**Risks**

1. **Environment Configuration Issues**
   * **Risk**: Problems with setting up the test environment, such as version incompatibilities and server configuration issues.
   * **Mitigation**: Ensure all tools and dependencies are correctly installed and up-to-date. Review environment setup documentation and perform initial tests to verify everything functions correctly.
2. **API or Application Errors**
   * **Risk**: The API or application might have errors or unexpected behaviors that were not detected during development.
   * **Mitigation**: Conduct thorough testing of all endpoints and functionalities. Document all encountered errors.
3. **Non-Representative Test Data**
   * **Risk**: Test data might not represent the actual data that will be handled in production.
   * **Mitigation**: Use a dataset that simulates real-use scenarios and perform tests with different data types to cover edge cases.
4. **Test Tool Limitations**
   * **Risk**: Issues with using test tools such as Postman or any other tools utilized.
   * **Mitigation**: Become familiar with the tools before starting the tests. Consult the documentation and seek help in forums or communities if problems arise.
5. **Accessibility and Usability Issues**
   * **Risk**: The user interface might not be accessible or usable for all users.
   * **Mitigation**: Conduct accessibility and usability tests to ensure the application meets accessibility standards and requirements.
6. **Time Constraints**
   * **Risk**: Lack of time to perform all necessary tests due to time restrictions.
   * **Mitigation**: Prioritize tests based on their importance and impact. Allocate sufficient time for critical tests and streamline testing processes where possible.