

# 1. Introduction

## Purpose:

Define and execute a testing strategy to validate backend functionality, API correctness, and non-functional performance under load.

## Scope:

In-scope: FastAPI backend routers + service layer logic.

Out of scope: Frontend, UI, and any unimplemented features.

## Tools Used:

- pytest (unit + API tests)
- K6 (performance testing)
- Docker (for running K6)

## References:

- D2 (SRS):  
[https://github.com/AliBerkaySeucen/YZM2021-Team-3/blob/main/other\\_files/Team-3\\_D2\\_SRS%20-son.pdf](https://github.com/AliBerkaySeucen/YZM2021-Team-3/blob/main/other_files/Team-3_D2_SRS%20-son.pdf)
  - D3 (SDD):  
[https://github.com/AliBerkaySeucen/YZM2021-Team-3/blob/main/other\\_files/Team3-D3\\_SDD.zip](https://github.com/AliBerkaySeucen/YZM2021-Team-3/blob/main/other_files/Team3-D3_SDD.zip)
- 

# 2. Test Strategy

## 2.1 Unit Testing Strategy

- **Approach:** Test service layer functions in isolation.
- **Mocking:** Supabase calls mocked to isolate DB/network.
- **Coverage Goal:** Core service logic and error handling paths.

## 2.2 Integration/API Testing Strategy

- **Approach:** Test FastAPI router endpoints using `TestClient`.







- **Environment:** Dependency overrides for auth; mocked service methods.

### 2.3 Performance Testing Strategy

- **Target Endpoint:** `/nodes/get_node_info`
  - **Tool:** K6
  - **Metrics:** p95 latency under load, error rate < 1%.
- 

## 3. Test Cases

Functional Test Case Table (Backend)

Test ID	Layer	Endpoint / Function	Scenario	Expected Result	Actual Result	Status
TC-01	Router	<code>POST /users/create_user</code>	Valid user payload	201 Created	User created	 Pass
TC-02	Router	<code>POST /users/create_user</code>	Missing fields	422 Validation Error	422 returned	 Pass
TC-03	Router	<code>POST /users/get_access_token</code>	Valid credentials	Access token returned	Token returned	 Pass
TC-04	Router	<code>POST /users/get_access_token</code>	Invalid credentials	404 / Error	Error returned	 Pass
TC-05	Router	<code>POST /images/get_upload_url</code>	Authorized request	Signed URL	URL returned	 Pass
TC-06	Router	<code>POST /nodes/create_node</code>	Valid node payload	Node created	Node created	 Pass

TC-07	Router	POST /nodes/create_node	Missing fields	422 Error	422 returned	✓ Pass
TC-08	Router	POST /links/create_link	Valid payload	Link created	Link created	✓ Pass
TC-09	Router	POST /links/create_link	Missing fields	422 Error	422 returned	✓ Pass

### Service Layer Test Cases

Test ID	Service	Scenario	Expected Result	Actual Result	Status
TC-10	UserService	Create user (DB OK)	User saved	User saved	✓ Pass
TC-11	UserService	Create user (DB error)	Exception raised	Exception raised	✓ Pass
TC-12	UserService	Login valid password	Token returned	Token returned	✓ Pass
TC-13	UserService	Login invalid password	Auth error	Error raised	✓ Pass
TC-14	ImageService	Generate upload URL	URL returned	URL returned	✓ Pass
TC-15	ImageService	Upload URL failure	HTTPException	Exception raised	✓ Pass
TC-16	NodeService	Update node	Node updated	Update OK	✓ Pass
TC-17	NodeService	Update node DB error	Exception	Exception raised	✓ Pass
TC-18	LinkService	Delete link	Link deleted	Delete OK	✓ Pass
TC-19	LinkService	Delete link DB error	Exception	Exception raised	✓ Pass

## 4. Test Implementation

### 4.1 Unit Tests

#### Component 1: UserService (`services/user_services.py`)

Scenario: Login invalid password

Implementation (from `test_services.py`):

```
def test_login_user_invalid_password(monkeypatch):
    payload = UserLogin(email="sam@example.com", password="wrong")
    response = DummyResponse([{"user_id": 12, "password_hash": "hashed-secret"}])
    supabase_stub = SupabaseStub(table_chain=TableChain(response=response))

    monkeypatch.setattr(user_services, "supabase", supabase_stub)
    monkeypatch.setattr(
        user_services.security_service, "verify_password", lambda raw, hashed: False
    )

    service = UserService()
    with pytest.raises(HTTPException) as exc_info:
        service.login_user(payload)

    assert exc_info.value.detail == "Email or password do not match!"
```

