Project Phase 3

Start Assignment

* **Due** Nov 17 by 11:59pm
* **Points** 100
* **Submitting** a text entry box
* **Available** Oct 27 at 12am - Nov 17 at 11:59pm

Congratulations! You’ve made it to Project Phase 3, the Implementation phase. Your task now is to implement the system you’ve been designing to this point.

Just as with Phase 2, you should start working on Phase 3 as soon as it opens, and not wait for your feedback from Phase 2 to begin work.  Many times, we find that most teams have found the same issues in the Phase 2 submission while they are coding, debugging, and testing their system and correct them as they crop up.

**Phase 3 Deliverables:**

1. Implementation source code
   * This must be committed to **GitHub just before or immediately after your demo.**
   * Do not publicly share your code, in partial or in whole, as this violates the GT Honor Code.
2. A live demonstration of your implementation
   * Your team should create an outline and come up with the appropriate content and features of your app that you think need to be demonstrated.  (To allow for creativity, there won't be giving any guidelines on what or what NOT to present - this is up to you to figure out!  The project spec provides a basic order of operations you can expand on.)
     + Rehearsing your demo several times beforehand is highly recommended!
   * We will describe on Piazza the sign-up process for teams to choose a time to demonstrate their project to an assigned TA.
     + Each slot will be a *maximum*of **25 minutes**.
   * **We may provide sample data or seed data guidelines**, shortly before the demo period opens.
     + Teams should be prepared to add this data for the demo within a short timeframe.
   * Your team will have only one demonstration opportunity, and you are graded on what functionality is presented during that demonstration.
     + **If a functionality or requirement is not covered during the demo, or it did not work or display properly for any reason, credit will not be granted.**
       - We will not accept requests for a code review or re-do of the demonstration to make up for credit lost due to missing functionality.
     + Except for emergency situations as listed in the syllabus, teams that do not appear for their scheduled demonstration time will receive a grade of zero for Phase 3.

**How You Should Work:**

We require teams to perform all source control on their assigned GT GitHub Enterprise team repo. This will ensure that you have checked in everything necessary, because if anyone fails to submit something important, the rest of the team will pick up on it and make sure it’s corrected.

Students have a wide variety of backgrounds and skill sets which means completing each phase may be easy for some and time consuming for others. The amount of time spent on the project does not always correlate to grade performance. Always ask yourself if the deliverables meet the assignment expectations, and if you’re unsure, ask prior to the deadline for clarification. When grading your submissions, we are only looking at the deliverables themselves, not how much effort.

**All team members should participate in the actual coding of the source files**(.php, .sql, .html, . rb).  Keep in mind, phase 3 is the most time consuming of the three project phases. The best strategy is to rank pieces of functionality by difficulty then assign them to each member based on ability.  Individual coding style will not be considered during the Phase 3 grading. Do NOT wait until the last minute to begin coding as this will reflect poorly on your demo!

All team members are expected to pull equal weight towards completing the project.  There are many online resources available to learn enough to be successful with the project in the time allowed.  One approach is for all team members should code their source files in individual branches and merge those into the main branch via pull requests. Note that we will review team evaluation comments and Git commits, and in the event it appears a teammate did not substantially contribute to phase 3, will adjust their grade so they do not share in the grade earned by the teammates who did contribute.

**Implementation Guidelines**

* We suggest using WAMP/WAPP/MAMP/MAPP (which means the combination of Mac, Windows, Apache, MySQL or PostgreSQL, and PHP, and can be downloaded as a package for either OS).
  + You are welcome to implement your system in any language or platform you wish (Python, Java, Ruby, .Net, etc.) subject to the caveat that you must **write your own SQL**and implement your own logic on the database.
  + Use of ORM or ORM-like tools such as hibernate, query-by-example frameworks (like Power Query) and others which handle your database interaction (and even your schema design) are strictly prohibited.
  + **SQLite or MS Access are not allowed**.
  + A standalone, full featured *relational*DBMS is required for the project: PostgreSQL, MySQL, MS SQL Server, or even Oracle.
    - Non-relational noSQL: Hadoop, Cassandra, MongoDB, etc. are not allowed.
    - “Built-in” or application-hosted databases should not be used.
    - Non-relational databases that utilize SQL as an interface are also not allowed.
    - Data warehouse-specific platforms, such as Snowflake or Teradata should also be avoided.
* You have great latitude in your UI, as long as you have a functional UI the details do not matter.
  + You must have a navigable user interface.
    - Do not make the mistake a team made a few semesters ago of claiming that their “UI” was posting requests using the Postman tool and JSON or XML payloads, or another team that simply ran queries at a command line prompt. These will not be acceptable.
  + If your implementation platform of choice has a GUI framework built into it (such as Windows Forms or Python's Tkinter) those are acceptable if you do not wish to build a web-based application.
  + If you are unsure if your UI design will be acceptable, please ask us in a private message to your Team+Instructors group on Piazza.
    - * This shouldn’t be needed for most, but if you think you’re being too clever in your design, just check with us first.
* The expectation is to have a fully functional finished product (no page crashes, all buttons work, data validation and error handling correctly handled, all queries successfully commit to the backend, reports are accurate, etc.)
* Business logic constraints must be ensured, and your app must prevent users from performing those actions which are prohibited (grayed out/hidden buttons, notifications, etc.).
* We recommend NOT using a **web hosting/cloud service**for the demo, but if you do, ensure you have a local backup ready to go in case there is a disruption to your hosted application.

