***TEST CASE***

***EZ***

Question1: What are test cases for the above  
Ans. Certainly! Here are rephrased test cases for the described file-sharing system:

### User 1: Operation User (Ops User)

1. \*\*Login:\*\*

- Positive Scenario: Successful login with valid credentials.

- Negative Scenario: Unsuccessful login with invalid credentials.

2. \*\*Upload File:\*\*

- Positive Scenario: Ops User uploads a valid pptx, docx, or xlsx file.

- Negative Scenarios:

- Attempt to upload an unsupported file format.

- Attempt to upload a file without proper authentication.

### User 2: Client User

3. \*\*Sign Up:\*\*

- Positive Scenario: Successful sign-up resulting in an encrypted URL.

- Negative Scenarios:

- Sign-up attempt with incomplete or invalid information.

- Sign-up attempt with an existing email.

4. \*\*Email Verify:\*\*

- Positive Scenario: Successful verification of the email through the provided link.

- Negative Scenario: Attempted verification with an incorrect or expired link.

5. \*\*Login:\*\*

- Positive Scenario: Successful login with valid credentials.

- Negative Scenario: Unsuccessful login with invalid credentials.

6. \*\*Download File:\*\*

- Positive Scenario: Successful download of a file using the provided encrypted URL.

- Negative Scenarios:

- Attempted download without proper authentication.

- Attempted download with an incorrect or expired URL.

7. \*\*List all uploaded files:\*\*

- Positive Scenario: Client User receives a list of accessible files.

- Negative Scenario: Attempted file listing without proper authentication.

8. \*\*Security Measures:\*\*

- Ops User attempting to access the download URL should be denied.

- Unauthorized attempts to access the download URL should result in denial.

Question2. How do you plan on deploying this to the production environment?

Ans. To deploy the file-sharing system to production:

1. Set up a secure server with a web server.

2. Choose and configure the database.

3. Package and deploy the application code.

4. Configure environment variables for security.

5. Implement HTTPS for secure data transmission.

6. Optionally, introduce load balancing for scalability.

7. Monitor system performance and set up logging.

8. Establish CI/CD for automated testing and deployment.

9. Implement backup, disaster recovery, and scaling strategies.

10. Maintain comprehensive documentation for reference.