**🎯 Minimal Testing Plan - Bare Minimum for API Functionality**

**📋 Goal: Verify APIs Work (Not Perfect Coverage)**

Test only the **core functionality** to ensure each API endpoint actually works.

**🎯 Priority 1: Critical Path Tests Only**

**userControllers.js (3 tests total)**

**1. registerUser**

test('should successfully create a new user')

* ✅ Mock User.create() to succeed
* ✅ Verify user is created with correct data
* ✅ Verify response is returned

**2. updateUser**

test('should successfully update user data')

* ✅ Mock User.findOneAndUpdate() to succeed
* ✅ Verify update operation is called
* ✅ Verify response is returned

**3. getUserById**

test('should successfully retrieve user by ID')

* ✅ Mock User.findOne() to return user data
* ✅ Verify user data is returned in response

**shippingAddressControllers.js (3 tests total)**

**1. createShippingAddress**

test('should successfully create shipping address')

* ✅ Mock User.findOneAndUpdate() with $push to succeed
* ✅ Verify address is added to user's array
* ✅ Verify response is returned

**2. updateShippingAddress**

test('should successfully update shipping address')

* ✅ Mock User.updateOne() with positional operator to succeed
* ✅ Verify address is updated
* ✅ Verify response is returned

**3. getShippingAddressById**

test('should successfully retrieve shipping address')

* ✅ Mock User.aggregate() to return address data
* ✅ Verify address data is returned

**defaultController.js (1 test)**

**1. defaultController**

test('should return Hello World message')

* ✅ Verify res.send('Hello World') is called

**urlNotFoundController.js (1 test)**

**1. urlNotFoundController**

test('should create and pass error to next middleware')

* ✅ Verify ResourceNotFoundError is created
* ✅ Verify next(error) is called

**📊 Total: 8 Tests Only**

| **Controller** | **Functions Tested** | **Tests** |
| --- | --- | --- |
| userControllers | 3 of 5 | 3 tests |
| shippingAddressControllers | 3 of 5 | 3 tests |
| defaultController | 1 of 1 | 1 test |
| urlNotFoundController | 1 of 1 | 1 test |
| **TOTAL** | **8 functions** | **8 tests** |

**🎯 What Each Test Verifies**

**Database Operations Work**

* ✅ The correct MongoDB method is called
* ✅ The method succeeds (doesn't throw errors)
* ✅ A response is returned

**API Endpoints Respond**

* ✅ Functions execute without crashing
* ✅ HTTP responses are sent
* ✅ Basic data flow works

**❌ What's NOT Tested (Intentionally)**

* ❌ Error handling scenarios
* ❌ Data validation edge cases
* ❌ Complex business logic
* ❌ Security vulnerabilities
* ❌ Performance issues
* ❌ Detailed data transformation
* ❌ Multiple scenarios per function

**🚀 Implementation Time**

* **Setup**: ~30 minutes (mocks, test structure)
* **Writing Tests**: ~2 hours (8 simple tests)
* **Total**: ~2.5 hours

**💡 Test Structure Template**

describe('ControllerName', () => {

test('should successfully [main action]', async () => {

// 1. Mock dependencies

User.create.mockResolvedValue(mockData);

// 2. Call function

await controllerFunction(req, res, next);

// 3. Verify database was called

expect(User.create).toHaveBeenCalled();

// 4. Verify response was sent

expect(res.json).toHaveBeenCalled();

});

});

**✅ Success Criteria**

After implementing these 8 tests:

* ✅ All APIs can be called without crashing
* ✅ Database operations are mocked and work
* ✅ Responses are returned
* ✅ Basic functionality is verified

**This is the absolute minimum to ensure your APIs are "working"!** 🎯