**🎯 Unit Test Types Summary for All Controllers**

Based on the controller functions I analyzed, here are the **common test types** applicable across all controllers:

**📋 1. Database Operation Tests**

**CRUD Operations**

* ✅ **CREATE**: User.create() - Insert new documents
* ✅ **READ**: User.findOne(), User.find(), User.aggregate() - Query documents
* ✅ **UPDATE**: User.findOneAndUpdate(), User.updateOne() - Modify documents
* ✅ **DELETE**: User.updateOne() with $pull - Remove subdocuments

**Database Test Pattern**

// Verify correct database method is called

expect(User.create).toHaveBeenCalledTimes(1);

expect(User.create).toHaveBeenCalledWith(expectedData);

**📋 2. Data Transformation Tests**

**ID Generation**

* ✅ Generate unique IDs using createRandomString()
* ✅ Assign IDs to new documents/subdocuments

**Password Security**

* ✅ Hash passwords with argon2.hash()
* ✅ Never store plain-text passwords

**Timestamps**

* ✅ Set creation timestamps using getCurrentDateTime()

**Object Manipulation**

* ✅ Create filters from request data
* ✅ Remove specific fields (e.g., user\_id from update objects)
* ✅ Build update objects with correct structure

**📋 3. Error Handling Tests**

**Database Errors**

* ✅ Connection failures
* ✅ Operation failures
* ✅ Validation errors

**Business Logic Errors**

* ✅ Resource not found (ResourceNotFoundError)
* ✅ Empty request body (EmptyRequestBodyError)
* ✅ Invalid user IDs
* ✅ Missing subdocuments

**Error Propagation**

* ✅ Errors passed to next() function
* ✅ Custom error types with correct status codes

**📋 4. Response Handling Tests**

**HTTP Status Codes**

* ✅ 200 - Successful GET operations
* ✅ 201 - Successful POST/PATCH operations
* ✅ 400 - Bad request errors
* ✅ 404 - Resource not found errors

**Response Format**

* ✅ JSON response structure
* ✅ Success responses with data
* ✅ Error responses with error details

**📋 5. Validation Tests**

**Input Validation**

* ✅ Required field presence
* ✅ Data type validation
* ✅ Format validation (email, phone, etc.)
* ✅ Length constraints

**Business Rule Validation**

* ✅ User existence checks
* ✅ Resource existence checks
* ✅ Permission validations

**📋 6. Subdocument Operations Tests**

**Array Operations**

* ✅ $push - Add subdocuments to arrays
* ✅ $pull - Remove subdocuments from arrays
* ✅ $set with positional operator $ - Update specific array elements

**Aggregation Pipelines**

* ✅ $match - Filter documents
* ✅ $unwind - Expand arrays
* ✅ $project - Select specific fields

**📋 7. Async Error Handling Tests**

**asyncErrorHandler Wrapper**

* ✅ Catch async function errors
* ✅ Pass errors to Express error handler
* ✅ Prevent unhandled promise rejections

**📋 8. Request/Response Object Tests**

**Request Handling**

* ✅ Extract data from req.body
* ✅ Extract parameters from req.params
* ✅ Handle different request types

**Response Methods**

* ✅ res.status().json() - Set status and send JSON
* ✅ res.send() - Send simple responses
* ✅ next() - Pass control to next middleware

**📋 9. Mock Dependencies Tests**

**External Libraries**

* ✅ Mock argon2 for password hashing
* ✅ Mock lodash for object manipulation
* ✅ Mock createRandomString for ID generation
* ✅ Mock getCurrentDateTime for timestamps

**Database Models**

* ✅ Mock User model operations
* ✅ Mock support functions (e.g., checkUserExists)

**📋 10. Edge Cases Tests**

**Empty Data**

* ✅ Empty request bodies
* ✅ Null/undefined values
* ✅ Empty arrays

**Boundary Conditions**

* ✅ Minimum valid values
* ✅ Maximum valid values
* ✅ Exact boundary values

**🎯 Universal Test Structure**

Every controller test follows this pattern:

describe('Controller Function', () => {

// Setup mocks

beforeEach(() => {

jest.clearAllMocks();

// Setup req, res, next

});

// Core functionality

test('should perform main database operation');

// Data transformation

test('should transform request data correctly');

// Error handling

test('should handle database failures');

// Response handling

test('should return correct response format');

});

**📊 Test Distribution**

| **Test Type** | **Applicable Controllers** | **Priority** |
| --- | --- | --- |
| Database Operations | All controllers | Critical |
| Data Transformation | Most controllers | High |
| Error Handling | All async controllers | Critical |
| Response Handling | All controllers | High |
| Validation | All controllers | High |
| Subdocument Operations | shippingAddressControllers | Medium |
| Async Error Handling | All async controllers | High |
| Request/Response | All controllers | Medium |
| Mock Dependencies | All controllers | High |
| Edge Cases | All controllers | Low |

These **10 test types** cover all the functionality across your controllers and ensure comprehensive coverage! 🚀