---

#### Component 2: ImageService (`services/image_services.py`)

Scenario: Fail to generate upload URL

Implementation:

```
def test_get_upload_url_no_response(monkeypatch):
    payload = ImageFilename(user_id=1, file_name="photo.png")
    supabase_stub = SupabaseStub(storage=StorageStub(None))

    monkeypatch.setattr(image_services, "supabase", supabase_stub)

    service = ImageService()
    with pytest.raises(HTTPException) as exc_info:
        service.get_upload_url(payload)

    assert exc_info.value.detail == "Failed to generate URL from Supabase"
```

## Component 3: NodeService ([services/node\\_services.py](#))

**Scenario:** Update node — DB error

**Implementation:**

```
def test_update_node_supabase_error(monkeypatch):
    payload = NodeUpdate(
        user_id=1, node_id="node-1", image_id="img-1", description="desc"
    )
    supabase_stub = SupabaseStub(table_chain=TableChain(exc=RuntimeError("db error")))

    monkeypatch.setattr(node_services, "supabase", supabase_stub)

    service = NodeService()
    monkeypatch.setattr(
        service,
        "_wrap_node_op",
        lambda payload: {
            node_df.user_id: 1,
            node_df.node_id: "node-1",
            node_df.image_id: "img-1",
            node_df.description: "desc",
        },
    )

    with pytest.raises(RuntimeError, match="db error"):
        service.update_node(payload)
```

---

## Component 4: LinkService ([services/link\\_services.py](#))

**Scenario:** Delete link — DB error

**Implementation:**

```
def test_delete_link_supabase_error(monkeypatch):
    payload = NodeLinkDelete(user_id=1, link_id=3)
    supabase_stub = SupabaseStub(table_chain=TableChain(exc=RuntimeError("delete fail")))

    monkeypatch.setattr(link_services, "supabase", supabase_stub)

    service = LinkService()
    with pytest.raises(RuntimeError, match="delete fail"):
        service.delete_link(payload)
```

---

## Component 5: Router/API (FastAPI) (`routers/users.py`)

**Scenario:** Create user with missing fields

**Implementation** (from `test_routers.py`):

```
def test_create_user_missing_fields_returns_422(client):
    payload = {"first_name": "Ada", "surname": "Lovelace", "password": "secret"}
    response = client.post("/users/create_user", json=payload)

    assert response.status_code == 422
```

---

### 4.2 Integration/API Tests

**Endpoint:** `POST /users/create_user`

Scenarios:

- `test_create_user_valid_payload` (200 OK)
- `test_create_user_missing_fields_returns_422` (422)

**Endpoint:** `POST /users/get_access_token`

Scenarios:

- `test_get_access_token_valid_credentials` (200 OK)
- `test_get_access_token_invalid_credentials_returns_404` (404)

**Endpoint:** `POST /images/get_upload_url`

Scenarios:

- `test_get_upload_url_authorized_flow` (200 OK)

**Endpoint:** `POST /nodes/create_node`

Scenarios:

- `test_create_node_valid_payload` (200 OK)
- `test_create_node_missing_fields_returns_422` (422)

## Endpoint: POST /nodelinks/create\_link

Scenarios:

- test\_create\_link\_valid\_payload (200 OK)
- test\_create\_link\_missing\_fields\_returns\_422 (422)

## Evidence:

```
• (swe) talha@talha:~/lectures/SWE/swe_project/backend$ pytest
===== test session starts =====
platform linux -- Python 3.11.14, pytest-9.0.2, pluggy-1.6.0 -- /home/talha/miniforge3/envs/swe/bin/python3.11
cachedir: .pytest_cache
rootdir: /home/talha/lectures/SWE/swe_project/backend
configfile: pytest.ini
plugins: anyio-4.12.0, asyncio-1.3.0
asyncio: mode=Mode.STRICT, debug=False, asyncio_default_fixture_loop_scope=None, asyncio_default_test_loop_scope=funct
ion
collected 19 items

tests/test_routers.py::test_create_user_valid_payload PASSED [ 5%]
tests/test_routers.py::test_create_user_missing_fields_returns_422 PASSED [ 10%]
tests/test_routers.py::test_get_access_token_valid_credentials PASSED [ 15%]
tests/test_routers.py::test_get_access_token_invalid_credentials_returns_404 PASSED [ 21%]
tests/test_routers.py::test_get_upload_url_authorized_flow PASSED [ 26%]
tests/test_routers.py::test_create_node_valid_payload PASSED [ 31%]
tests/test_routers.py::test_create_node_missing_fields_returns_422 PASSED [ 36%]
tests/test_routers.py::test_create_link_valid_payload PASSED [ 42%]
tests/test_routers.py::test_create_link_missing_fields_returns_422 PASSED [ 47%]
tests/test_services.py::test_create_user_success PASSED [ 52%]
tests/test_services.py::test_create_user_supabase_error PASSED [ 57%]
tests/test_services.py::test_login_user_valid_password PASSED [ 63%]
tests/test_services.py::test_login_user_invalid_password PASSED [ 68%]
tests/test_services.py::test_get_upload_url_success PASSED [ 73%]
tests/test_services.py::test_get_upload_url_no_response PASSED [ 78%]
tests/test_services.py::test_update_node_success PASSED [ 84%]
tests/test_services.py::test_update_node_supabase_error PASSED [ 89%]
tests/test_services.py::test_delete_link_success PASSED [ 94%]
tests/test_services.py::test_delete_link_supabase_error PASSED [100%]

===== 19 passed in 0.05s =====
```

