# BOOK VERSE-UNIT TESTING

IT-314 Software Engineering
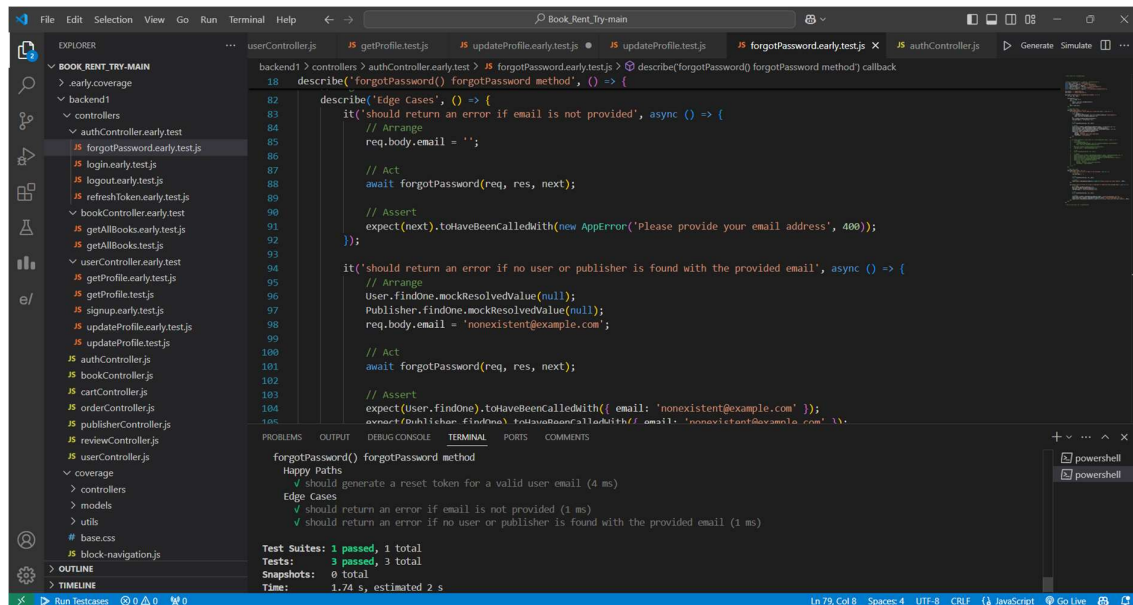
BOOK VERSE-G20

