This rubric shows both what you are expected to complete for the midterm project as well as how it will be graded. **Submit a word or PDF file of your project and the DB file**. **As a general requirement for the project, submit all necessary screenshots to present your work fully, completely, and clearly. Project due 11/13, 11:59PM, late: 50% till 11/15, zero afterwards.**

**Requirements of DB Backend Design (50% of total project grade):**

| *You must use a report writing style to describe your work, not just screenshots* | **Score** |
|---|---|
| 1. describe system operations; sketch system meta model | 80 |
| 2. Entity Relationship Diagram, complete with connectivity and cardinality | 100 |
| 3. Relationship Schema | 100 |
| 4. Normalization. Normalize for at least 4 tables. You must document the normalization process for your DB. If normalization gives "strange" results and you would like to override the results (e.g. keep a table in 2NF instead of breaking it to two in 3NF, you must justify and explain in details your decision. The goal is to have a robust DB design.<br>** If all your tables do not need normalization, create the dependency diagram for each table and state why it does not need normalization. | 100 |
| 5. Data Dictionaries for at least 4 tables | 100 |
| 6. Include (minimum):<br>- minimum of 4 Tables (both design and data views screenshots), each with at least 10 records | 100 (25/T) |
| 7. Include (minimum):<br>- 1 sorting query<br>- 1 formatting and concatenation query<br>- 1 left outer join query<br>- 1 right outer join query<br>- 1 update query<br>- 1 insert query<br>- 1 delete query<br>Submit screenshots of each SQL query in the two views: SQL, and Data. | 420 (60 each) |
| **Total** | **1000** |

**Requirements of DB GUI design (40% of total project grade):**

| *You must use a report writing style to describe your work, not just screenshots* | **Score** |
|---|---|
| Submit screenshots of HTML code, Python code, all Webpages running in browser, all forms filled, verification of results (e.g. add data, extract info) via using SQL queries in terminal | |
| 1. Front End design: design of your website (front end) that supports your biz logic and operations. Show your website home page and how all pages link to each other and the site map (how user navigates) of your website | 140 |
| 2. Front End implement: At least five Web pages, including the home webpage. Submit 2 screenshots/page (rendered page and HTML code)<br>• Each Webpage must have HTML elements to support its content to support its objective and make it functional<br>• Each Webpage must good color mix of its elements and its background to make it visually appealing and attractive<br>• Entire website must use ONLY one css file to support its color scheme<br>• Each Webpage must use  tag to format its content for attractive layout<br>• Each Webpage must have a form for user interactions, except home webpage<br>• All Webpages must use all HTML elements collectively: text, colors, | 240 (60/Web page) |

| | |
|---|---|
| links, images, forms, formatting. Collectively means no need to use all elements in each page unless your Web design warrants it<br>• All Webpages must use all form elements collectively: textbox, checkbox, radio, dropdown list, text area, submit, reset<br>• Forms of Webpages must support all DB tasks: add data, extract info, generate reports, generate plot/charts<br>• NO need for backend development of your website and Webpages for this task | |
| 3. Front End forms Processing: use Python to process user input to:<br>- Add data to project DB<br>- Extract info. From project DB<br>- Create reports from project DB<br>- Create charts/plots from project DB | 360 (45/form) |
| 4. Include (minimum):<br>- 4 uses cases of how to use the above forms: add data, extract info, create reports, create charts/plots | 60 (15/use case) |
| **Total** | **800** |

**Overall quality of project depth and execution (10% of total project grade):**

| *You must use a report writing style to describe your work, not just screenshots* | **Score** |
|---|---|
| Quality of project DB (backend) depth measured by the number of tables beyond the 4 minimum. Does the project DB need more tables to make the DB design robust and complete | 50 |
| Quality of project GUI (frontend) depth measured by the number of Webpages beyond the 4 minimum. Does the project website need more Webpages to make the GUI design robust and complete | 50 |
| Quality of project DB (backend) execution measured by the number of each table attributes and number of records beyond 10 required minimum. Does the project DB tables need more attributes to make the DB tables design robust and complete | 50 |
| Quality of project GUI (frontend) depth measured by the content and formatting of Webpages beyond the required HTML elements. Does the project website need more formatting using CSS to make the GUI look and feel more exciting and attractive | 50 |
| **Total** | **200** |