A. CONCURRENCY CONTROL AND CONSISTENCY

Suite 1: Read Uncommitted Isolation Level

Test Case	Scenario	Input	Expected Result	Pass/Fail
Simultaneous reads should return consistent results	**Given** three users want to see data corresponding to the same ID **When** the users input the ID and click the search button simultaneously **Then** the application should display the same data for all users	1. User 1 enters “174202” in the search field 2. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 2 enters “174202” in the search field 2. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 3 enters “174202” in the search field 2. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976	Pass

			• Genre: Drama • Rank: 10 • Director: Cua • Actor 1: Gaba • Actor 2: Gonzales	
Simultaneous reading and writing should result in a dirty read	**Given** two users read data that another user is currently updating **When** the users submit the read and write requests simultaneously **Then** the application should return a dirty read for the read request	1. User 1 enters “174202” in the search field 2. User 1 clicks the Search button	The application should display one entry with the following attributes: • ID: 174202 • Title: Edited Entry • Year: 1976 • Genre: Comedy • Rank: 6.6 • Director: Gaba • Actor 1: Gonzales • Actor 2: Lee Where the entry is displayed before the transaction executed by User 2 has been committed	Pass
		1. User 2 enters the following details in the update fields: • ID: 174202 • Title: Edited Entry • Genre: Comedy • Rank: 6.6 • Director: Gaba • Actor 1: Gonzales • Actor 2: Lee 2. User 2 clicks the Update button 3. User 2 enters “174202” in the search	The application should display one entry with the following attributes: • ID: 174202 • Title: Edited Entry • Year: 1976 • Genre: Comedy • Rank: 6.6 • Director: Gaba • Actor 1: Gonzales • Actor 2: Lee	Pass

		field 4. User 2 clicks the Search button		
		1. User 3 enters “174202” in the search field 2. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Edited Entry ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the entry is displayed before the transaction executed by User 2 has been committed	Pass
Simultaneous writes accomplished within the lock wait timeout interval should return consistent results	**Given** three users want to update data corresponding to the same ID **When** the users input the update details and click the update button simultaneously **Then** the application should display the data corresponding to the entered details of the user whose write request was processed last	1. User 1 enters the following details in the update fields: ● ID: 174202 ● Title: First Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 1 clicks the Update button 3. User 1 enters “174202” in the search field 4. User 1 clicks the	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass

		Search button		
		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Second Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button 3. User 2 enters “174202” in the search field 4. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass
		1. User 3 enters the following details in the update fields: ● ID: 174202 ● Title: Third Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 3 clicks the Update button 3. User 3 enters “174202” in the search field 4. User 3 clicks the	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass

		Search button		
Writes not executed within the lock wait timeout interval should be rejected	**Given** three users want to update data corresponding to the same ID **When** the users input the update details and click the update button simultaneously, and the duration of the first write transaction exceeds the lock wait timeout interval **Then** the application should execute the first write transaction but reject the remaining transactions	1. User 1 enters the following details in the update fields: ● ID: 174202 ● Title: First Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 1 clicks the Update button 3. User 1 enters “174202” in the search field 4. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: First Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the write transaction exceeds 100 seconds	Pass
		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Second Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button	The application should display an alert with the following message: “Our servers are busy at the moment. Please try again later.”	Pass

		1. User 3 enters the following details in the update fields: ● ID: 174202 ● Title: Third Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 3 clicks the Update button	The application should display an alert with the following message: “Our servers are busy at the moment. Please try again later.”	Pass

Suite 2: Read Committed Isolation Level

Test Case	Scenario	Input	Expected Result	Pass/Fail
Simultaneous reads should return consistent results	**Given** two users want to see data corresponding to the same ID **When** the users input the ID and click the search button simultaneously **Then** the application should display the same data for all users	1. User 1 enters “174202” in the search field 2. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 2 enters “174202” in the search field 2. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 3 enters “174202” in the search field 2. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10	Pass

			● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	
Simultaneous reading and writing should not result in a dirty read	**Given** two users read data that another user is currently updating **When** the users submit the read and write requests simultaneously **Then** the application should return consistent data for the read requests	1. User 1 enters “174202” in the search field 2. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Edited Entry ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button 3. User 2 enters “174202” in the search field 4. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Edited Entry ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee	Pass

		1. User 3 enters “174202” in the search field 2. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
Simultaneous writes accomplished within the lock wait timeout interval should return consistent results	**Given** three users want to update data corresponding to the same ID **When** the users input the update details and click the update button simultaneously **Then** the application should display the data corresponding to the entered details of the user whose write request was processed last	1. User 1 enters the following details in the update fields: ● ID: 174202 ● Title: First Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 1 clicks the Update button 3. User 1 enters “174202” in the search field 4. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass

		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Second Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button 3. User 2 enters “174202” in the search field 4. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass
		1. User 3 enters the following details in the update fields: ● ID: 174202 ● Title: Third Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 3 clicks the Update button 3. User 3 enters “174202” in the search field 4. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass

Writes not executed within the lock wait timeout interval should be rejected	**Given** three users want to update data corresponding to the same ID **When** the users input the update details and click the update button simultaneously, and the duration of the first write transaction exceeds the lock wait timeout interval **Then** the application should execute the first write transaction but reject the remaining transactions	1. User 1 enters the following details in the update fields: ● ID: 174202 ● Title: First Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 1 clicks the Update button 3. User 1 enters “174202” in the search field 4. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: First Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the write transaction exceeds 100 seconds	Pass
		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Second Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button	The application should display an alert with the following message: “Our servers are busy at the moment. Please try again later.”	Pass

		1. User 3 enters the following details in the update fields: ● ID: 174202 ● Title: Third Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 3 clicks the Update button	The application should display an alert with the following message: “Our servers are busy at the moment. Please try again later.”	Pass

Suite 3: Repeatable Read Isolation Level

Test Case	Scenario	Input	Expected Result	Pass/Fail
Simultaneous reads should return consistent results	**Given** two users want to see data corresponding to the same ID **When** the users input the ID and click the search button simultaneously **Then** the application should display the same data for all users	1. User 1 enters “174202” in the search field 2. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 2 enters “174202” in the search field 2. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 3 enters “174202” in the search field 2. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10	Pass

			● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	
Simultaneous reading and writing should not result in a dirty read	**Given** two users read data that another user is currently updating **When** the users submit the read and write requests simultaneously **Then** the application should return return consistent data for the read requests	1. User 1 enters “174202” in the search field 2. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Edited Entry ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button 3. User 2 enters “174202” in the search field 4. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Edited Entry ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee	Pass

		1. User 3 enters “174202” in the search field 2. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
Simultaneous writes accomplished within the lock wait timeout interval should return consistent results	**Given** three users want to update data corresponding to the same ID **When** the users input the update details and click the update button simultaneously **Then** the application should display the data corresponding to the entered details of the user whose write request was processed last	1. User 1 enters the following details in the update fields: ● ID: 174202 ● Title: First Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 1 clicks the Update button 3. User 1 enters “174202” in the search field 4. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass

