p>Action: Click the "Submit" button.</p>	Error message displayed "Workout title is required."	same
TC12: Create Schedule missing duration field	<p>Precondition: TC1</p> <p>Input Data: { day: "Monday", workoutTitle: "Morning Yoga 1", videoLink: "https://example.com/morning-yoga", description: "A refreshing yoga session to start the week." },</p> <p>Action: Click the "Submit" button.</p>	Error message displayed "Please enter a valid duration in minutes."	same
TC13: Create Schedule	Precondition: TC1	Pop-up window is closed	Pop-up window is closed

missing videoLink field	<p>Input Data: { day: "Monday", workoutTitle: "Morning Yoga 1", duration: 30, description: "A refreshing yoga session to start the week." },</p> <p>Action: Click the “Submit” button.</p>		
TC14: Create Schedule missing description field	<p>Precondition: TC1</p> <p>Input Data: { day: "Monday", workoutTitle: "Morning Yoga 2", duration: 30, videoLink: "https://example.com/morning-yoga", },</p> <p>Action: Click the “Submit” button.</p>	Pop-up window is closed	Pop-up window is closed

TC15: View result of created schedules 2	<p>Precondition: TC8, TC10, TC11, TC12, TC13, TC16</p> <p>Input Data: N/A</p> <p>Action1: Click the “MONDAY” tab button.</p>	<p>The updated workout plans are displayed by weekday.</p> <p>When the “MONDAY” tab is clicked, 3 cards is displayed showing the updated workout plan:</p> <pre>1: { workoutTitle: "Morning Yoga", duration: 30, videoLink: "https://example.com/morning-yoga", description: "A refreshing yoga session to start the week." }, 2: { workoutTitle: "Morning Yoga 1", duration: 30, description: "A refreshing yoga session to start the week." }, 3: { workoutTitle: "Morning Yoga 2", duration: 30, videoLink: "https://example.com/morning-yoga", }</pre>	
TC16: Navigate the Video Link in Card	<p>Precondition: TC8</p> <p>Input Data: N/A</p> <p>Action:</p> <ol style="list-style-type: none"> Click the “MONDAY” tab to display the workout plan card. Locate "Morning Yoga 1" Workout Plan card 	<p>No “WATCH VIDEO” displayed.</p>	<p>Same</p>

	6. Click on the “WATCH VIDEO” button.		
TC17: Create Schedule duplicate workoutTitle field 1	<p>Precondition: TC0-TC16</p> <p>Input Data: { day: "Monday", workoutTitle: "Morning Yoga", duration: 20, videoLink: "https://example.com/morning-yoga", description: "A refreshing yoga session to start the week. (duplicated)" },</p> <p>Action: Click the “Submit” button.</p>	Error message “ Workout title must be unique. ” is displayed	Same
TC18: Create Schedule duplicate workoutTitle field 2	<p>Precondition: TC10-TC16</p> <p>Input Data: { day: "Wednesday", workoutTitle: "Morning Yoga", duration: 20, videoLink: "https://example.com/morning-yoga", description: "A refreshing yoga session to start the week. (duplicated)" },</p> <p>Action: Click the “Submit” button.</p>	Pop-up window is closed	Pop-up window is closed
TC19: View result of created schedules 3	<p>Precondition: TC10-TC18</p> <p>Input Data: N/A</p> <p>Action1: Click the “MONDAY” tab button.</p> <p>Action2: Click the “WEDNESDAY” tab button.</p>	<p>The updated workout plans are displayed by weekday.</p> <p>When the “MONDAY” tab is clicked, 3 cards are displayed showing the updated workout plan.</p> <p>When the “WEDNESDAY” tab is clicked, 2 cards are displayed showing the updated workout plan.</p>	Same

TC20: Delete schedule	<p>Precondition: TC19</p> <p>Input Data: N/A</p> <p>Action1: Click the “WEDNESDAY” tab button.</p> <p>Action2: Click the “EDIT” button.</p> <p>Action3: Click the “REMOVE” button from the card with the Title “Morning Yoga”.</p>	<p>The selected card is removed from the screen.</p> <p>When the “WEDNESDAY” tab is clicked, 1 card1 is displayed showing the updated workout plan.</p>	Same
TC:21Navigate to Events Page	<p>Precondition: The Burnout application is running, and the user is logged in. Open the application in a web browser and navigate to the “EVENTS” section from the navigation bar.</p>	The “EVENTS” page is displayed, showing the list of available events.	The “EVENTS” page is displayed, showing the available events.
TC: 22 Fetch Events	<p>Precondition: The Burnout application is running, and the user is on the events page.</p> <p>Input Data: N/A Action</p>	Load the events page.	Events are fetched and displayed correctly.
TC: 23 Filter Events	<p>Precondition: TC22</p> <p>Input Data: "Yoga"</p>	The displayed events are filtered to show only those that match the search query ("Yoga Class").	Filtered events are displayed correctly.
TC: 24 Open Event Details in Modal	<p>Precondition: TC22.</p> <p>Action: Click on the “Yoga Class” event.</p>	A modal opens displaying detailed information about the “Yoga Class,” including description and location.	The event details modal opens as expected.
TC: 25 Close Event Details Modal	<p>Precondition: TC24.</p> <p>Action: Click the “CLOSE” button in the modal.</p>	The event details modal is closed.	The modal closes successfully.
TC: 26 Enroll in Event	<p>Precondition: TC24.</p> <p>Action: Click the “ENROLL” button in the event details modal.</p>	The user is enrolled in the event, and the button changes to “UNENROLL”	Enrollment is successful, button changes to “UNENROLL”
TC: 27 Unenroll from Event	<p>Precondition: TC26. Action: Click the “UNENROLL” button in the event details modal.</p>	The user is unenrolled from the event, and the button changes back to “ENROLL”	Unenrollment is successful, button changes to “ENROLL”
TC: 28 Handle Enrollment Failure	<p>Precondition: TC24.</p> <p>Action: Click the “ENROLL” button, but simulate an enrollment failure.</p>	An error message is displayed stating "Enrollment failed"	Error message is displayed as expected.
TC: 29 Check Google Maps Link	<p>Precondition: TC24.</p> <p>Action: Click on the event location link</p>	Google Maps opens in a new tab showing the location for the event.	Google Maps opens correctly with the location.

TC: 30 Handle No Events Case	Precondition: TC20. Simulate an API response with no events. Action: Refresh the EVENTS page.	A message stating "No events found" is displayed on the page.	The appropriate message is displayed.
TC: 31 Handle Fetch Error	Precondition: TC20. Simulate an API fetch error. Action: Refresh the EVENTS page.	An error message stating "Error fetching events" is displayed on the page.	The appropriate error message is displayed.
TC: 32 Validate Enrollment Status	Precondition: TC21. Action: Click on the "Yoga Class" event after enrollment.	The button should display "UNENROLL" if the user is already enrolled in the event.	Enrollment status is displayed correctly.