**Seminarul 1 — Partea 2 (Ghid arhivă + Worksheet)**

# Ghid de utilizare a arhivei „s1p2-lab.zip” & Worksheet detaliat

## 1. Ce conține arhiva  
- `src/lib/\*` — funcțiile pure ale laboratorului (parsare, transformare, filtrare, formatare).  
- `src/cli-runner.js` — orchestrator CLI cu I/O injectabil pentru teste.  
- `bin/cli.js` — intrarea CLI (se poate „linka” ca bin).  
- `data/\*` — fișiere exemplu.  
- `tests/vitest/\*` și `tests/jest/\*` — suite paralele cu aceleași contracte.  
- Config: `eslint.config.mjs`, `vitest.config.ts`, `jest.config.cjs`, `babel.config.cjs`, `.gitignore`.

## 2. Pași de rulare  
1. `npm i`  
2. `npm test` (rulează Vitest & Jest)  
3. `npm run dev` (raport tabelar pe `data/sample.csv`)  
4. Experimentează cu opțiuni: `--format json`, `--limit 10`, `--faculty CSIE`, `--new-only`.

## 3. Worksheet (detaliat)  
- \*A. Parsare\*: `parseCSV` tratează antetul ca schemă; normalizează `interests` și `is\_new`.  
- \*B. Transform\*: `countBy` și `groupBy` compun agregări; `topK` extrage primele K după comparator.  
- \*C. Filtru\*: `filterNewMembers(refDate, days)` și `byFaculty(name)`.  
- \*D. Runner\*: `run(argv, io)` gestionează usage, fișier lipsă, format invalid și formatul de ieșire.

### Checklist de acceptare  
- [ ] Testele unitare trec (Vitest & Jest).  
- [ ] CLI returnează cod 0 pe căi fericite, 1 pe erori.  
- [ ] `formatTable` produce secțiunile cerute (Total, By faculty, Top interests).  
- [ ] Codul este lint‑clean.

## 4. Troubleshooting  
- \*\*ESM în Jest\*\*: folosește `babel-jest` + `extensionsToTreatAsEsm`.  
- \*\*Eșec pe CSV\*\*: verifică separatorul și `  
  
` vs `  
`.  
- \*\*Date lipsă\*\*: normalizează cu valori implicite și `??`.

## 5. Extensii sugerate  
- Export `.md`/`.html`, \*stdin\* pipe, opțiuni scurte `-f`, `-l`.