MENTOR:SAURABH TIWARI

Software Engineering

# Forgot password:
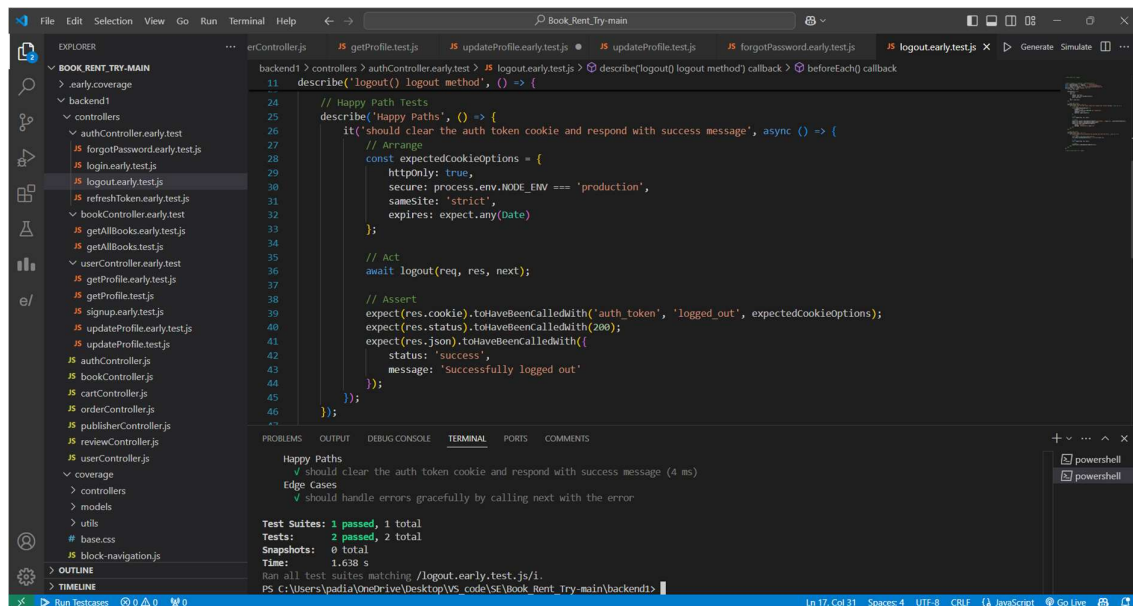
```
describe('forgotPassword() forgotPassword method', () => {
  describe('Edge Cases', () => {
    it('should return an error if email is not provided', async () => {
      // Arrange
      req.body.email = '';

      // Act
      await forgotPassword(req, res, next);

      // Assert
      expect(next).toHaveBeenCalledWith(new AppError('Please provide your email address', 400));
    });

    it('should return an error if no user or publisher is found with the provided email', async () => {
      // Arrange
      User.findOne.mockResolvedValue(null);
      Publisher.findOne.mockResolvedValue(null);
      req.body.email = 'nonexistent@example.com';

      // Act
      await forgotPassword(req, res, next);

      // Assert
      expect(User.findOne).toHaveBeenCalledWith({ email: 'nonexistent@example.com' });
