localhost:3000	Step	Description	Status	Expected Result	Actual Result
Test Case 1	1	Valid input-API healthy check			
	1.1	**GET** http://localhost:3000/			
	1.2	headers input valid API key key：x-api-key value：MyUniqueApiKey			
	1.3	response status code 200	passed	status code 200	status code 200
Test Case 2	2	Invalid input-Call API without auth key			
	2.1	**GET** http://localhost:3000/			
	2.2	headers input invalid API key key：x-api-key value：0000000000			
	2.3	response status code 401	passed	status code 401	status code 401
	2.4	response Invalid API key	passed	Invalid API key	Invalid API key
Test Case 3	3	Valid input-Register a user in correct way			
	3.1	**POST** http://localhost:3000/api/users/register			
	3.2	input Pre-requsets Script let email in dynamic with correct type and set email into environment user.email			
	3.3	input correct type email and password in body			
	3.4	response status code 201	passed	status code 201	status code 201
	3.5	check data have an id paired with the input email	passed	data have an id paired with the input email	data have an id paired with the input email
	3.6	check data email is equal the user.email which set in environment	passed	data email is equal the user.email which set in environment	data email is equal the user.email which set in environment
	3.7	make id set into environment user.id	passed	id set into environment user.id	id set into environment user.id
Test Case 4	4	Invalid input-Register a user with a already signed email			
	4.1	**POST** http://localhost:3000/api/users/register			
	4.2	input already signed email and password email：use test case 3 email which set in environment user.email			
	4.3	response status code 400	passed	status code 400	status code 400 Bad Request
	4.4	response Email address already in use	passed	Email address already in use	Email address already in use
Test Case 5	5	Invalid input-Register a user with a wrong type email			
	5.1	**POST** http://localhost:3000/api/users/register			
	5.2	input wrong type email and password email：abc12876.gro.com password：nascar123			
	5.3	response status code 400	failed	status code 400	status code 201
	5.4	response invalid email	failed	invalid email	created user
Test Case 6	6	Valid input-Login user in correct way			
	6.1	**POST** http://localhost:3000/api/users/tokens			
	6.2	input already signed email and password email：use test case 3 email which set in environment user.email			
	6.3	response status code 200	passed	status code 200	status code 200
	6.4	response token	passed	got a token	got a token
	6.5	check data has token	passed	data has token	data has token
	6.6	make data token set into environment user1.token	passed	data token set into environment user1.token	data token set into environment user1.token
Test Case 7	7	Invalid input-Login user with unsigned email			
	7.1	**POST** http://localhost:3000/api/users/tokens			
	7.2	input unsigned email and password email：0000abcd@gmail.com password：nascar123			
	7.3	response status code 400	passed	status code 400	status code 400
	7.4	response Either email or password is invalid	passed	Either email or password is invalid	Either email or password is invalid
	7.5	check response not have token	passed	response not have token	response not have token
	8	Invalid input-Login user with wrong type email			
	8.1	**POST** http://localhost:3000/api/users/tokens			

Test Case 8	8.2	input wrong type email and password email：abc0000000.gro.com password：nascar123			
	8.3	response status code 400	passed	status code 400	status code 400
	8.4	response Either email or password is invalid	passed	invalid email	Either email or password is invalid
	8.5	check response not have token	passed	response not have token	response not have token
Test Case 9	9	Valid input-Create new movie into list for user in correct way			
	9.1	**POST** http://localhost:3000/api/movies			
	9.2	Body must be JSON.			
	9.3	Adds a new movie for the current user.			
	9.4	Authorization Bearer token required in the headers.			
	9.5	User id comes from the token.			
	9.6	response status code 201	passed	status code 201	status code 201
	9.7	response the new movie data	passed	got a new movie data	got a new movie data
	9.8	make data id set into environment movie1.user1	passed	data id set into environment movie1.user1	data id set into environment movie1.user1
	9.9	check respone data has keys inciude id,title,year	passed	respone data has keys inciude id,title,year	respone data has keys inciude id,title,year
Test Case 10	10	Invalid input-Create new movie into list for user with worng body type			
	10.1	**POST** http://localhost:3000/api/movies			
	10.2	Body is not JSON.			
	10.3	Adds a new movie for the current user.			
	10.4	Authorization Bearer token required in the headers.			
	10.5	User id comes from the token.			
	10.6	response status code 400	failed	status code 400	status code 201
	10.7	response body is not JSON	failed	body is not JSON	got a new movie id
	10.8	check respone data not have any keys inciude id,title,year	failed	respone data not any keys inciude id,title,year	respone data have id