**Demonstration Guidelines**

* Not all team members are required to attend the demo.
  + If a teammate chooses not to attend, and due to this is evaluated poorly on the team evaluation, no adjustment will be made to that individual's participation grade.
  + Team members who do attend should be on time, joining at least 5 minutes ahead of schedule.
  + If the team arrives late and/or starts late, you will be limited to present within whatever time from your original 25 minutes remains, meaning you may not be able to completely demonstrate your system, and likely negatively impact your grade!
* Your demonstration will be recorded and will be watched at least twice to ensure we didn’t miss anything that you demonstrated.
* One team member should act as "presenter" for the Phase 3 demo and need to screen share.
  + The presenter should run your team's project local to his/her device using a **good internet connection(>15 Mbps**).
  + To conserve bandwidth, webcams should be disabled prior to joining the demo.
  + It would be smart to have a "backup presenter" ready if something happens to your main presenter’s computer during the demo.
* The goal of the demo is to ensure your app works and the interaction between the UI (buttons/forms) and DB (insert/delete/select) is functional.
  + If sample or seed data is provided, it is recommended to begin your demonstration by reviewing any reports to validate the data was loaded correctly.
  + Your grader may also ask you to demonstrate some specific test cases or other functionality to ensure your app works as expected.
  + Be prepared to also show the results of operations in a SQL command prompt or editor connected to your database when prompted to do so by your grader.

**How and What to Submit**

On Canvas, you will submit the final commit ID for your implementation code by following these steps:

1. If you use branching, make sure to move all work onto your main branch.
   * If you provide a commit that is not from the main branch, there may be a penalty on your grade.
   * Make sure all commits are pushed to Georgia Tech's GitHub, not merely committed locally.
2. Find the commit ID of the last commit using any Git tool you choose
   * “git log -1” will quickly display the last commit ID
   * A commit ID is a SHA hash such as “40da96c63a0a82d71e531cf137b6c95e282f9289”
3. One team member should submit the commit ID in the submission text for this assignment.  
   For example, for commit *0b0134604ced2182b8b11d7b299ac59dc70ed722*, someone would submit: 0b0134604ced2182b8b11d7b299ac59dc70ed722
   * Please submit a full commit ID as we need it in order to navigate directly to your commit and **do not** submit anything in addition to the commit ID.
   * Shortened commit ID submissions may receive a penalty.
   * Only one team member needs to make the submission, once submitted, it is associated with and visible to everyone on the team.
   * Should you need to submit a new commit, any team member may do so, and again, the new submission will become associated with everyone on the team.
   * Do not include any other content than what's specified here as part of your submission, and be sure to make a submission, not a submission comment.
   * If Canvas does not show this assignment as submitted, you did not properly submit, and you should review [this Canvas help articleLinks to an external site.](https://community.canvaslms.com/t5/Student-Guide/How-do-I-submit-a-text-entry-assignment/ta-p/303) to see how to do it.
   * Any team member will be able to submit the commit ID to be graded. Please make sure every team member agrees on the correct commit ID before submission.  Your team's final submission will be graded, so please be sure all team members agree on what will be submitted. We will not grade previous submissions and will not consider any requests to do so.

You **may**perform work on your project after the Phase 3 assignment due date listed in Canvas up until your demo date/time.  However, to ensure all teams are on an equal footing, we will not be able to respond to any inquiries regarding the project once demos have begun. If submitting after the due date but while the assignment is still open, Canvas may mark your assignment submission as "late" but do not worry, we will not consider it late or penalize for a late submission.  (Since teams' scheduled demonstration times fall throughout demo week, to avoid any confusion, we are setting this assignment to be due before demos begin, but it will still accept submissions until the end of the demo period to allow submitting a commit ID just before or just after completing your demo.)

Feel free to post a private question to your Team+Instructors group on Piazza if you need further clarification for your team project.