Project Report:

- Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).
    i. We enhanced our platform by adding three advanced search/view features to help users better understand the compatibility and information of each component.
        1. The first search focuses on checking the compatibility between the selected GPU and CPU. Users can input a partial or full name of the desired GPU and CPU with their expected budget. The platform can then create a list of PC options within the provided budget, including the specified GPU and CPU.
        2. The second advanced search focuses on ranking manufacturers based on the number of products they provide, from the most to the least. It allows users to quickly identify which manufacturers dominate the market and help users make decisions by comparing the products from different brands
        3. The third advanced view feature allows users to access and review the PC lists they have chosen, helping users better understand their build.
    ii. We initially proposed a "Hot" feature that would let users upload custom PC builds, share component details, costs, and performance reviews, and engage with the community through likes and comments, with popular posts highlighted. This feature also included a search tool to filter posts by components. However, we later decided to remove it to focus on data analysis rather than showcasing popular PC configurations. As a result, the project shifted from a community-driven content platform to a data-driven analysis tool.
    iii. In the original proposal, we planned to utilize e-commerce APIs from platforms like Amazon to retrieve information about components. However, in the current implementation, we shifted to web scraping as the primary method for data retrieval.

- Discuss what you think your application achieved or failed to achieve regarding its usefulness.
    i. Achievement:
        1. The application offers a user-friendly platform for selecting PC components, which makes the process of building a customized PC more efficient. Users can input the partial or full name of each preferred component. Then the platform could generate a list corresponding with the specified name. It helps users easily find the right components.
        2. Once users complete their build, the platform automatically checks the compatibility of all selected components. It eliminates the need for users to spend time manually verifying compatibility, which simplifies the process and reduces the potential error.
        3. After the compatibility of PC components are confirmed, the platform saves this PC build to the user's account. It allows users to revisit their configurations at any time, review their choices, or delete saved builds if they no longer need it.
        4. It also provides two advanced search options, providing the PC build with specified CPU, GPU, and expected budget, and the rank of manufacturer based on the amount of products they offer. It helps users better understand the information and compatibility across different components, which makes the PC build much more efficiently.
        5. This platform also includes components pages which provide the detailed information list of components. Users can input the name of the component they prefer to check its functionality. If the user is Admin, it has the access to adjust the product price within a proper range.
    ii. Limitation:
        1. This platform doesn't provide forums where users can discuss and exchange their ideas or experience of customizing personal PCs.
        2. The application doesn't achieve real-time data retrieval since we utilized web scraping to extract data instead of e-commerce APIs.
        3. While we do have some interactive features for users to explore the functionality and compatibility of components, the customization options could be expanded, such as

making a recommendation list of components when customizing a PC.

- Discuss if you change the schema or source of the data for your application
    i. Data Source: The original plan was to utilize APIs from e-commerce platforms such as Amazon to retrieve real-time component information. Due to API restrictions and time constraints, we decided to use web scraping to collect component data.
    ii. Schema: : The relational tables SupportCable and CanHandleRamSpeed in the original design are removed from the final design and actual implementation, the reason for removing them will be specified in the next section. The attributes of User table are modified, we changed its primary key from name to newly defined user_id, so therefore we can use a unique identifier as primary key instead of names, allowing users to have the same names. We also added two columns, “email” and “admin”, email allows better login functionality, and the admin column allows us to specify administrator users and develop functions restricted to admin users such as change the prices of components on the web end.
- Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?
    i. Differences: In the original design there were two relational tables SupportCable (holds the compatibility relationship between Motherboard and PowerSupply) and CanHandleRamSpeed(compatibility between CPU and Memory/RAM), in the final design we removed the two tables.
    ii. Why: SupportCable was removed since as we get better understanding of how power supply compatibility works, we realized that what determines the compatibility of power supply in a computer is not with motherboard, but in fact depend on the combined TDP(Thermal Design Power) of CPU and GPU, so the compatibility of power supplies are implemented using triggers in the database that always make sure the combined TDP of CPU and GPU never exceed the wattage of power supply. CanHandleRamSpeed was removed because the data we pulled does not include the necessary supported ram speed data that is

needed to build this table, and we couldn't find it anywhere else, so by the end this table was unfortunately removed.

iii. What do you think is a more suitable design? Using triggers to perform compatibility of power supply is definitely a more suitable design since it includes values from multiple tables. But if we could acquire the data needed and actually build the CanHandleRamSpeed table, it would be better than the current design.