      expect(Publisher.findOne).toHaveBeenCalledWith({ email: 'nonexistent@example.com' });
```

```
forgotPassword() forgotPassword method
  Happy Paths
    √ should generate a reset token for a valid user email (4 ms)
  Edge Cases
    √ should return an error if email is not provided (1 ms)
    √ should return an error if no user or publisher is found with the provided email (1 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        1.74 s, estimated 2 s
```
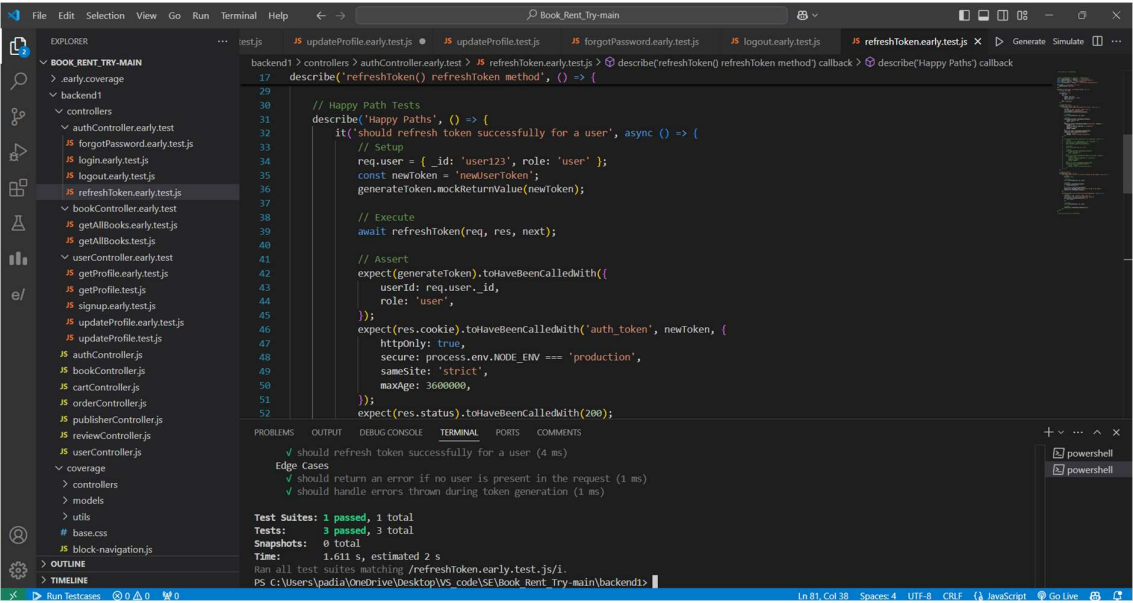
# Logout:

```
describe('logout() logout method', () => {
  // Happy Path Tests
  describe('Happy Paths', () => {
    it('should clear the auth token cookie and respond with success message', async () => {
      // Arrange
      const expectedCookieOptions = {
        httpOnly: true,
        secure: process.env.NODE_ENV === 'production',
        sameSite: 'strict',
