Test Plan Template

1. **Introduction**
   1. Test Plan Objective = The goal is to validate the delivery routing system, ensuring it accurately calculates the shortest delivery routes, adheres to truck capacity constraints, and effectively handles the package delivery instructions
2. **Scope**
   1. **Correct route mapping for blue, green and yellow trucks**
   2. **Accurate shortest path calculations, avoiding buildings**
   3. **Proper handling of delivery specifications , including weight , box size and destination**
3. **Test Strategy**
   1. **First of all in this project , we will be testing the exploratory phase i.e. we will first check the proper implementation instead of going deep into the accuracy part. Certain kinds of tests will be implemented to make sure the smooth flow of the code.**   
      3.1. System Test – This test make sure that entire delivery application operates according to specifications. Basically it will check if the system is working correctly.  
      3.2. Performance Test – This test will measure aplication’s efficiency , like how quickly it calculates routes and responds to user inputs under normal conditions.  
      3.3. Security Test – This will ensure application respinds properly to invalid or corrupt input datas. Like it should reject the trash data and should function properly rather than getting crashed.  
      3.4. Automated Test – This includes unit tests for individual fucntions and integration tests for overall route calculation process. Basically , core functionalities will be validated.  
      3.5. Stress and Volume Test – Stress basically means stress i.e. pushing the system to its limitby giving or entering more and more requests than usual to check if it can handle high loads.Volume also means handling large number of inputs.  
      3.6. Recovery Test – This basically means application’s ability to recover from potential loses such as while getting invalid inputs or load exceeding of the trucks. That is all what will be happening in recovery test.  
      3.7. Documentation Test – Documentations test means accurate completeness of the system or we can say complete code. This includes adding proper comments , readme file and system documentation.  
      3.8. Beta Test – Releasing outside the circle of working devs to beta users. Not happening here  
      3.9. User Acceptance Test – Testing with users to check it is great fit and it works perfect in live conditins.
   2. **You could describe the test design process and give an overview of how it will be conducted. You could provide a broad overview of** 
      1. **how to understand requirements, - Carefully read project decoumentation i.e milestone pdf. Identify functional requirements like route mapping, shortest path calculation, truck capacity and logic for package delivery.**
      2. **build a traceability matrix, - Traceability matrix template has been provided , from the template I can describe in little brief that (R001-R007) corresponds to business requirements and (T001-T015) should refer to test cases and them accordingly**
      3. **prepare test cases, - Verifying starting and ending points for the trucks . Executing shortest path details**
      4. **and have them reviewed - Peer review sessions should be organized . They will present test cases to each other. We can even use a checklist to ensure the consistency.**
4. **Environment Requirements**
   1. **Hardware requirements = Developer system - CPU: Minimum Quad-core Processor 2.0 GHz, Ram – 8gb min, Storage – 256 SSD min**
   2. **Software Requirements = Windows 10 pro or 11, IDE Code: Blocks with GCC compiler , Git for version control with repositories .**
   3. **Testing tools = Issue tracking = JIRA, Unit testing = Visual studio or vs code**
   4. **Communication = WhatsApp, Teams , Outlook**
5. **Execution Strategy**
   1. Entry and Exit criteria = Entry – The solution should be built properly, and each file should be on its place and stable. All code must be reviewed. Exit – All test cases should be passed with 95% or higher for non-critical tests and for main vulnerable tests, the rate should be 100%.

**Critical defects =** Incorrect route calculation or crash.

**High severity =** Functioning, but not properly

**Medium severity =** Bugs in non-critical features

**Low severity =** Minor issues

**Cosmetic =** Visual defects

* 1. **Test Reporting =** Test reports will be generated daily and after each significant testing cycle. These reports will include metrics on the number of tests run, passed, failed, blocked, and details on new, open, and closed defects.
  2. **Feedback =** Testers will provide feedback through defect tracking system known as JIRA. With regular over the call or MS team meetings, the bugs will be identified.

1. **Test Schedule**
   1. **Planning :** The test schedule will be divided into phases, such as unit testing, integration testing, system testing, and user acceptance testing
   2. **Timeline : Tetsing to be completed before 18 MARCH**
2. **Control Procedures**
   1. 6.1 Reviews = Weekly review meetings to assess progress, with special attention to high-risk areas.  
      6.2 Bug Review Meetings = 10-15 minute meetings evry two days to review and fix bugs  
      6.3 Change Request = Through mode of communication upon everyone and professor’s permission  
      6.4 Defect Reporting = Throgh JIRA
3. **Functions To Be Tested**

**= All functions related to map population, route addition, pathfinding, and UI interactions.**

1. **Resources and Responsibilities**  
   8.1. Resources = Allocation of hardware , software accoridngly  
   8.2. Responsibilities = Clear task division
2. **Deliverables = All code files , header , .c and test cases , scripts and reports.**
3. **Suspension / Exit Criteria = Suspension on critical defect and exit on 95% on non-criticial tests and 100% pass rate on critical tests.**
4. **Resumption Criteria = Defect solving and achieving passing rate leads to resumption**
5. **Dependencies**  
   12.1 Personnel Dependencies = Specific team member roles as already assigned  
   12.2 Software Dependencies = On librarires , services etcetera  
   12.3 Hardware Dependencies = On the machine which code is being run on  
   12.3 Test Data & Database = Test data availability and database integration
6. **Risks**  
   13.1. Schedule = Delays casing project delay  
   13.2. Technical = Technical challneges such as software bug  
   13.3. Management = Resoruce allocation , contact following  
   13.4. Personnel = Availability  
   13.5 Requirements = Changes in requirements
7. **Tools** 
   1. **Testing tools = Tools for managing and executing tests**
   2. **Development tools = IDE’s , Version control systems and all**
8. **Documentation** 
   1. **Test strategy document**
   2. **Test Plan**
   3. **Test case specs**
9. **Approvals**

**= from Team lead, Project Manager and other team members.**