



Individual Project Phase 1

Course: PROG2500-26W-Sec1-Open-Source Full Stack Development

Deployment & Integrity Checks	Excellent (9-10 points) 10 points	Good (5-8 points) 8 points	Needs Improvement (1-4 points) 4 points	Unsatisfactory (0 points) 0 points	Criterion Score
Deployment & Integrity Checks	(9-10 points)	(5-8 points)	(1-4 points)	(0 points)	/ 10

Project is deployed to a live public URL (Render/Heroku/etc.) and accessible. GitHub repository shows a healthy history of regular commits throughout the development phase.

Project is deployed and accessible. Git history is present but commits are sporadic or lack descriptive messages.

Project is deployed but unstable/slow. Git history is suspicious (e.g., only 1-2 massive commits).

Project is NOT deployed (localhost only), OR submitted as a Zip file, OR Git history is missing/empty. **Assignment may receive a total grade of 0.**

Data Modeling & Entity Relationships	Excellent (21-25 points) 25 points	Good (14-20 points) 20 points	Needs Improvement (6-13 points) 13 points	Unsatisfactory (0-5 points) 0 points	Criterion Score
--------------------------------------	---------------------------------------	----------------------------------	--	---	-----------------

Data Modeling & Entity Relationships	Excellent (21-25 points) 25 points	Good (14-20 points) 20 points	Needs Improvement (6-13 points) 13 points	Unsatisfactory (0-5 points) 0 points	Criterion Score
Data Modeling & Entity Relationships	(21-25 points) Schema is efficient and logical. The "Rule of 3" is fully implemented. Data is organized so that no information is duplicated across tables (e.g., user address isn't repeated in every order). Foreign Keys are correctly used to link tables.	(14-20 points) 3 Tables exist and connect, but some data redundancy exists (e.g., repeating a text value instead of using an ID). Relationships work but may rely on application logic rather than database constraints.	(6-13 points) Database exists but is "flat" or inefficient. Tables may be unrelated or store multiple values in a single cell (e.g., comma-separated lists).	(0-5 points) No database connection, or uses hard-coded JSON files instead of a real SQL database.	/ 25

Endpoint Execution & Status Handling	Excellent (25-30 points) 30 points	Good (16-24 points) 24 points	Needs Improvement (6-15 points) 15 points	Unsatisfactory (0-5 points) 0 points	Criterion Score
--------------------------------------	---------------------------------------	----------------------------------	--	---	-----------------

Endpoint Execution & Status Handling	Excellent (25-30 points) 30 points	Good (16-24 points) 24 points	Needs Improvement (6-15 points) 15 points	Unsatisfactory (0-5 points) 0 points	Criterion Score
Endpoint Execution & Status Handling	(25-30 points) All endpoints work as documented. Full CRUD (Create, Read, Update, Delete) is flawless. Correct HTTP Status codes (200, 201, 404, 500) are returned for all scenarios.	(16-24 points) Main "Happy Path" works (can Read/Write), but edge cases (e.g., deleting a non-existent ID) cause errors or crashes. Status codes are inconsistent.	(6-15 points) Basic "Read" works, but "Write/Update" operations are buggy. Server crashes frequently during testing.	(0-5 points) API does not start or endpoints return 404/500 errors consistently.	/ 30

Identity Management & Data Protection	Excellent (17-20 points) 20 points	Good (11-16 points) 16 points	Needs Improvement (4-10 points) 10 points	Unsatisfactory (0-3 points) 0 points	Criterion Score
---------------------------------------	---------------------------------------	----------------------------------	--	---	-----------------

Identity Management & Data Protection	Excellent (17-20 points) 20 points	Good (11-16 points) 16 points	Needs Improvement (4-10 points) 10 points	Unsatisfactory (0-3 points) 0 points	Criterion Score
Identity Management & Data Protection	(17-20 points) Passwords are securely hashed (bcrypt). JWT implementation is secure (token generation & verification). Protected routes correctly block unauthenticated requests.	(11-16 points) Passwords hashed. JWT works but implementation has minor flaws (e.g., infinite expiry). Middleware is present but not applied to all private routes.	(4-10 points) Basic Auth attempts made, but flaws exist (e.g., returning the password in the user object). Tokens may not verify correctly.	(0-3 points) SECURITY RISK: Plain-text passwords stored in DB, or no authentication implemented at all.	/ 20

Code Quality & Documentation	Excellent (14-15 points) 15 points	Good (10-13 points) 13 points	Needs Improvement (4-9 points) 9 points	Unsatisfactory (0-3 points) 0 points	Criterion Score
------------------------------	---------------------------------------	----------------------------------	--	---	-----------------

Code Quality & Documentation	Excellent (14-15 points) 15 points	Good (10-13 points) 13 points	Needs Improvement (4-9 points) 9 points	Unsatisfactory (0-3 points) 0 points	Criterion Score
Code Quality & Documentation	(14-15 points) Code is modular (Routes, Controllers, Models separated). Naming is professional. Postman/Swagger/Talend documentation is complete with examples for every route. Comments found throughout and is highly readable and maintainable.	(10-13 points) Code works but is messy (logic in route files). Documentation exists but is missing examples or some params. Comments are found but sparse and/or inconsistent formatting. Code is mostly readable.	(4-9 points) "Spaghetti code" (everything in server.js). Documentation is vague or incomplete. Significant redundancy, poor naming conventions, lack of comments, or messy formatting make the code difficult to read and understand.	(0-3 points) Code is unreadable or undocumented. No documentation provided.	/ 15

Total

/ 100

Overall Score

Level 5

95 points minimum

Level 4

80 points minimum

Level 3

60 points minimum

Level 2

25 points minimum

Level 1

0 points minimum