        expires: expect.any(Date)
      };

      // Act
      await logout(req, res, next);

      // Assert
      expect(res.cookie).toHaveBeenCalledWith('auth_token', 'logged_out', expectedCookieOptions);
      expect(res.status).toHaveBeenCalledWith(200);
      expect(res.json).toHaveBeenCalledWith({
        status: 'success',
        message: 'Successfully logged out'
      });
    });
  });
});
```
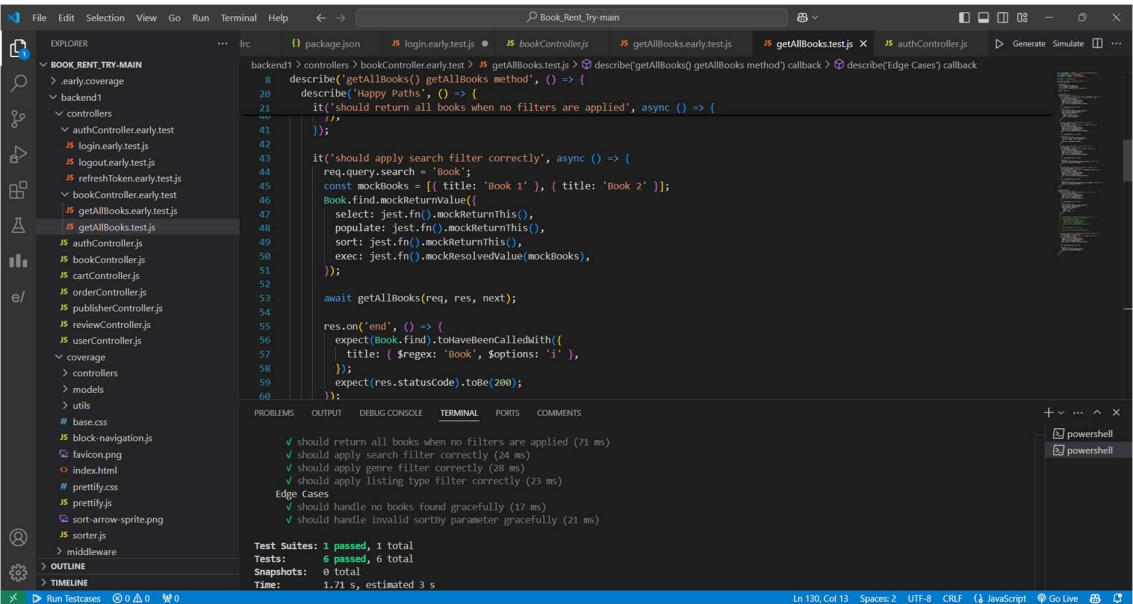
```
  Happy Paths
    √ should clear the auth token cookie and respond with success message (4 ms)
  Edge Cases
    √ should handle errors gracefully by calling next with the error

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.638 s
Ran all test suites matching /logout.early.test.js/i.
PS C:\Users\padia\OneDrive\Desktop\VS_code\SE\Book_Rent_Try-main\backend1>
```
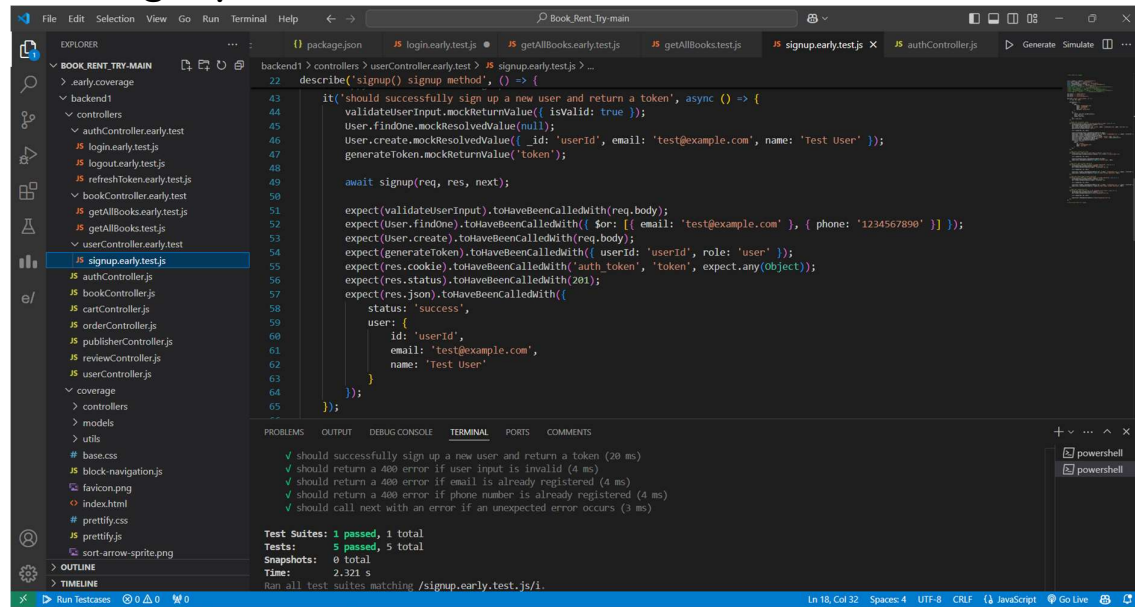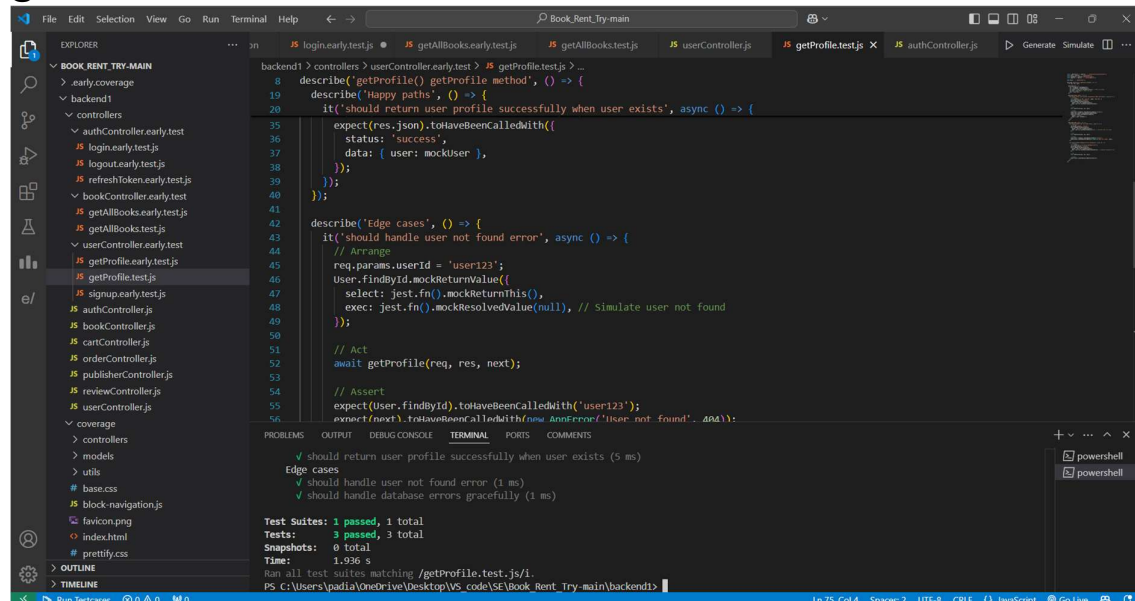
## RefreshTokens:



