

# Final Project Instructions

You are allowed to submit the final project in pairs.

In this Ex your job is to expend Ex2 code or your partner's Ex2 code.



You will

1. Expend your app
  - a. Invent a more complicated story around your Ex2 story.
  - b. Idea & User are entities. Add at least 4 more entities
  - c. Add at least 10 more endpoints and 4 more screens (can be virtual screens, yet meaningful)
2. Split out the persistence ability to it's own server
  - a. The code that persists the data should get spin off to it's own http/express server - call it *persist.js*
  - b. All the new entities must be persisted as well
  - c. The new persistence server and the main server should communicate using http
  - d. Add */data/persist* endpoints (logged-in mode only). It should persist immediately and return HTTP response only once the data has been persisted successfully
  - e. Add */data/clean* endpoints (logged-in mode only). It should clean and persist the data immediately and return HTTP response only once the data has been cleared and persisted successfully
  - f. Make sure that once your server starts up it reads the data from the persistence server

3. (**OPTIONAL:** up to 3 point Bonus) Convert it to a mobile app and use standard FWs
  - a. Learn how to use apache Cordova & Bootstrap
    - i. <https://cordova.apache.org/docs/en/latest/>
    - ii. <https://ccoetraets.github.io/cordova-tutorial/create-cordova-project.html>
    - iii. <https://getbootstrap.com/>
  - b. Build your app as a Cordova Android app
    - i. Test it
    - ii. Add at least one smartphone-only feature
  - c. Build your app responsive leveraging the bootstrap framework
4. Make sure you follow the below technical instructions:
  - a. Make sure your server is tolerant to DOS attack.
    - i. Make sure that upon exception your server does not crash
    - ii. Make sure that upon concurrent entrance of 3-5 users your server does not crash
  - b. You are not allowed to use "var" in JS ( i.e. you should use "let")
  - c. Express.js is highly recommended
    - i. Read: <https://expressjs.com/en/starter/hello-world.html>
      1. <https://expressjs.com/en/guide/routing.html>
      2. <https://expressjs.com/en/guide/writing-middleware.html>
      3. <https://expressjs.com/en/guide/using-middleware.html>
  - d. Add a **readme3.html** that consists of answers to the following items:
    - i. Your ID, name
    - ii. what's the story of your app?
    - iii. Instructions how to use your app - What was challenging?
    - iv. What was fun?
  - e. Zip everything and submit - **EACH STUDENT MUST SUBMIT SEPARATELY WITH THE CORRECT INFO IN THE README FILE.**

How do we score this project?

- We will test it utilizing chrome mobile emulator
- 85 points on the execution
- 10 points on experience and style (mostly user-experience)
- 5 points on utilization of the browser features in addition to doing async programming correctly

**Good luck**