		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Second Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button 3. User 2 enters “174202” in the search field 4. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass
		1. User 3 enters the following details in the update fields: ● ID: 174202 ● Title: Third Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 3 clicks the Update button 3. User 3 enters “174202” in the search field 4. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass

Writes not executed within the lock wait timeout interval should be rejected	**Given** three users want to update data corresponding to the same ID **When** the users input the update details and click the update button simultaneously, and the duration of the first write transaction exceeds the lock wait timeout interval **Then** the application should execute the first write transaction but reject the remaining transactions	1. User 1 enters the following details in the update fields: ● ID: 174202 ● Title: First Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 1 clicks the Update button 3. User 1 enters “174202” in the search field 4. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: First Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the write transaction exceeds 100 seconds	Pass
		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Second Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button	The application should display an alert with the following message: “Our servers are busy at the moment. Please try again later.”	Pass

		1. User 3 enters the following details in the update fields: ● ID: 174202 ● Title: Third Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 3 clicks the Update button	The application should display an alert with the following message: “Our servers are busy at the moment. Please try again later.”	Pass

Suite 4: Serializable Isolation Level

Test Case	Scenario	Input	Expected Result	Pass/Fail
Simultaneous reads should return consistent results	**Given** two users want to see data corresponding to the same ID **When** the users input the ID and click the search button simultaneously **Then** the application should display the same data for all users	1. User 1 enters “174202” in the search field 2. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 2 enters “174202” in the search field 2. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10 ● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	Pass
		1. User 3 enters “174202” in the search field 2. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Sample Entry ● Year: 1976 ● Genre: Drama ● Rank: 10	Pass

			● Director: Cua ● Actor 1: Gaba ● Actor 2: Gonzales	
Simultaneous reading and writing should result in lock contention issues	**Given** two users read data that another user is currently updating **When** the users submit the read and write requests simultaneously **Then** the application should return consistent data for the read requests	1. User 1 enters “174202” in the search field 2. User 1 clicks the Search button	The application should encounter a lock contention issue (This issue is resolved when the autocommit mode of the database is disabled, as detailed in the technical paper)	Pass
		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Edited Entry ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button 3. User 2 enters “174202” in the search field 4. User 2 clicks the Search button	The application should encounter a lock contention issue (This issue is resolved when the autocommit mode of the database is disabled, as detailed in the technical paper)	Pass

		1. User 3 enters “174202” in the search field 2. User 3 clicks the Search button	The application should encounter a lock contention issue (This issue is resolved when the autocommit mode of the database is disabled, as detailed in the technical paper)	Pass
Simultaneous writes accomplished within the lock wait timeout interval should return consistent results	**Given** three users want to update data corresponding to the same ID **When** the users input the update details and click the update button simultaneously **Then** the application should display the data corresponding to the entered details of the user whose write request was processed last	1. User 1 enters the following details in the update fields: ● ID: 174202 ● Title: First Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 1 clicks the Update button 3. User 1 enters “174202” in the search field 4. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass

		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Second Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button 3. User 2 enters “174202” in the search field 4. User 2 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass
		1. User 3 enters the following details in the update fields: ● ID: 174202 ● Title: Third Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 3 clicks the Update button 3. User 3 enters “174202” in the search field 4. User 3 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Third Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the transaction executed by User 3 is accomplished last	Pass

Writes not executed within the lock wait timeout interval should be rejected	**Given** three users want to update data corresponding to the same ID **When** the users input the update details and click the update button simultaneously, and the duration of the first write transaction exceeds the lock wait timeout interval **Then** the application should execute the first write transaction but reject the remaining transactions	1. User 1 enters the following details in the update fields: ● ID: 174202 ● Title: First Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 1 clicks the Update button 3. User 1 enters “174202” in the search field 4. User 1 clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: First Edit ● Year: 1976 ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee Where the write transaction exceeds 100 seconds	Pass
		1. User 2 enters the following details in the update fields: ● ID: 174202 ● Title: Second Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 2 clicks the Update button	The application should display an alert with the following message: “Our servers are busy at the moment. Please try again later.”	Pass

		1. User 3 enters the following details in the update fields: ● ID: 174202 ● Title: Third Edit ● Genre: Comedy ● Rank: 6.6 ● Director: Gaba ● Actor 1: Gonzales ● Actor 2: Lee 2. User 3 clicks the Update button	The application should display an alert with the following message: “Our servers are busy at the moment. Please try again later.”	Pass

B. GLOBAL FAILURE RECOVERY

Suite 1: All nodes available

Test Case	Scenario	Input	Expected Result	Pass/Fail
Home page should show all data by default	**Given** the user wants to see all the data in the database **When** the user opens the web application **Then** the application should display all of the data from the central node in a table	1. User navigates to https://cggl-distributed-db.herokuapp.com/	The application should display 173394 entries with the following attributes: ● ID ● Title ● Year ● Genre ● Rank ● Director ● Actor 1 ● Actor 2	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 1 ● Title: $ ● Year: 1971 ● Genre: Comedy ● Rank: 6.4 ● Director: Richard (I) Brooks ● Actor 1: Arthur Brauss ● Actor 2: Monica Stender	Pass

			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 173394 ● Title: . 19,99 ● Year: 1998 ● Genre: Comedy ● Rank: 6.3 ● Director: Mart Dominicus ● Actor 1: Abdenbi Azzaoui ● Actor 2: Thomas Acda	Pass
Displayed data should be filtered by ID when a search based on ID is performed	**Given** the user wants to see data corresponding to a particular ID **When** the user inputs an ID and clicks the search button **Then** the application should display the data from the central node matching the provided ID	1. User enters “171404” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171404 ● Title: Zeder ● Year: 1983 ● Genre: Horror ● Rank: 5.8 ● Director: Pupi Avati ● Actor 1: Adolfo Belletti ● Actor 2: Maria Teresa Tofano	Pass
Displayed data should be filtered by title when a search based on title is performed	**Given** the user wants to see data corresponding to a particular movie title **When** the user inputs a title and clicks the search	1. User enters “Zenobia” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171484 ● Title: Zenobia	Pass

	button **Then** the application should display the data from the central node matching the provided movie title		• Year: 1939 • Genre: Comedy • Rank: 5.8 • Director: Gordon Douglas • Actor 1: Alice Brady • Actor 2: Olga Zenner	
Displayed data should be filtered by genre when a search based on genre is performed	**Given** the user wants to see data corresponding to a particular movie genre **When** the user inputs a genre and clicks the search button **Then** the application should display the data from the central node matching the provided movie genre	1. User enters “Action” in the search field 2. User clicks the Search button	The application should display 11574 entries with the following common attribute value: • Genre: Action	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 18 • Title: $windle • Year: 2002 • Genre: Action • Rank: 5.4 • Director: K.C. Bascombe • Actor 1: Alain Goulem • Actor 2: Jack Daniel Wells	Pass
			The entry with the highest ID displayed in the application should have the following attribute values:	Pass

