

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<testsuites>
  <testsuite name="pytest" errors="2" failures="3" skipped="0" tests="9" time="0.831" timestamp="2024-12-31T20:00:49.038949+08:00"
    hostname="LAPTOP-QQOUPES4">
    <testcase classname="test_authentication" name="test_account_lockout" file="test_authentication.py" line="40" time="0.021">
      <system-out>----- Captured Log ----- WARNING root:authentication_controller.py:66
      lockuser account locked due to multiple failed attempts. ----- Captured Out -----
      Invalid username or password. You have 2 attempt(s) remaining. Invalid username or password. You have 1 attempt(s) remaining. Account
      locked due to multiple failed attempts. Please try again after 5 minutes. Account is locked. Please try again after 4 minute(s) and 59
      second(s). User 'lockuser' account has been unlocked. </system-out>
      <system-err>----- Captured Err ----- </system-err>
    </testcase>
    <testcase classname="test_authentication" name="test_invalid_login" file="test_authentication.py" line="36" time="0.017">
      <system-out>----- Captured Log ----- ----- Captured
      Out ----- Invalid username or password. </system-out>
      <system-err>----- Captured Err ----- </system-err>
    </testcase>
    <testcase classname="test_authentication" name="test_valid_login" file="test_authentication.py" line="23" time="0.005">
      <failure message="OSError: pytest: reading from stdin while output is captured! Consider using `s`.">auth_controller =
      <controllers.authentication_controller.AuthenticationController object at 0x000001A126D869D0> def test_valid_login(auth_controller):
      user_service = auth_controller.user_service username = 'testuser' password_plain = 'Test@1234' # Ensure the test user exists if not
      user_service.get_user_by_username(username): # Encrypt the password encrypted_password = encrypt_password(password_plain)
      user_service.add_user(Customer(username, encrypted_password, 'COMPANY123')) > user = auth_controller.login(username, password_plain)
      test_authentication.py:33:
      controllers\authentication_controller.py:51: in login if self.perform_mfa(user): controllers\authentication_controller.py:80: in
      perform_mfa input_code = input("Enter the MFA code sent to your registered device: ")
      self = <_pytest.capture.DontReadFromInput object at 0x000001A125149340> size = -1 def read(self, size: int =
      -1) -> str: > raise OSError( "pytest: reading from stdin while output is captured! Consider using `s`." ) E OSError: pytest: reading from
      stdin while output is captured! Consider using `s`. C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\lib\site-
      packages\_pytest\capture.py:209: OSError</failure>
      <system-out>----- Captured Log ----- ----- Captured
      Out ----- Enter the MFA code sent to your registered device: </system-out>
      <system-err>----- Captured Err ----- </system-err>
    </testcase>
    <testcase classname="test_order" name="test_list_customer_orders" file="test_order.py" line="53" time="0.016">
      <error message="failed on setup with "py.error.EEXIST: [File exists]: mkdir('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen
      Siu\\pytest-5\\test_list_customer_orders0\\data',)">E FileExistsError: [WinError 183] Cannot create a file when that file already exists:
      'C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-5\\test_list_customer_orders0\\data' All traceback entries are
      hidden. Pass `--full-trace` to see hidden and internal frames. During handling of the above exception, another exception occurred: tmpdir
      = local('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-5\\test_list_customer_orders0') @pytest.fixture def
      product_service(tmpdir): > data_dir = tmpdir.mkdir("data") test_order.py:27:
      self = local('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-5\\test_list_customer_orders0')
      args = ('data',) p = local('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-5\\test_list_customer_orders0\\data')
      def mkdir(self, *args): """Create & return the directory joined with args.""" p = self.join(*args) > error.checked_call(os.mkdir,
      os.fspath(p)) E py.error.EEXIST: [File exists]: mkdir('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-
```

