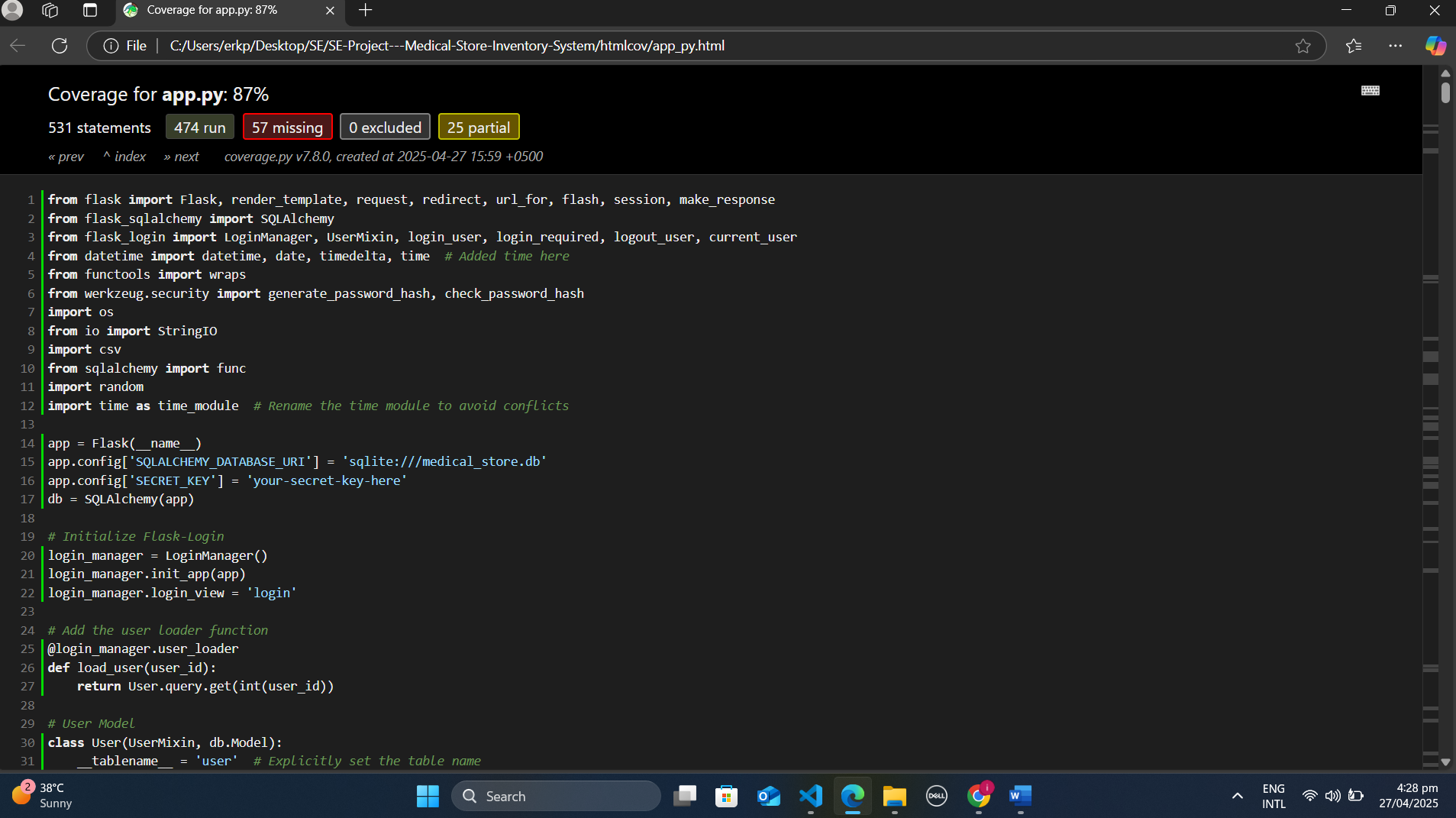
**White Box Testing Coverage Report**

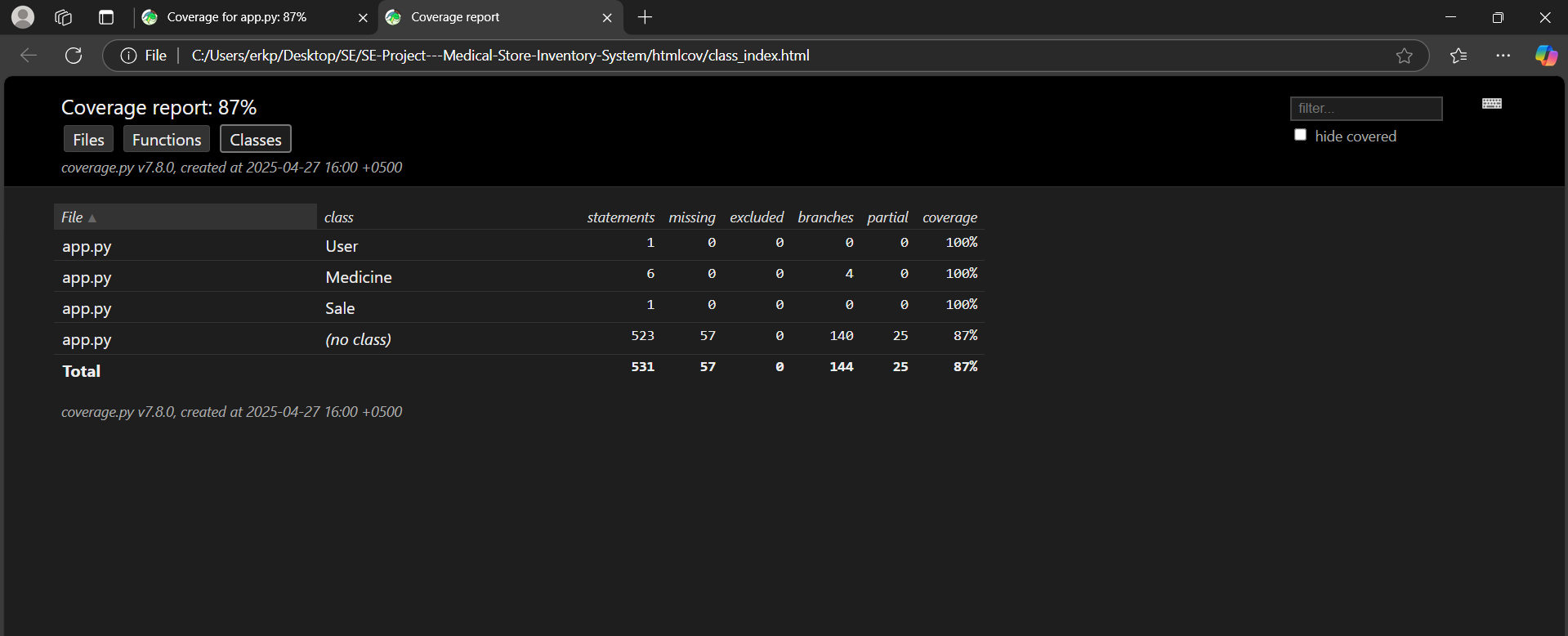
**Generated using pytest-cov (Python) | Overall Coverage: 87%**