			• ID: 173317 • Title: Itimo Narco, El • Year: 1992 • Genre: Action • Rank: • Director: Víctor Herrera Zenil • Actor 1: Edgardo (I) Gazcón • Actor 2: Patricia Rivera	
Displayed data should be filtered by director when a search based on director is performed	**Given** the user wants to see data corresponding to a particular director **When** the user inputs the name of a director and clicks the search button **Then** the application should display the data from the central node matching the provided director	1. User enters “Naosuke Kurosawa” in the search field 2. User clicks the Search button	The application should display five entries with the following common attribute value: • Director: Naosuke Kurosawa	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 3826 • Title: Ajiu: yaru! • Year: 1981 • Genre: Drama • Rank: • Director: Naosuke Kurosawa • Actor 1: Jun Izumi • Actor 2: Rumi Tama	Pass
			The entry with the highest ID displayed in the	Pass

			application should have the following attribute values: ● ID: 171402 ● Title: Zeccho Shimai Ochiru ● Year: 1982 ● Genre: Drama ● Rank: ● Director: Naosuke Kurosawa ● Actor 1: Moeko Ezawa ● Actor 2:	
Displayed data should be filtered by actor when a search based on actor is performed	**Given** the user wants to see data corresponding to a particular actor **When** the user inputs the name of an actor and clicks the search button **Then** the application should display the data from the central node matching the provided actor	1. User enters “Austin Stoker” in the search field 2. User clicks the search Search button	The application should display nine entries with the following common attribute value: ● Actor 1: Austin Stoker	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 1848 ● Title: Abby ● Year: 1974 ● Genre: Horror ● Rank: 5.8 ● Director: William Girdler ● Actor 1: Austin Stoker ● Actor 2: Carol Speed	Pass

			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 171396 ● Title: Zebra Killer, The ● Year: 1974 ● Genre: Drama ● Rank: 6.7 ● Director: William Girdler ● Actor 1: Austin Stoker ● Actor 2: Carla Rueckert	Pass
New data should be reflected in the database when an insertion is performed	**Given** the user wants to add a new movie to the database **When** the user inputs the movie details and clicks the insert button **Then** the application should update the database to include the new entry	1. User enters the following details in the insert fields: ● Title: New Movie ● Year: 2022 ● Genre: Horror ● Rank: 4.0 ● Director: Josuke Higashikata ● Actor 1: Giorno Giovanna ● Actor 2: Jolyne Cujoh 2. User clicks the Insert button 3. User enters “New Movie” in the search field 4. User clicks the Search	The application should display one entry with the following attributes: ● ID: 174201 ● Title: New Movie ● Year: 2022 ● Genre: Horror ● Rank: 4 ● Director: Josuke Higashikata ● Actor 1: Giorno Giovanna ● Actor 2: Jolyne Cujoh	Pass
			The three nodes should have the following numbers of entries: ● Central Node: 173395 ● Node 2: 92003	Pass

		button	● Node 3: 81392	
			There should be no entries that are in both Node 2 and Node 3	Pass
Data should be updated in the database when an update is performed	**Given** the user wants to update movie details in the database **When** the user inputs the updated details and clicks the update button **Then** the application should update the attributes of the matching data entry	1. User enters the following details in the update fields: ● ID: 174201 ● Title: Updated Title ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar 2. User clicks the Update button 3. User enters “174201” in the search field 4. User clicks the Search button	The application should display one entry with the following attributes: ● ID: 174201 ● Title: Updated Title ● Year: 2022 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass
			The central node should have a data entry with the following attributes: ● ID: 174201 ● Title: Updated Title ● Year: 2022 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass

			Node 3 should have a data entry with the following attributes: ● ID: 174201 ● Title: Updated Title ● Year: 2022 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass
Data should be deleted from the database when a deletion is performed	**Given** the user wants to delete an existing movie from the database **When** the user inputs the movie ID and clicks the delete button **Then** the application should remove the matching data entry from the database	1. User enters “174201” in the delete field 2. User clicks the Delete button 3. User enters “174201” in the search field 4. User clicks the Search button	The application should display the string “No data available in table”	Pass
			The central node should not have a data entry with the ID “174201”	Pass
			Node 3 should not have a data entry with the ID “174201”	Pass

Suite 2: Central node down

Test Case	Scenario	Input	Expected Result	Pass/Fail
Home page should show all data by default	**Given** the user wants to see all the data in the database **When** the user opens the web application **Then** the application should display all of the data combined from nodes 2 and 3 in a table	1. User navigates to https://cggl-distributed-db.herokuapp.com/	The application should display 173394 entries with the following attributes: ● ID ● Title ● Year ● Genre ● Rank ● Director ● Actor 1 ● Actor 2	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 1 ● Title: $ ● Year: 1971 ● Genre: Comedy ● Rank: 6.4 ● Director: Richard (I) Brooks ● Actor 1: Arthur Brauss ● Actor 2: Monica Stender	Pass
			The entry with the highest ID displayed in the application should have	Pass

			the following attribute values: ● ID: 173394 ● Title: . 19,99 ● Year: 1998 ● Genre: Comedy ● Rank: 6.3 ● Director: Mart Dominicus ● Actor 1: Abdenbi Azzaoui ● Actor 2: Thomas Acda	
Displayed data should be filtered by ID when a search based on ID is performed	**Given** the user wants to see data corresponding to a particular ID **When** the user inputs an ID and clicks the search button **Then** the application should display the data combined from nodes 2 and 3 matching the provided ID	1. User enters “171404” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171404 ● Title: Zeder ● Year: 1983 ● Genre: Horror ● Rank: 5.8 ● Director: Pupi Avati ● Actor 1: Adolfo Belletti ● Actor 2: Maria Teresa Tofano	Pass
Displayed data should be filtered by title when a search based on title is performed	**Given** the user wants to see data corresponding to a particular movie title **When** the user inputs a title and clicks the search button **Then** the application should display the data combined from nodes 2	1. User enters “Zenobia” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171484 ● Title: Zenobia ● Year: 1939 ● Genre: Comedy ● Rank: 5.8 ● Director: Gordon	Pass

	and 3 matching the provided movie title		Douglas ● Actor 1: Alice Brady ● Actor 2: Olga Zenner	
Displayed data should be filtered by genre when a search based on genre is performed	**Given** the user wants to see data corresponding to a particular movie genre **When** the user inputs a genre and clicks the search button **Then** the application should display the data combined from nodes 2 and 3 matching the provided movie genre	1. User enters “Action” in the search field 2. User clicks the Search button	The application should display 11574 entries with the following common attribute value: ● Genre: Action	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 18 ● Title: $windle ● Year: 2002 ● Genre: Action ● Rank: 5.4 ● Director: K.C. Bascombe ● Actor 1: Alain Goulem ● Actor 2: Jack Daniel Wells	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 173317 ● Title: Itimo Narco, El ● Year: 1992 ● Genre: Action	Pass