Test Case 11	11	Invalid input-Create new movie into list for user without token			
	11.1	**POST** http://localhost:3000/api/movies			
	11.2	Body must be JSON.			
	11.3	Adds a new movie for the current user.			
	11.4	no input token in the headers and Authorization Bearer token			
	11.5	response status code 401	failed	status code 401	status code 500
	11.6	response Invaild id	failed	invaild id	TypeError: Cannot read properties of undefined
Test Case 12	12	Valid input-Get list of all movies for user in correct way			
	12.1	**GET** http://localhost:3000/api/movies			
	12.2	No body required			
	12.3	Authorization Bearer token required in the headers			
	12.4	response status code 200	passed	status code 200	status code 200
	12.5	response list of all movies for the current user based on the id in the signed token.	passed	return list of all movies for the current user	list of all movies for the current user
	12.6	check response data is an array	passed	response data is an array	response data is an array
	12.7	checek response data array all objects have keys inciude id,title,year,owner	passed	response data array all objects have keys inciude id,title,year,owner	response data array all objects have keys inciude id,title,year,owner
Test Case 13	13	Invalid input-Get list of all movies for user without token			
	13.1	**GET** http://localhost:3000/api/movies			
	13.2	No body required			
	13.3	no input Bearer token in the headers and Authorization			
	13.4	response status code 401	failed	status code 401	status code 500
	13.5	response Invaild id	failed	invaild id	TypeError: Cannot read properties of undefined
Test Case 14	14	Valid input-Get specific movie information in correct way			
	14.1	**GET** http://localhost:3000/api/movies/:id			
	14.2	No body required.			
	14.3	Authorization Bearer token required in the headers.			
	14.4	input the specific movie's valid id in url			
	14.5	response status code 200	passed	status code 200	status code 200

	14.6	response get details of specific movie for current user, based on the token.	passed	get details of the specific movie	get details of the specific movie
	14.7	checek response data have keys inciude id,title,year,owner	passed	response data have keys inciude id,title,year,owner	response data have keys inciude id,title,year,owner
	14.8	check the user id is paired with owner	passed	the user id is paired with owner	the user id is paired with owner
Test Case 15	15	Invalid input-Get specific movie information with invaild id			
	15.1	**GET** http://localhost:3000/api/movies/:id			
	15.2	No body required.			
	15.3	Authorization Bearer token required in the headers.			
	15.4	input a invalid id in url			
	15.5	response status code 400	passed	status code 400	status code 400
	15.6	response no match for specified movie id	passed	No match for specified movie id	No match for specified movie id
	15.7	check respone data not have any keys inciude id,title,year,owner	passed	respone data not have any keys inciude id,title,year,owner	respone data not have any keys inciude id,title,year,owner
Test Case 16	16	Invalid input-Get specific movie information without token			
	16.1	**GET** http://localhost:3000/api/movies/:id			
	16.2	No body required			
	16.3	no input Bearer token in the headers and Authorization			
	16.4	input the specific movie's valid id in url			
	16.5	response status code 401	failed	status code 401	status code 500
	16.6	response Invaild id	failed	invaild id	TypeError: Cannot read properties of undefined
Test Case 17	17	Valid input-Delete specific movie with valid id			
	17.1	**DELETE** http://localhost:3000/api/movies/:id			
	17.2	No body required			
	17.3	Authorization Bearer token required in the headers.			
	17.4	input the movie's valid id which will be deleted in url			
	17.5	The owner of the movie must match the user id in the token.			
	17.6	response status code 200	passed	status code 200	status code 200
	17.7	response delete success	passed	Delete Success	Delete Success
	17.8	check response data id is the same with url	passed	response data id is the same with url	response data id is the same with url
Test Case 18	18	Invalid input-Delete specific movie with invalid id			
	18.1	**DELETE** http://localhost:3000/api/movies/:id			
	18.2	No body required			
	18.3	Authorization Bearer token required in the headers.			
	18.4	input the movie's invalid id in url			
	18.5	status code 400	passed	status code 400	status code 400
	18.6	response invalid movie id	passed	invalid movie id	invalid movie id
Test Case 19	19	Invalid input-Delete specific movie without token			
	19.1	**DELETE** http://localhost:3000/api/movies/:id			
	19.2	No body required			
	19.3	no input Authorization Bearer token in the headers and Authorization .			
	19.4	input the movie's valid id which will be deleted in url			
	19.5	response status code 401	failed	status code 401	status code 500
	19.6	response Invaild id	failed	invaild id	TypeError: Cannot read properties of undefined