Users – Document the Testing Process

#	Endpoint	Purpose	Input Data	Expected Output	Actual Output	Result
1	`POST /api/v1/users/`	Create a new user profile	`{ "first_name": "Jane", "last_name": "Doe", "email": "jane.doe@example.com", "phone_number": "+6123456789", "encrypted_password": "12345678" }`	Status: `201` Response: user fields match input	Status: `201` Response matches expected	✅ Passed
2	`POST /api/v1/users/`	Validate first name shorter than 2 characters	`{ "first_name": "D", "last_name": "Johnson", ... }`	Status: `400`	Status: `400`	✅ Passed
3	`POST /api/v1/users/`	Validate first name is not a string	`{ "first_name": 12345, "last_name": "Johnson", ... }`	Status: `400`	Status: `400`	✅ Passed
4	`POST /api/v1/users/`	Validate last name shorter than 2 characters	`{ "first_name": "Alice", "last_name": "L", ... }`	Status: `400`	Status: `400`	✅ Passed

5	`POST /api/v1/users/`	Validate last name is not a string	`{ "first_name": "Alice", "last_name": 98765, ... }`	Status: `400`	Status: `400`	✅ Passed
6	`POST /api/v1/users/`	Prevent duplicate email usage	Second request with same `"email"`	Status: `400`	Status: `400`	✅ Passed
7	`POST /api/v1/users/`	Validate invalid email format	`{ "email": "user.example.com" }`	Status: `400`	Status: `400`	✅ Passed
8	`POST /api/v1/users/`	Validate invalid phone number format	`{ "phone_number": "123ABC456" }`	Status: `400`	Status: `400`	✅ Passed
9	`POST /api/v1/users/`	Validate short password (too short)	`{ "encrypted_password": "123" }`	Status: `400`	Status: `400`	✅ Passed
10	`GET /api/v1/users/<user_id>`	Retrieve user by ID	Valid `user_id` from created user	Status: `200` Response: user fields match expected values	Status: `200` Response matches expected	✅ Passed

11	`GET /api/v1/users/invalidID`	Handle invalid user ID lookup	`"invalidID"`	Status: `404`	Status: `404`	✅ Passed
12	`PUT /api/v1/users/<user_id>`	Update existing user details	`{ "first_name": "John", "last_name": "Smith", "email": "john.smith@example.com", "phone_number": "+61123456788" }`	Status: `200` Updated fields match input	Status: `200` Response matches expected	✅ Passed
13	`PUT /api/v1/users/invalidID`	Handle update with invalid ID	`{ "first_name": "John", "last_name": "Smith", ... }`	Status: `404`	Status: `404`	✅ Passed
14	`PUT /api/v1/users/<user_id>`	Validate update with invalid input	`{ "first_name": "", "last_name": "Smith", ... }`	Status: `400`	Status: `400`	✅ Passed