			● Rank: ● Director:Víctor Herrera Zenil ● Actor 1: Edgardo (I) Gazcón ● Actor 2: Patricia Rivera	
Displayed data should be filtered by director when a search based on director is performed	**Given** the user wants to see data corresponding to a particular director **When** the user inputs the name of a director and clicks the search button **Then** the application should display the data combined from nodes 2 and 3 matching the provided director	1. User enters “Naosuke Kurosawa” in the search field 2. User clicks the Search button	The application should display five entries with the following common attribute value: ● Director: Naosuke Kurosawa	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 3826 ● Title: Ajiu: yaru! ● Year: 1981 ● Genre: Drama ● Rank: ● Director: Naosuke Kurosawa ● Actor 1: Jun Izumi ● Actor 2: Rumi Tama	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 171402	Pass

			• Title: Zeccho Shimai Ochiru • Year: 1982 • Genre: Drama • Rank: • Director: Naosuke Kurosawa • Actor 1: Moeko Ezawa • Actor 2:	
Displayed data should be filtered by actor when a search based on actor is performed	**Given** the user wants to see data corresponding to a particular actor **When** the user inputs the name of an actor and clicks the search button **Then** the application should display the data combined from nodes 2 and 3 matching the provided actor	1. User enters “Austin Stoker” in the search field 2. User clicks the Search button	The application should display nine entries with the following common attribute value: • Actor 1: Austin Stoker	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 1848 • Title: Abby • Year: 1974 • Genre: Horror • Rank: 5.8 • Director: William Girdler • Actor 1: Austin Stoker • Actor 2: Carol Speed	Pass
			The entry with the highest ID displayed in the application should have the following attribute	Pass

			values: ● ID: 171396 ● Title: Zebra Killer, The ● Year: 1974 ● Genre: Drama ● Rank: 6.7 ● Director: William Girdler ● Actor 1: Austin Stoker ● Actor 2: Carla Rueckert	
New data should be reflected in the database when an insertion is performed	**Given** the user wants to add a new movie to the database **When** the user inputs the movie details and clicks the insert button **Then** the application should update the database to include the new entry	1. User enters the following details in the insert fields: ● Title: New Movie ● Year: 2022 ● Genre: Horror ● Rank: 4.0 ● Director: Josuke Higashikata ● Actor 1: Giorno Giovanna ● Actor 2: Jolyne Cujoh 2. User clicks the Insert button 3. User enters “New Movie” in the search field 4. User clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: New Movie ● Year: 2022 ● Genre: Horror ● Rank: 4 ● Director: Josuke Higashikata ● Actor 1: Giorno Giovanna ● Actor 2: Jolyne Cujoh	Pass
			The three nodes should have the following numbers of entries: ● Central Node: 173394 ● Node 2: 92003 ● Node 3: 81392	Pass
			There should be no entries that are in both Node 2	Pass

			and Node 3	
Data should be updated in the database when an update is performed	**Given** the user wants to update movie details in the database **When** the user inputs the updated details and clicks the update button **Then** the application should update the attributes of the matching data entry	1. User enters the following details in the update fields: ● ID: 174202 ● Title: Updated Title ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar 2. User clicks the Update button 3. User enters “174202” in the search field 4. User clicks the Search button	The application should display one entry with the following attributes: ● ID: 174202 ● Title: Updated Title ● Year: 2022 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass
			The central node should not have a data entry with the ID “174202”	Pass
			Node 3 should have a data entry with the following attributes: ● ID: 174202 ● Title: Updated Title ● Year: 2022 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass

Data should be deleted from the database when a deletion is performed	**Given** the user wants to delete an existing movie from the database **When** the user inputs the movie ID and clicks the delete button **Then** the application should remove the matching data entry from the database	1. User enters “174202” in the delete field 2. User clicks the Delete button 3. User enters “174202” in the search field 4. User clicks the Search button	The application should display the string “No data available in table”	Pass
			Node 3 should not have a data entry with the ID “174202”	Pass

Suite 3: Node 2 down

Test Case	Scenario	Input	Expected Result	Pass/Fail
Home page should show all data by default	**Given** the user wants to see all the data in the database **When** the user opens the web application **Then** the application should display all of the data from the central node in a table	1. User navigates to https://cggl-distributed-db.herokuapp.com/	The application should display 173394 entries with the following attributes: ● ID ● Title ● Year ● Genre ● Rank ● Director ● Actor 1 ● Actor 2	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 1 ● Title: $ ● Year: 1971 ● Genre: Comedy ● Rank: 6.4 ● Director: Richard (I) Brooks ● Actor 1: Arthur Brauss ● Actor 2: Monica Stender	Pass
			The entry with the highest ID displayed in the application should have	Pass

			the following attribute values: ● ID: 173394 ● Title: . 19,99 ● Year: 1998 ● Genre: Comedy ● Rank: 6.3 ● Director: Mart Dominicus ● Actor 1: Abdenbi Azzaoui ● Actor 2: Thomas Acda	
Displayed data should be filtered by ID when a search based on ID is performed for a movie in the database	**Given** the user wants to see data corresponding to a particular ID **When** the user inputs an ID and clicks the search button **Then** the application should display the data from the central node matching the provided ID	1. User enters “171404” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171404 ● Title: Zeder ● Year: 1983 ● Genre: Horror ● Rank: 5.8 ● Director: Pupi Avati ● Actor 1: Adolfo Belletti ● Actor 2: Maria Teresa Tofano	Pass
Displayed data should be filtered by title when a search based on title is performed	**Given** the user wants to see data corresponding to a particular movie title **When** the user inputs a title and clicks the search button **Then** the application should display the data from the central node	1. User enters “Zenobia” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171484 ● Title: Zenobia ● Year: 1939 ● Genre: Comedy ● Rank: 5.8 ● Director: Gordon	Pass

	matching the provided movie title		Douglas ● Actor 1: Alice Brady ● Actor 2: Olga Zenner	
Displayed data should be filtered by genre when a search based on genre is performed	**Given** the user wants to see data corresponding to a particular movie genre **When** the user inputs a genre and clicks the search button **Then** the application should display the data from the central node matching the provided movie genre	1. User enters “Action” in the search field 2. User clicks the Search button	The application should display 11574 entries with the following common attribute value: ● Genre: Action	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 18 ● Title: $windle ● Year: 2002 ● Genre: Action ● Rank: 5.4 ● Director: K.C. Bascombe ● Actor 1: Alain Goulem ● Actor 2: Jack Daniel Wells	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 173317 ● Title: Itimo Narco, El ● Year: 1992 ● Genre: Action	Pass