## GetAllBooks:

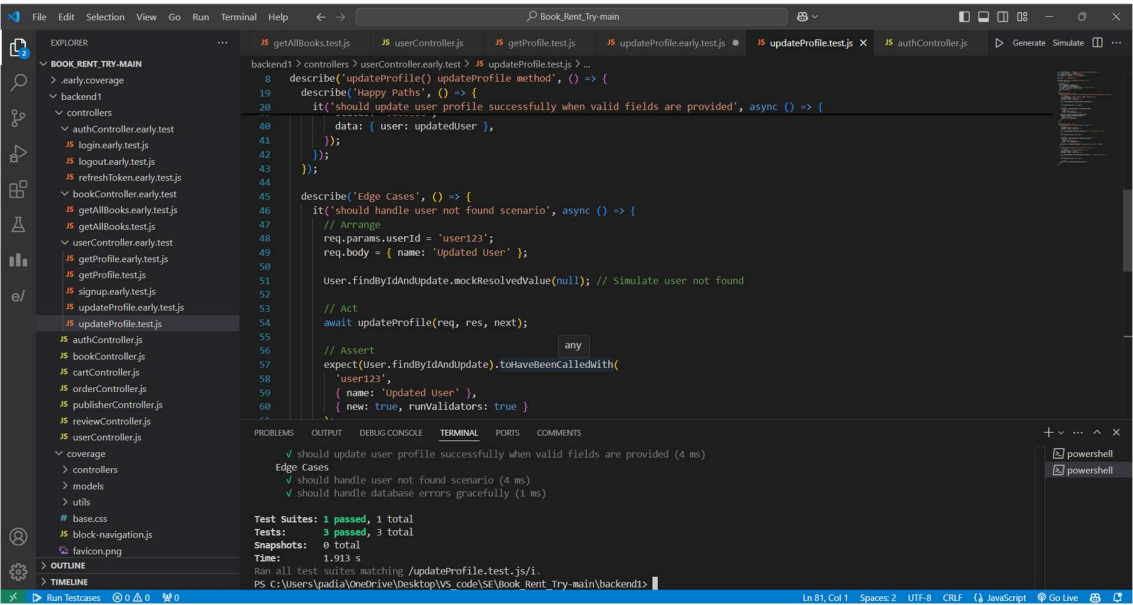## UserSignup:



```js
describe('signup() signup method', () => {
  it('should successfully sign up a new user and return a token', async () => {
    validateUserInput.mockReturnValue({ isValid: true });
    User.findOne.mockResolvedValue(null);
    User.create.mockResolvedValue({ _id: 'userId', email: 'test@example.com', name: 'Test User' });
    generateToken.mockReturnValue('token');

    await signup(req, res, next);

    expect(validateUserInput).toHaveBeenCalledWith(req.body);
    expect(User.findOne).toHaveBeenCalledWith({ $or: [{ email: 'test@example.com' }, { phone: '1234567890' }] });
    expect(User.create).toHaveBeenCalledWith(req.body);
    expect(generateToken).toHaveBeenCalledWith({ userId: 'userId', role: 'user' });
    expect(res.cookie).toHaveBeenCalledWith('auth_token', 'token', expect.any(Object));
    expect(res.status).toHaveBeenCalledWith(201);
    expect(res.json).toHaveBeenCalledWith({
      status: 'success',
      user: {
        id: 'userId',
        email: 'test@example.com',
        name: 'Test User'
      }
    });
  });
```

Terminal output:
```
√ should successfully sign up a new user and return a token (20 ms)
√ should return a 400 error if user input is invalid (4 ms)
√ should return a 400 error if email is already registered (4 ms)
√ should return a 400 error if phone number is already registered (4 ms)
√ should call next with an error if an unexpected error occurs (3 ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        2.321 s
Ran all test suites matching /signup.early.test.js/i.
```

## getProfile:



```js
describe('getProfile() getProfile method', () => {
  describe('Happy paths', () => {
    it('should return user profile successfully when user exists', async () => {
      expect(res.json).toHaveBeenCalledWith({
        status: 'success',
        data: { user: mockUser },
      });
    });
  });

  describe('Edge cases', () => {
    it('should handle user not found error', async () => {
      // Arrange
      req.params.userId = 'user123';
      User.findById.mockReturnValue({
        select: jest.fn().mockReturnThis(),
        exec: jest.fn().mockResolvedValue(null), // Simulate user not found
      });

      // Act
      await getProfile(req, res, next);

      // Assert
      expect(User.findById).toHaveBeenCalledWith('user123');
      expect(next).toHaveBeenCalledWith(new AppError('User not found', 404));
```

Terminal output:
```
√ should return user profile successfully when user exists (5 ms)
  Edge cases
    √ should handle user not found error (1 ms)
    √ should handle database errors gracefully (1 ms)

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        1.936 s
Ran all test suites matching /getProfile.test.js/i.
PS C:\Users\padia\OneDrive\Desktop\VS_code\SE\Book_Rent_Try-main\backend1>
```

# updateProfile:



# Coverage



## All files controllers

**62.87%** Statements 105/167  **43.83%** Branches 32/73  **53.84%** Functions 7/13  **62.87%** Lines 105/167

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

| File ▲ | Statements | | Branches | | Functions | | Lines | |
|--------|-----------|--------|----------|-------|-----------|------|--------|-------|
| authController.js | | 78.04% | 32/41 | 58.33% | 7/12 | 75% | 3/4 | 78.04% | 32/41 |
| bookController.js | | 57.81% | 37/64 | 40.54% | 15/37 | 33.33% | 1/3 | 57.81% | 37/64 |
| userController.js | | 58.06% | 36/62 | 41.66% | 10/24 | 50% | 3/6 | 58.06% | 36/62 |

Code coverage generated by istanbul at 2024-12-02T14:00:19.846Z