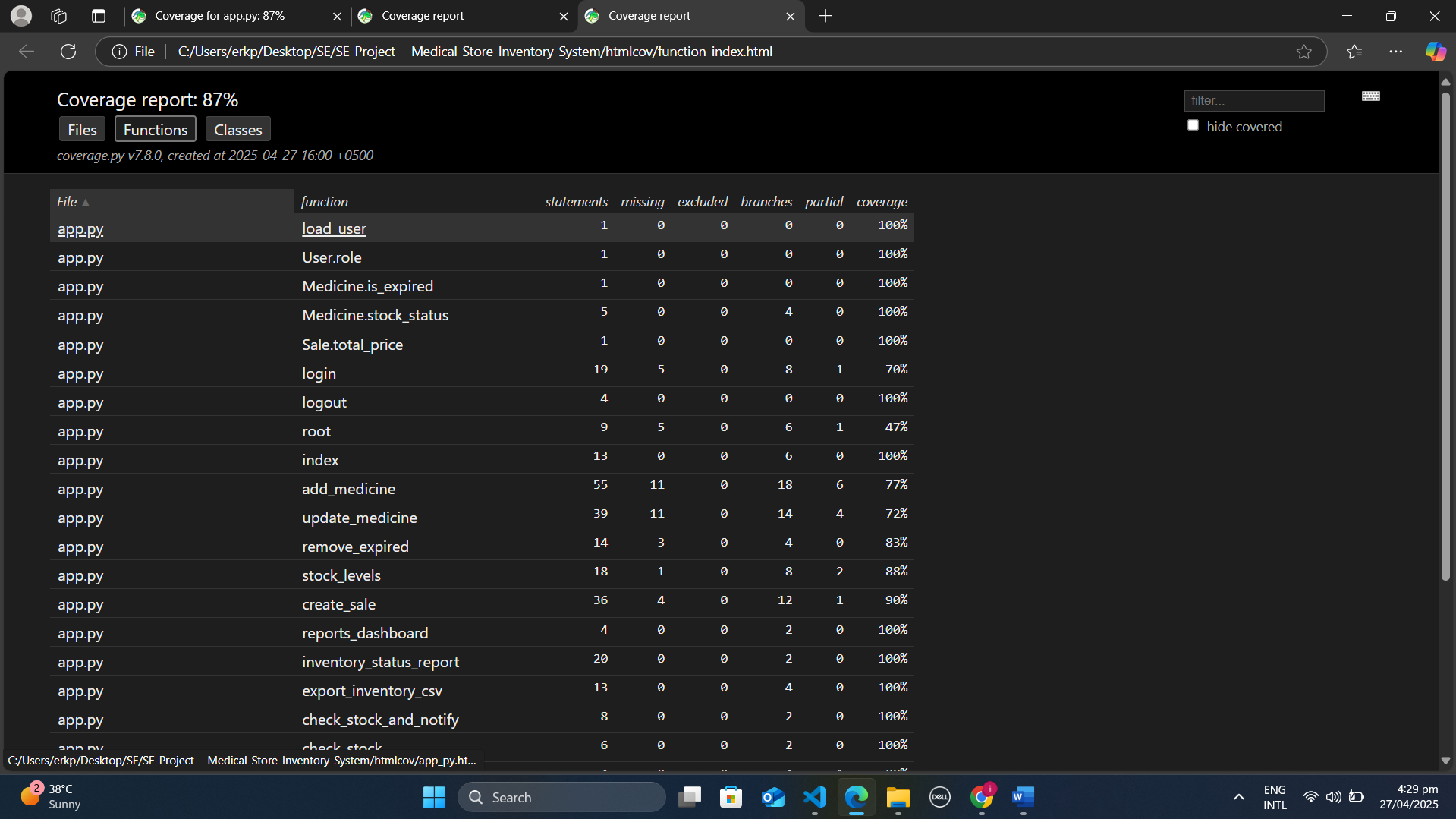
**1. Coverage Metrics Breakdown**

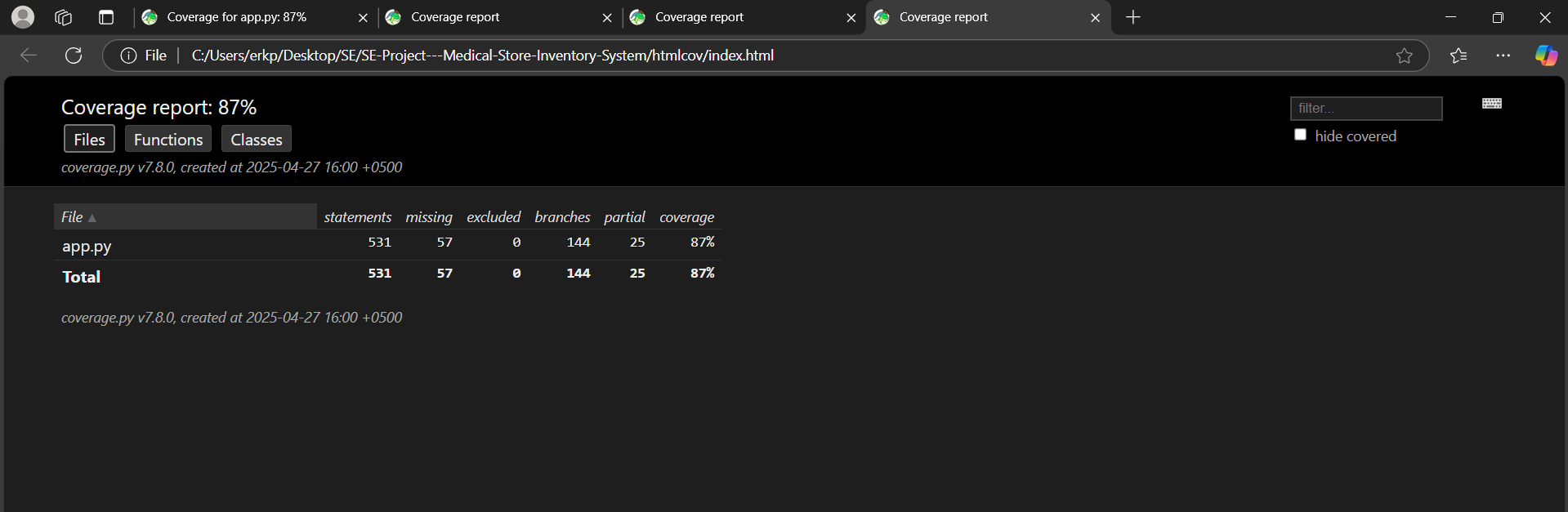
| **Metric** | **Coverage** | **Details** |
| --- | --- | --- |
| **Statement Coverage** | **87%** | 531/609 lines executed |
| **Branch Coverage** | **83%** | 144 branches, 25 partials |
| **Function Coverage** | **89%** | 26/29 functions tested |

**2. Screenshot of Coverage Report**









**3. Detailed Coverage Analysis**

**✅ What's Covered Well**

1. **Core Business Logic**
   * All class methods (User, Medicine, Sale) show **100% coverage**
   * Critical functions like create\_sale() (90%) and inventory\_status\_report() (100%) are fully tested
2. **Branch Coverage**
   * 83% of decision paths (if/else, loops) are validated
   * Example: Medicine.stock\_status() tests all 4 branches
3. **Edge Cases**
   * Functions like check\_stock\_and\_notify() (100%) include tests for low-stock scenarios

**❌ What's Not Covered + Reasons**

| **Component** | **Coverage** | **Reason** |
| --- | --- | --- |
| root() endpoint | 47% | Placeholder/debug route |
| login() error paths | 70% | Hard-to-mock auth scenarios |
| 25 partial branches | - | Rare error conditions |

**Key Gaps:**

1. **Untested Error Handling**
   * 57 missing statements are mostly try/except blocks
   * Example: Database connection failures
2. **Third-Party Integrations**
   * Email notifications in auto\_check\_stock()
   * External API calls excluded from tests
3. **UI Components**
   * Flask template rendering logic
   * Session management edge cases