			• Rank: • Director: Víctor Herrera Zenil • Actor 1: Edgardo (I) Gazcón • Actor 2: Patricia Rivera	
Displayed data should be filtered by director when a search based on director is performed	**Given** the user wants to see data corresponding to a particular director **When** the user inputs the name of a director and clicks the search button **Then** the application should display the data from the central node matching the provided director	1. User enters “Naosuke Kurosawa” in the search field 2. User clicks the Search button	The application should display five entries with the following common attribute value: • Director: Naosuke Kurosawa	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 3826 • Title: Ajiu: yaru! • Year: 1981 • Genre: Drama • Rank: • Director: Naosuke Kurosawa • Actor 1: Jun Izumi • Actor 2: Rumi Tama	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: • ID: 171402	Pass

			• Title: Zeccho Shimai Ochiru • Year: 1982 • Genre: Drama • Rank: • Director: Naosuke Kurosawa • Actor 1: Moeko Ezawa • Actor 2:	
Displayed data should be filtered by actor when a search based on actor is performed	**Given** the user wants to see data corresponding to a particular actor **When** the user inputs the name of an actor and clicks the search button **Then** the application should display the data from the central node matching the provided actor	1. User enters "Austin Stoker" in the search field 2. User clicks the Search button	The application should display nine entries with the following common attribute value: • Actor 1: Austin Stoker	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 1848 • Title: Abby • Year: 1974 • Genre: Horror • Rank: 5.8 • Director: William Girdler • Actor 1: Austin Stoker • Actor 2: Carol Speed	Pass
			The entry with the highest ID displayed in the application should have the following attribute	Pass

			values: ● ID: 171396 ● Title: Zebra Killer, The ● Year: 1974 ● Genre: Drama ● Rank: 6.7 ● Director: William Girdler ● Actor 1: Austin Stoker ● Actor 2: Carla Rueckert	
New data should be reflected in the database when an insertion is performed	**Given** the user wants to add a new movie to the database **When** the user inputs the movie details and clicks the insert button **Then** the application should update the database to include the new entry	1. User enters the following details in the insert fields: ● Title: New Movie ● Year: 1970 ● Genre: Horror ● Rank: 4.0 ● Director: Josuke Higashikata ● Actor 1: Giorno Giovanna ● Actor 2: Jolyne Cujoh 2. User clicks the Insert button 3. User enters “New Movie” in the search field 4. User clicks the Search button	The application should display one entry with the following attributes: ● ID: 174203 ● Title: New Movie ● Year: 1970 ● Genre: Horror ● Rank: 4 ● Director: Josuke Higashikata ● Actor 1: Giorno Giovanna ● Actor 2: Jolyne Cujoh	Pass
			The three nodes should have the following numbers of entries: ● Central Node: 173395 ● Node 2: 92003 ● Node 3: 81391	Pass
			There should be no entries that are in both Node 2	Pass

			and Node 3	
Data should be updated in the database when an update is performed	**Given** the user wants to update movie details in the database **When** the user inputs the updated details and clicks the update button **Then** the application should update the attributes of the matching data entry	1. User enters the following details in the update fields: ● ID: 174203 ● Title: Updated Title ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar 2. User clicks the Update button 3. User enters “174203” in the search field 4. User clicks the Search button	The application should display one entry with the following attributes: ● ID: 174203 ● Title: Updated Title ● Year: 1970 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass
			The central node should have a data entry with the following attributes: ● ID: 174203 ● Title: Updated Title ● Year: 1970 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass
			Node 2 should not have a data entry with the ID “174203”	Pass

Data should be deleted from the database when a deletion is performed	**Given** the user wants to delete an existing movie from the database **When** the user inputs the movie ID and clicks the delete button **Then** the application should remove the matching data entry from the database	1. User enters “174203” in the delete field 2. User clicks the Delete button 3. User enters “174203” in the search field 4. User clicks the Search button	The application should display the string “No data available in table”	Pass
			The central node should not have a data entry with the ID “174203”	Pass

Suite 4: Node 3 down

Test Case	Scenario	Input	Expected Result	Pass/Fail
Home page should show all data by default	**Given** the user wants to see all the data in the database **When** the user opens the web application **Then** the application should display all of the data from the central node in a table	1. User navigates to https://cggl-distributed-db.herokuapp.com/	The application should display 173394 entries with the following attributes: ● ID ● Title ● Year ● Genre ● Rank ● Director ● Actor 1 ● Actor 2	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 1 ● Title: $ ● Year: 1971 ● Genre: Comedy ● Rank: 6.4 ● Director: Richard (I) Brooks ● Actor 1: Arthur Brauss ● Actor 2: Monica Stender	Pass
			The entry with the highest ID displayed in the application should have	Pass

			the following attribute values: ● ID: 173394 ● Title: . 19,99 ● Year: 1998 ● Genre: Comedy ● Rank: 6.3 ● Director: Mart Dominicus ● Actor 1: Abdenbi Azzaoui ● Actor 2: Thomas Acda	
Displayed data should be filtered by ID when a search based on ID is performed for a movie in the database	**Given** the user wants to see data corresponding to a particular ID **When** the user inputs an ID and clicks the search button **Then** the application should display the data from the central node matching the provided ID	1. User enters “171404” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171404 ● Title: Zeder ● Year: 1983 ● Genre: Horror ● Rank: 5.8 ● Director: Pupi Avati ● Actor 1: Adolfo Belletti ● Actor 2: Maria Teresa Tofano	Pass
Displayed data should be filtered by title when a search based on title is performed	**Given** the user wants to see data corresponding to a particular movie title **When** the user inputs a title and clicks the search button **Then** the application should display the data from the central node	1. User enters “Zenobia” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171484 ● Title: Zenobia ● Year: 1939 ● Genre: Comedy ● Rank: 5.8 ● Director: Gordon	Pass

	matching the provided movie title		Douglas ● Actor 1: Alice Brady ● Actor 2: Olga Zenner	
Displayed data should be filtered by genre when a search based on genre is performed	**Given** the user wants to see data corresponding to a particular movie genre **When** the user inputs a genre and clicks the search button **Then** the application should display the data from the central node matching the provided movie genre	1. User enters “Action” in the search field 2. User clicks the Search button	The application should display 11574 entries with the following common attribute value: ● Genre: Action	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 18 ● Title: $windle ● Year: 2002 ● Genre: Action ● Rank: 5.4 ● Director: K.C. Bascombe ● Actor 1: Alain Goulem ● Actor 2: Jack Daniel Wells	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 173317 ● Title: Itimo Narco, El ● Year: 1992 ● Genre: Action	Pass

			• Rank: • Director: Víctor Herrera Zenil • Actor 1: Edgardo (I) Gazcón • Actor 2: Patricia Rivera	
Displayed data should be filtered by director when a search based on director is performed	**Given** the user wants to see data corresponding to a particular director **When** the user inputs the name of a director and clicks the search button **Then** the application should display the data from the central node matching the provided director	1. User enters “Naosuke Kurosawa” in the search field 2. User clicks the Search button	The application should display five entries with the following common attribute value: • Director: Naosuke Kurosawa	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 3826 • Title: Ajiu: yaru! • Year: 1981 • Genre: Drama • Rank: • Director: Naosuke Kurosawa • Actor 1: Jun Izumi • Actor 2: Rumi Tama	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: • ID: 171402	Pass

