

Project Evaluation Form

Mandatory assessment: max 100 points • Bonus: up to +20 points (overall max 120)

Project title	Persona Budget Tracker	Team / members	Yiğit Yağız & David Mikolaj
Deployment URL (spider.foi.hr)		Repository / source-code link	https://github.com/yeet30/Personal-Budget-Tracker
Evaluator		Date	
Notes (optional)			

Gating checklist :

Requirement	Met (Y/N)	Negative points	Notes
Deployed and runnable on spider.foi.hr		-20	
Authentication not present; passwords stored securely (not plain text)		-10	
Role-based authorization enforced on backend API		-10	

Mandatory assessment (max 100 points)

Area	Criterion	Max	Points	Notes / evidence
System Design & Documentation 15 points (minimum required 5 points)	Architecture (frontend/backend/db + integration points)	5		??
	Data model / ERD + relationships + constraints	3		Can be found in the project.
	Roles & business rules (incl. 2 regular user types)	3		Already submitted
	API documentation (endpoints, auth, errors, examples)	4		Can be found in the project.
Backend Implementation 35 points (minimum required 5 points)	Core entities & CRUD (Users, Categories, Main entities + status history)	15		Managed core entities such as users, categories, and primary domain objects.
	Business rules enforced server-side (per roles/types)	7		Business rules based on user roles, restricting certain API operations so that only authorized roles could read, create, update, or delete specific data types.
	Data integrity & validation (mandatory, unique, FK, invalid data blocked)	5		Enforced mandatory fields, unique constraints, and foreign key relationships, while rejecting any invalid or inconsistent data before it could be stored in the

				database.
	Security & access control (auth, confirmations, secure passwords)	5		Implemented user authentication, password hashing, and authorization services, making sure that only confirmed and authorized users could access or modify protected values.
	Error handling + audit logging (who/when for critical actions)	3		??
Frontend Implementation	Role-based UI aligned with backend capabilities	10		Adjusted the UI and pages based on user roles, showing or hiding features aligned with the backend's authorization rules.
(Angular)	Usability, responsiveness, clear error messages	8		Provided input feedback, loading messages and displaying server side errors.
35 points	Server-side handling of large data + filtering + sorting	6		Used server-side processing to handle large datasets efficiently, implemented filtering and sorting on the backend.
(minimum required 5 points)	Chart in at least one place about bussines data (not system data e.g. logging)	4		Listed the transaction history of each budget and their impact on the budgets.
	Angular code quality & standards (services/guards/interceptors)	4		Organized logic into services, used guards for route protection.
	Integration stability (consistent API use, auth state handling)	3		API usage and proper handling of authentication state.
Presentation / Demo (mandatory)	Clarity & structure	7		
	Functionality demonstrated (end-to-end scenarios)	8		
TOTAL (mandatory)		100		

Bonus (up to +20 points; total bonus capped at 20)

Bonus features (max 10 points each)	Evidence (where/how)	Points	Notes
JWT (access/refresh) or robust session handling (incl. CSRF where applicable)			Stored user session cookies in the database, allowing persistent logins.
PDF report export (e.g., categories, tasks, audit trail)			
CAPTCHA / rate limiting / brute-force protection			

TOTP / 2FA			
XSS hardening (sanitization, CSP basics)			
Other relevant, well-integrated improvement (specify) not mentioned previously			
TOTAL (bonus)		max 20	

Final score summary

Component	Max	Awarded	Notes
Mandatory	100		
Bonus (capped)	20		
Negative points	-40		
TOTAL	100		