```

5\\test_list_customer_orders0\\data',) C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\lib\site-
packages\_pytest\_py\path.py:889: EEXIST</error>
<system-out>----- Captured Log ----- Captured
Out ----- </system-out>
<system-err>----- Captured Err ----- </system-err>
</testcase>
<testcase classname="test_order" name="test_place_order" file="test_order.py" line="38" time="0.009">
  <error message="failed on setup with "py.error.EEXIST: [File exists]: mkdir('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen
  Siu\\pytest-5\\test_place_order0\\data',)">E FileExistsError: [WinError 183] Cannot create a file when that file already exists:
  'C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-5\\test_place_order0\\data' All traceback entries are hidden.
  Pass `--full-trace` to see hidden and internal frames. During handling of the above exception, another exception occurred: tmpdir =
  local('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-5\\test_place_order0') @pytest.fixture def
  product_service(tmpdir): > data_dir = tmpdir.mkdir("data") test_order.py:27: _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
  _ _ _ _ _ _ _ _ _ _ self = local('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-5\\test_place_order0') args =
  ('data',) p = local('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-5\\test_place_order0\\data') def mkdir(self,
  *args): """Create & return the directory joined with args.""" p = self.join(*args) > error.checked_call(os.mkdir, os.fspath(p)) E
  py.error.EEXIST: [File exists]: mkdir('C:\\Users\\Helen Siu\\AppData\\Local\\Temp\\pytest-of-Helen Siu\\pytest-
  5\\test_place_order0\\data',) C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\lib\site-packages\_pytest\_py\path.py:889:
  EEXIST</error>
  <system-out>----- Captured Log ----- Captured
  Out ----- </system-out>
  <system-err>----- Captured Err ----- </system-err>
</testcase>
<testcase classname="test_product" name="test_add_product" file="test_product.py" line="24" time="0.008">
  <failure message="assert False is True">product_service = <services.product_service.ProductService object at 0x000001A126D86CA0> def
  test_add_product(product_service): product = Product('1', "Shampoo", "Hair Care", 9.99, 100) added = product_service.add_product(product)
  > assert added is True E assert False is True test_product.py:28: AssertionError</failure>
  <system-out>----- Captured Log ----- Captured
  Out ----- Product with ID '1' already exists. </system-out>
  <system-err>----- Captured Err ----- </system-err>
</testcase>
<testcase classname="test_product" name="test_delete_product" file="test_product.py" line="37" time="0.011">
  <system-out>----- Captured Log ----- Captured
  Out ----- </system-out>
  <system-err>----- Captured Err ----- </system-err>
</testcase>
<testcase classname="test_product" name="test_list_products" file="test_product.py" line="43" time="0.007">
  <failure message="assert 4 == 1 + where 4 = len([<models.product.Product object at 0x000001A127A97D90>, <models.product.Product object at
  0x000001A127A97F10>, <models.product.Product object at 0x000001A127AABB80>, <models.product.Product object at
  0x000001A127AABEB0>])">product_service = <services.product_service.ProductService object at 0x000001A127A97490> def
  test_list_products(product_service): product_service.add_product(Product('4', "Hair Spray", "Styling", 6.99, 80)) products =
  product_service.products > assert len(products) == 1 E assert 4 == 1 E + where 4 = len([<models.product.Product object at
  0x000001A127A97D90>, <models.product.Product object at 0x000001A127A97F10>, <models.product.Product object at 0x000001A127AABB80>,
  <models.product.Product object at 0x000001A127AABEB0>]) test_product.py:47: AssertionError</failure>
  <system-out>----- Captured Log ----- Captured
  Out ----- Product with ID '4' already exists. </system-out>
  <system-err>----- Captured Err ----- </system-err>

```

```
</testcase>
<testcase classname="test_product" name="test_update_stock" file="test_product.py" line="30" time="0.008">
  <system-out>----- Captured Log ----- Captured
    Out ----- Product with ID '2' already exists. </system-out>
    <system-err>----- Captured Err ----- </system-err>
  </testcase>
</testsuite>
</testsuites>
```