			• Title: Zeccho Shimai Ochiru • Year: 1982 • Genre: Drama • Rank: • Director: Naosuke Kurosawa • Actor 1: Moeko Ezawa • Actor 2:	
Displayed data should be filtered by actor when a search based on actor is performed	**Given** the user wants to see data corresponding to a particular actor **When** the user inputs the name of an actor and clicks the search button **Then** the application should display the data from the central node matching the provided actor	1. User enters "Austin Stoker" in the search field 2. User clicks the Search button	The application should display nine entries with the following common attribute value: • Actor 1: Austin Stoker	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 1848 • Title: Abby • Year: 1974 • Genre: Horror • Rank: 5.8 • Director: William Girdler • Actor 1: Austin Stoker • Actor 2: Carol Speed	Pass
			The entry with the highest ID displayed in the application should have the following attribute	Pass

			values: ● ID: 171396 ● Title: Zebra Killer, The ● Year: 1974 ● Genre: Drama ● Rank: 6.7 ● Director: William Girdler ● Actor 1: Austin Stoker ● Actor 2: Carla Rueckert	
New data should be reflected in the database when an insertion is performed	**Given** the user wants to add a new movie to the database **When** the user inputs the movie details and clicks the insert button **Then** the application should update the database to include the new entry	1. User enters the following details in the insert fields: ● Title: New Movie ● Year: 2022 ● Genre: Horror ● Rank: 4.0 ● Director: Josuke Higashikata ● Actor 1: Giorno Giovanna ● Actor 2: Jolyne Cujoh 2. User clicks the Insert button 3. User enters “New Movie” in the search field 4. User clicks the Search button	The application should display one entry with the following attributes: ● ID: 174204 ● Title: New Movie ● Year: 2022 ● Genre: Horror ● Rank: 4 ● Director: Josuke Higashikata ● Actor 1: Giorno Giovanna ● Actor 2: Jolyne Cujoh	Pass
			The three nodes should have the following numbers of entries: ● Central Node: 173395 ● Node 2: 92003 ● Node 3: 81391	Pass
			There should be no entries that are in both Node 2	Pass

			and Node 3	
Data should be updated in the database when an update is performed	**Given** the user wants to update movie details in the database **When** the user inputs the updated details and clicks the update button **Then** the application should update the attributes of the matching data entry	1. User enters the following details in the update fields: ● ID: 174204 ● Title: Updated Title ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar 2. User clicks the Update button 3. User enters “174204” in the search field 4. User clicks the Search button	The application should display one entry with the following attributes: ● ID: 174204 ● Title: Updated Title ● Year: 2022 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass
			The central node should have a data entry with the following attributes: ● ID: 174204 ● Title: Updated Title ● Year: 2022 ● Genre: Action ● Rank: 4.2 ● Director: Johnny Joestar ● Actor 1: Josuke Higashikata ● Actor 2: Shizuka Joestar	Pass
			Node 3 should not have a data entry with the ID “174204”	Pass

Data should be deleted from the database when a deletion is performed	**Given** the user wants to delete an existing movie from the database **When** the user inputs the movie ID and clicks the delete button **Then** the application should remove the matching data entry from the database	1. User enters “174204” in the delete field 2. User clicks the Delete button 3. User enters “174204” in the search field 4. User clicks the Search button	The application should display the string “No data available in table”	Pass
			The central node should not have a data entry with the ID “174204”	Pass

Suite 5: Central node and Node 2 down

Test Case	Scenario	Input	Expected Result	Pass/Fail
Home page should show all data by default	**Given** the user wants to see all the data in the database **When** the user opens the web application **Then** the application should display all of the data from Node 3 in a table	1. User navigates to https://cggl-distributed-db.herokuapp.com/	The application should display 81391 entries with the following attributes: ● ID ● Title ● Year ● Genre ● Rank ● Director ● Actor 1 ● Actor 2 Where the attribute values for "Year" are greater than or equal to 1980	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 6 ● Title: $30 ● Year: 1999 ● Genre: Comedy ● Rank: 7.5 ● Director: Gregory (I) Cooke ● Actor 1: Erik MacArthur ● Actor 2: Sara (I) Gilbert	Pass

			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 173394 ● Title: . 19,99 ● Year: 1998 ● Genre: Comedy ● Rank: 6.3 ● Director: Mart Dominicus ● Actor 1: Abdenbi Azzaoui ● Actor 2: Thomas Acda	Pass
Displayed data should be filtered by ID when a search based on ID is performed for a movie in Node 3	**Given** the user wants to see data corresponding to a particular ID for a movie released during or after 1980 **When** the user inputs an ID and clicks the search button **Then** the application should display the data from Node 3 matching the provided ID	1. User enters “171399” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171399 ● Title: Zebrahead ● Year: 1992 ● Genre: Drama ● Rank: 5.9 ● Director: Anthony Drazan ● Actor 1: Abdul Hassan Sharif ● Actor 2: N’Bushe Wright	Pass
No data should be displayed when a search based on ID is performed for a movie not in Node 3	**Given** the user wants to see data corresponding to a particular ID for a movie released before 1980 **When** the user inputs an	1. User enters “127773” in the search field 2. User clicks the Search button	The application should display the string “No data available in table”	Pass

	ID and clicks the search button **Then** the application should not display any data			
Displayed data should be filtered by title when a search based on title is performed for a movie in Node 3	**Given** the user wants to see data corresponding to a particular movie title released during or after 1980 **When** the user inputs a title and clicks the search button **Then** the application should display the data from Node 3 matching the provided movie title	1. User enters “Zeiramu” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171427 ● Title: Zeiramu ● Year: 1991 ● Genre: Action ● Rank: 6.8 ● Director: Keita Amamiya ● Actor 1: Edie Mirman ● Actor 2: Yûko Moriyama	Pass
No data should be displayed when a search based on title is performed for a movie not in Node 3	**Given** the user wants to see data corresponding to a particular ID for a movie released before 1980 **When** the user inputs an ID and clicks the search button **Then** the application should not display any data	1. User enters “Zebra in the Kitchen” in the search field 2. User clicks the Search button	The application should display the string “No data available in table”	Pass
Displayed data should be filtered by genre when a search based on genre is performed for movies in Node 3	**Given** the user wants to see data corresponding to a particular movie genre with movies released during or after 1980	1. User enters “Action” in the search field 2. User clicks the Search button	The application should display 7900 entries with the following common attribute value: ● Genre: Action	Pass

	When the user inputs a genre and clicks the search button **Then** the application should display the data from Node 3 matching the provided movie genre		Where the attribute values for “Year” are greater than or equal to 1980	
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 18 ● Title: $windle ● Year: 2002 ● Genre: Action ● Rank: 5.4 ● Director: K.C. Bascombe ● Actor 1: Alain Goulem ● Actor 2: Jack Daniel Wells	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 173317 ● Title: Itimo Narco, El ● Year: 1992 ● Genre: Action ● Rank: ● Director: Víctor Herrera Zenil ● Actor 1: Edgardo (I) Gazcón	Pass