---

## 4.3 Performance Test

Tool: K6

Scenario: `read_node_info` @ 50 iterations/sec

Result: p95 ≈ 5.55s (threshold p95 < 500ms failed)

---

## 5. Test Execution Report

### 5.1 Automated Test Logs

```
• (swe) talha@talha:~/lectures/SWE/swe_project/backend$ pytest
===== test session starts =====
platform linux -- Python 3.11.14, pytest-9.0.2, pluggy-1.6.0 -- /home/talha/miniforge3/envs/swe/bin/python3.11
cachedir: .pytest_cache
rootdir: /home/talha/lectures/SWE/swe_project/backend
configfile: pytest.ini
plugins: anyio-4.12.0, asyncio-1.3.0
asyncio: mode=Mode.STRICT, debug=False, asyncio_default_fixture_loop_scope=None, asyncio_default_test_loop_scope=funct
ion
collected 19 items

tests/test_routers.py::test_create_user_valid_payload PASSED [ 5%]
tests/test_routers.py::test_create_user_missing_fields_returns_422 PASSED [ 10%]
tests/test_routers.py::test_get_access_token_valid_credentials PASSED [ 15%]
tests/test_routers.py::test_get_access_token_invalid_credentials_returns_404 PASSED [ 21%]
tests/test_routers.py::test_get_upload_url_authorized_flow PASSED [ 26%]
tests/test_routers.py::test_create_node_valid_payload PASSED [ 31%]
tests/test_routers.py::test_create_node_missing_fields_returns_422 PASSED [ 36%]
tests/test_routers.py::test_create_link_valid_payload PASSED [ 42%]
tests/test_routers.py::test_create_link_missing_fields_returns_422 PASSED [ 47%]
tests/test_services.py::test_create_user_success PASSED [ 52%]
tests/test_services.py::test_create_user_supabase_error PASSED [ 57%]
tests/test_services.py::test_login_user_valid_password PASSED [ 63%]
tests/test_services.py::test_login_user_invalid_password PASSED [ 68%]
tests/test_services.py::test_get_upload_url_success PASSED [ 73%]
tests/test_services.py::test_get_upload_url_no_response PASSED [ 78%]
tests/test_services.py::test_update_node_success PASSED [ 84%]
tests/test_services.py::test_update_node_supabase_error PASSED [ 89%]
tests/test_services.py::test_delete_link_success PASSED [ 94%]
tests/test_services.py::test_delete_link_supabase_error PASSED [100%]

===== 19 passed in 0.05s =====
```

### 5.2 Performance Graphs / Logs

#### THRESHOLDS

```
http_req_duration
x 'p(95)<500' p(95)=5.55s
```

```
http_req_failed
✓ 'rate<0.01' rate=0.00%
```

#### TOTAL RESULTS

```
checks_total.....: 1213    9.721816/s
checks_succeeded....: 100.00% 1213 out of 1213
checks_failed.....: 0.00%   0 out of 1213
```

```
✓ status is 200
```

#### HTTP

```
http_req_duration.....: avg=4.93s min=270.82ms med=4.92s max=23.67s p(90)=5.42s p(95)=5.55s
{ expected_response:true }...: avg=4.93s min=270.82ms med=4.92s max=23.67s p(90)=5.42s p(95)=5.55s
http_req_failed.....: 0.00%   0 out of 1213
http_reqs.....: 1213    9.721816/s
```

#### EXECUTION

```
dropped_iterations.....: 4787    38.366309/s
iteration_duration.....: avg=4.93s min=271.87ms med=4.92s max=23.67s p(90)=5.42s p(95)=5.55s
iterations.....: 1213    9.721816/s
vus.....: 10    min=10    max=50
vus_max.....: 50    min=33    max=50
```

#### NETWORK

```
data_received.....: 432 kB 3.5 kB/s
data_sent.....: 389 kB 3.1 kB/s
```



### 5.3 Defect Report

Bug ID	Description	Status	Severity
BUG-01	JWT token expired mid-run, causing 401 failures	Mitigated	Major
BUG-02	Endpoint mismatch ( <code>read_node_info</code> vs <code>get_node_info</code> )	Fixed	Medium
BUG-03	Request schema mismatch (query vs JSON body)	Resolved	Medium
BUG-04	<code>_wrap_node_op</code> missing self parameter	Fixed	High
BUG-05	p95 latency 5.5s at load (SLA fail)	Known limitation	Medium
BUG-06	K6 summary write permission denied	Minor	Low