CS157A F19 – Project requirements

1. **The goal** of the project is to design, implement, and test a Web-based or Java-based application with database connectivity (i.e. JDBC for Java) on an assigned topic. The focus is on implementing data transactions, such as data entry, modifications, and data display.
2. Projects are <u>done in teams of three</u> (3). Ask in the class or use a discussion forum to find a team/team member. All team members must use the same chosen technology stack.
3. The <u>team chooses one topic</u> from the 5 posted in the "Project topics" document. Each topic can be taken by at most two teams.
4. You need to declare your team, a topic, and technology stack used on a discussion forum on Canvas, the instructor will accept your choice by responding to your post.
5. Ask the instructor or RA (Brett.Dispoto@sjsu.edu) for the further clarification of requirements or help with the project.
6. Utilize the Canvas page created for your team (i.e. discussion forum to communicate with team members/instructor/TA).
7. You need to <u>create a GitHub repository</u> and show equal commitment from all team members (use version control system, such as Git to commit). You need to add all team members and RA (Git username: *DispotoBrett*) as collaborators.
8. No collaboration outside your project group is allowed –not doing so is regarded as a violation of academic integrity.
9. **Deliverables:** source code, GitHub link, technical report, and (if time permits) presentation with a live demo.
10. Tentative **due date**: 11/27 23:59 PM (any changes to the deadline will be posted on Canvas)
11. **Grading criteria**:
    a. Designing an optimal table model (in normalized form – you need to provide a formal proof that your DB is in 3NF).
    b. A correct definition of database tables and referential integrity constraints.
    c. Correct usage of database connectivity libraries (i.e. JDBC)
    d. Correct implementation of the functionality required for the project.
    e. Testing the application by defining and performing test cases with sample data.
    f. Technical documentation:
        i. Problem statement – analysis of the problem area to be solved
        ii. Design of the solution – high-level architecture of your system, database design (table model in 3NF), UI design.
        iii. Description of the chosen technology stack for the solution and any other IT tools used (i.e. for the design, for teamwork, etc.)
        iv. Description of the implementation, (i.e. description of the tables, attributes, constraints, classes etc.). For Java, it is recommended to use Javadoc utility to generate documentation.
        v. Instructions to deploy and run your application.
        vi. Listed contributions of each team member.
    g. Teamwork (GitHub) - your individual grade for the project will be adjusted based on your commitment as shown on GitHub
    h. Presentation (if time permits)

General Rubric for CS157A Project

	Beginning **1**	**Needs Improvement** **2**	**Acceptable** **3**	**Accomplished** **4**	**Exemplary** **5**
Understands the Problem and Requirements	Student's work shows incomplete understanding of problem and/or requirements	Student's work shows slight understanding of problem and requirements	Student's work shows understanding of problem and most requirements	Student's work shows complete understanding of problem and all requirements	Student recognizes potential conflicts b/t requirements and seeks clarification from client/user
Uses Appropriate Data tables	No use of data tables	Use of data tables; but are none are appropriate for task	Use of data tables; but some are not most appropriate for task	Use of data tables; all are appropriate for task	Uses advanced table model that improves program performance
Uses Appropriate Algorithms and libraries	Student 'hacks out' program with no thought to algorithm design	Student chooses/ designs algorithm(s) that are incorrect	Student chooses/ designs algorithm(s) that is/are correct but somewhat inefficient	Student chooses/ designs efficient algorithm(s)	Student researches tradeoffs b/t different algorithms & implements the results of this research
Designs Appropriate User Interface	Implements very poor I/O functionality	Only implements basic I/O functionality	Some concepts of 'user-friendly' I/O used (e.g. prompts on input & labels on output)	Uses well-designed 'user-friendly' I/O interface appropriate for task and client	'User-friendly' I/O interface with GUI components
Tests Program for Correctness	No evidence of any testing by student	Evidence of only one case tested	Evidence of a few cases tested	Evidence of "typical cases tested, but only assuming valid inputs	'Robust design' with extensive testing.
Documents Program	Absolutely no documentation other than name.	Little or no documentation; few or no internal comments	Some documentation, but sparse internal comments	Complete documentation with numerous internal comments	Thorough documentation; Use of javadoc or similar docs generator software