			• Actor 2: Patricia Rivera	
Displayed data should be filtered by director when a search based on director is performed for movies in Node 3	**Given** the user wants to see data corresponding to a particular director with movies released during or after 1980 **When** the user inputs the name of a director and clicks the search button **Then** the application should display the data from Node 3 matching the provided director	1. User enters "Anthony Drazan" in the search field 2. User clicks the Search button	The application should display four entries with the following common attribute value: • Director: Anthony Drazan Where the attribute values for "Year" are greater than or equal to 1980	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 69666 • Title: Hurlyburly • Year: 1998 • Genre: Drama • Rank: 5.6 • Director: Anthony Drazan • Actor 1: Anna Paquin • Actor 2: Robin Wright Penn	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: • ID: 171399	Pass

			• Title: Zebrahead • Year: 1992 • Genre: Drama • Rank: 5.9 • Director: Anthony Drazan • Actor 1: Abdul Hassan Sharif • Actor 2: N'bushe Wright	
No data should be displayed when a search based on director is performed for movies not in Node 3	**Given** the user wants to see data corresponding to a particular director with movies released before 1980 **When** the user inputs an ID and clicks the search button **Then** the application should not display any data	1. User enters "Vasili Goncharov" in the search field 2. User clicks the Search button	The application should display the string "No data available in table"	Pass
Displayed data should be filtered by actor when a search based on actor is performed for movies in Node 3	**Given** the user wants to see data corresponding to a particular actor with movies released during or after 1980 **When** the user inputs the name of an actor and clicks the search button **Then** the application should display the data from Node 3 matching the provided actor	1. User enters "Mills Pierre" in the search field 2. User clicks the Search button	The application should display two entries with the following common attribute value: • Actor 1: Mills Pierre OR • Actor 2: Mills Pierre	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values:	Pass

			• ID: 122561 • Title: Quiet Storm, The • Year: 2004 • Genre: Thriller • Rank: • Director: Shango Bsang • Actor 1: Anthony Beselle • Actor 2: Mills Pierre	
			The entry with the highest ID displayed in the application should have the following attribute values: • ID: 171519 • Title: Zero Principal • Year: 2005 • Genre: Comedy • Rank: • Director: Helene Udy • Actor 1: Mills Pierre • Actor 2:	Pass
No data should be displayed when a search based on actor is performed for movies not in Node 3	**Given** the user wants to see data corresponding to a particular actor with movies released before 1980 **When** the user inputs the name of an actor and clicks the search button **Then** the application should not display any	1. User enters “Alfred Paget” in the search field 2. User clicks the Search button	The application should display the string “No data available in table”	Pass

	data			

Suite 6: Central node and Node 3 down

Test Case	Scenario	Input	Expected Result	Pass/Fail
Home page should show all data by default	**Given** the user wants to see all the data in the database **When** the user opens the web application **Then** the application should display all of the data from Node 2 in a table	1. User navigates to https://cggl-distributed-db.herokuapp.com/	The application should display 92003 entries with the following attributes: ● ID ● Title ● Year ● Genre ● Rank ● Director ● Actor 1 ● Actor 2 Where the attribute values for "Year" are less than 1980	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 1 ● Title: $ ● Year: 1971 ● Genre: Comedy ● Rank: 6.4 ● Director: Richard (I) Brooks ● Actor 1: Arthur Brauss ● Actor 2: Monica Stender	Pass

			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 173392 ● Title: nz de milharim ● Year: 1965 ● Genre: Adventure ● Rank: ● Director: Bilge Olgaç ● Actor 1: Aliye Rona ● Actor 2: Tuncel Kurtiz	Pass
Displayed data should be filtered by ID when a search based on ID is performed for a movie in Node 2	**Given** the user wants to see data corresponding to a particular ID for a movie released before 1980 **When** the user inputs an ID and clicks the search button **Then** the application should display the data from Node 3 matching the provided ID	1. User enters "124440" in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 124440 ● Title: Red Man's View, The ● Year: 1909 ● Genre: Short ● Rank: 4.4 ● Director: D.W. Griffith ● Actor 1: Alfred Paget ● Actor 2: Dorothy (I) West	Pass
No data should be displayed when a search based on ID is performed for a movie not in Node 2	**Given** the user wants to see data corresponding to a particular ID for a movie released during or after 1980 **When** the user inputs an ID and clicks the search	1. User enters "171399" in the search field 2. User clicks the Search button	The application should display the string "No data available in table"	Pass

	button **Then** the application should not display any data			
Displayed data should be filtered by title when a search based on title is performed for a movie in Node 2	**Given** the user wants to see data corresponding to a particular movie title released before 1980 **When** the user inputs a title and clicks the search button **Then** the application should display the data from Node 2 matching the provided movie title	1. User enters “In the Border States” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 72024 ● Title: In the Border States ● Year: 1910 ● Genre: Drama ● Rank: 6.2 ● Director: D.W. Griffith ● Actor 1: Alfred Paget ● Actor 2: Dorothy (I) West	Pass
No data should be displayed when a search based on title is performed for a movie not in Node 2	**Given** the user wants to see data corresponding to a particular ID for a movie released during or after 1980 **When** the user inputs an ID and clicks the search button **Then** the application should not display any data	1. User enters “Zen Noir” in the search field 2. User clicks the Search button	The application should display the string “No data available in table”	Pass
Displayed data should be filtered by genre when a search based on genre is performed for movies in	**Given** the user wants to see data corresponding to a particular movie genre with movies released	1. User enters “Action” in the search field 2. User clicks the Search button	The application should display 3674 entries with the following common attribute value:	Pass

Node 2	before 1980 **When** the user inputs a genre and clicks the search button **Then** the application should display the data from Node 2 matching the provided movie genre		● Genre: Action Where the attribute values for “Year” are less than 1980	
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 169 ● Title: …e cos divinnero i tre supermen del West ● Year: 1973 ● Genre: Action ● Rank: ● Director: Anthony Blond ● Actor 1: Ágata Lys ● Actor 2:	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 173267 ● Title: Itima lucha, La ● Year: 1959 ● Genre: Action ● Rank: ● Director: Julián Soler ● Actor 1: Ángel (I) Fernández	Pass

			• Actor 2: Ricardo Adalid	
Displayed data should be filtered by director when a search based on director is performed for movies in Node 2	**Given** the user wants to see data corresponding to a particular director with movies released before 1980 **When** the user inputs the name of a director and clicks the search button **Then** the application should display the data from Node 2 matching the provided director	1. User enters "D.W. Griffith" in the search field 2. User clicks the Search button	The application should display 490 entries with the following common attribute value: • Director: D.W. Griffith Where the attribute values for "Year" are less than 1980	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 2035 • Title: Abraham Lincoln • Year: 1930 • Genre: Drama • Rank: 5.6 • Director: D.W. Griffith • Actor 1: Cameron Prud'Homme • Actor 2: Helen Ware	Pass
			The entry with the highest ID displayed in the application should have the following attribute values:	Pass

			• ID: 172201 • Title: Zulu's Heart, The • Year: 1908 • Genre: Action • Rank: • Director: D.W. Griffith • Actor 1:Alfred Paget • Actor 2: Harry Solter	
No data should be displayed when a search based on director is performed for movies not in Node 2	**Given** the user wants to see data corresponding to a particular director with movies released during or after 1980 **When** the user inputs an ID and clicks the search button **Then** the application should not display any data	1. User enters "Helene Udy" in the search field 2. User clicks the Search button	The application should display the string "No data available in table"	Pass
Displayed data should be filtered by actor when a search based on actor is performed for movies in Node 2	**Given** the user wants to see data corresponding to a particular actor with movies released before 1980 **When** the user inputs the name of an actor and clicks the search button **Then** the application should display the data from Node 2 matching the provided actor	1. User enters "Alfred Paget" in the search field 2. User clicks the Search button	The application should display 158 entries with the following common attribute value: • Actor 1: Alfred Paget	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 2888 • Title: Adventures of	Pass

			Billy, The ● Year: 1911 ● Genre: Drama ● Rank: ● Director: D.W. Griffith ● Actor 1: Alfred Paget ● Actor 2: Claire McDowell	
			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 172201 ● Title: Zulu's Heart, The ● Year: 1908 ● Genre: Action ● Rank: ● Director: D.W. Griffith ● Actor 1:Alfred Paget ● Actor 2: Harry Solter	Pass
No data should be displayed when a search based on actor is performed for movies not in Node 2	**Given** the user wants to see data corresponding to a particular actor with movies released during or after 1980 **When** the user inputs the name of an actor and clicks the search button **Then** the application should not display any	1. User enters "Aleksei Panin" in the search field 2. User clicks the Search button	The application should display the string "No data available in table"	Pass

	data			

Suite 7: Node 2 and Node 3 down

Test Case	Scenario	Input	Expected Result	Pass/Fail
Home page should show all data by default	**Given** the user wants to see all the data in the database **When** the user opens the web application **Then** the application should display all of the data from the central node in a table	1. User navigates to https://cggl-distributed-db.herokuapp.com/	The application should display 173394 entries with the following attributes: ● ID ● Title ● Year ● Genre ● Rank ● Director ● Actor 1 ● Actor 2	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 1 ● Title: $ ● Year: 1971 ● Genre: Comedy ● Rank: 6.4 ● Director: Richard (I) Brooks ● Actor 1: Arthur Brauss ● Actor 2: Monica Stender	Pass
			The entry with the highest ID displayed in the application should have	Pass

			the following attribute values: ● ID: 173394 ● Title: . 19,99 ● Year: 1998 ● Genre: Comedy ● Rank: 6.3 ● Director: Mart Dominicus ● Actor 1: Abdenbi Azzaoui ● Actor 2: Thomas Acda	
Displayed data should be filtered by ID when a search based on ID is performed	**Given** the user wants to see data corresponding to a particular ID **When** the user inputs an ID and clicks the search button **Then** the application should display the data from the central node matching the provided ID	1. User enters “171404” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171404 ● Title: Zeder ● Year: 1983 ● Genre: Horror ● Rank: 5.8 ● Director: Pupi Avati ● Actor 1: Adolfo Belletti ● Actor 2: Maria Teresa Tofano	Pass
Displayed data should be filtered by title when a search based on title is performed	**Given** the user wants to see data corresponding to a particular movie title **When** the user inputs a title and clicks the search button **Then** the application should display the data	1. User enters “Zenobia” in the search field 2. User clicks the Search button	The application should display one entry with the following attribute values: ● ID: 171484 ● Title: Zenobia ● Year: 1939 ● Genre: Comedy ● Rank: 5.8	Pass

	from the central node matching the provided movie title		● Director: Gordon Douglas ● Actor 1: Alice Brady ● Actor 2: Olga Zenner	
Displayed data should be filtered by genre when a search based on genre is performed	**Given** the user wants to see data corresponding to a particular movie genre **When** the user inputs a genre and clicks the search button **Then** the application should display the data from the central node matching the provided movie genre	1. User enters “Action” in the search field 2. User clicks the Search button	The application should display 11574 entries with the following common attribute value: ● Genre: Action	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: ● ID: 18 ● Title: $windle ● Year: 2002 ● Genre: Action ● Rank: 5.4 ● Director: K.C. Bascombe ● Actor 1: Alain Goulem ● Actor 2: Jack Daniel Wells	Pass
			The entry with the highest ID displayed in the application should have the following attribute values: ● ID: 173317 ● Title: Itimo Narco, El ● Year: 1992	Pass

			• Genre: Action • Rank: • Director: Víctor Herrera Zenil • Actor 1: Edgardo (I) Gazcón • Actor 2: Patricia Rivera	
Displayed data should be filtered by director when a search based on director is performed	**Given** the user wants to see data corresponding to a particular director **When** the user inputs the name of a director and clicks the search button **Then** the application should display the data from the central node matching the provided director	1. User enters “Naosuke Kurosawa” in the search field 2. User clicks the Search button	The application should display five entries with the following common attribute value: • Director: Naosuke Kurosawa	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 3826 • Title: Ajiu: yaru! • Year: 1981 • Genre: Drama • Rank: • Director: Naosuke Kurosawa • Actor 1: Jun Izumi • Actor 2: Rumi Tama	Pass
			The entry with the highest ID displayed in the application should have the following attribute values:	Pass

			• ID: 171402 • Title: Zeccho Shimai Ochiru • Year: 1982 • Genre: Drama • Rank: • Director: Naosuke Kurosawa • Actor 1: Moeko Ezawa • Actor 2:	
Displayed data should be filtered by actor when a search based on actor is performed	**Given** the user wants to see data corresponding to a particular actor **When** the user inputs the name of an actor and clicks the search button **Then** the application should display the data from the central node matching the provided actor	1. User enters “Austin Stoker” in the search field 2. User clicks the search Search button	The application should display nine entries with the following common attribute value: • Actor 1: Austin Stoker	Pass
			The entry with the lowest ID displayed in the application should have the following attribute values: • ID: 1848 • Title: Abby • Year: 1974 • Genre: Horror • Rank: 5.8 • Director: William Girdler • Actor 1: Austin Stoker • Actor 2: Carol Speed	Pass
			The entry with the highest ID displayed in the application should have	Pass

			the following attribute values: ● ID: 171396 ● Title: Zebra Killer, The ● Year: 1974 ● Genre: Drama ● Rank: 6.7 ● Director: William Girdler ● Actor 1: Austin Stoker ● Actor 2: Carla Rueckert	