- Discuss what functionalities you added or removed. Why?
  i. Added functions: We added a whole system of admin users, allowing users with permissions to do more than normal users can do, the admin users can change the prices of components on the web end, and since the GPU prices are always varying a lot on the market, admin users can check if a certain GPU product's price is abnormal (500 less than average of manufacturer or 500 less than average of average price of each manufacturer), and delete it if the price is abnormal. When the admin user changes the prices of components, we also have restrictions in the procedure that prevents input of prices that are abnormally higher or lower. We also have added advanced functionalities: getting a ranking of all manufacturers ordering by the number of products they have; as well as searching by keywords to find products they have in mind. We also added procedures to check the compatibility of a certain PC design. At last, we used a transaction to prevent the situation where the user selects a component but during the process its price was changed and causing trouble, by having the transaction we can stop the process of updating/creating PC plans when there's a price conflict.

  ii. Removed functions: the recommending function is slightly simplified, users can still provide a budget to get recommendations but not specify their "functionality" needs, since in our data there's not enough information about how the performance of components will be like and we are not that familiar with the technology to add them manually, instead the users now can provide keywords on CPU and GPU to get the recommended design, since these two are the most important part of a PC and somewhat contributes the most to the final performance, the users can still in a way specify their need for performance and functionality. For the same reason, we removed the function that allows users to get a description of

their PC plans' performance and functionality, as similarly the data do not have that information. The error report function is also removed, part of it was moved to admin users' check unusual price functionality, and the other part of reporting a new component and insert into the website is removed, since we decided that a professional technology website(or its administrators) should hold the responsibility itself to add the newest product and delete errors, instead of letting the users do it, if everyone can add/remove things this application would become a mess.

- Explain how you think your advanced database programs complement your application.
    i. Advanced programs associating with admin users permission like check and delete unusual prices and change prices of components gives us, the maintainers of the web application, more freedom and convenience to modify and organize the web application instead of always having to do it in the database, having interfaces for admin users makes it more like a professional application in real practice, which we can modify as it runs without causing trouble, while the power of administrator is not unlimited, if the admin user enters extremely strange price the process would be stop by program, just as it should be in the real world application. By having the transaction to prevent price conflict, we can avoid the practical situation of letting users view a wrong price and cause trouble, this enhances the dependability of our application. Searching keywords makes it easier for users to search the product they want, this is the function that all websites like this should have as a complete application. Recommendation is also important and is a key feature that makes the process of PC building more efficient and convenient. The program of pulling out ranking of manufacturers based on product count can give users who don't know the computer industry a straightforward way to familiarize with the brands and know who the big and popular manufacturers are.
- Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.
    i. Jin Fan: As a front-end designer, one technical challenge we faced was layout inconsistency. Since two team members were working on the layout simultaneously, we noticed that after one person

pushed their changes and the other pulled them, subtle changes in the layout would occur, even though the core functionality remained intact. To resolve this issue, we added unique prefixes to common element names like "table" and "button" in each page's CSS file. This approach minimized unexpected layout shifts and ensured greater stability in the design. Future teams working on similar projects can avoid this issue by using more specific, prefixed class names to prevent CSS conflicts.

ii. Encheng Xie: There are many restrictions on interacting with GCP. For login to OAuth2.0 API, we need to register app urls via GCP/API Services/credentials for origins to satisfy CORS policy with backend server also configuring them in corsOptions. For deployment, environment variables are likely to be missing, and we have to re-declare them on GCP/Cloud run/<service>/Container. Additionally, for frontend, tsx file has trouble accessing environment variables as it specifically requires 'VITE_' as the prefix, so that we also need to copy and re-declare them just as the deployment to GCP.

iii. Yanxin Jiang: Initially, we used EJS, which worked well for basic functionalities but with limitations when implementing more interactive features like search and filtering. Therefore, we decided to convert the front-end to React to enhance functionality. We have faced issues when fetching and processing data from the backend. Data becomes undefined even though data was fetched successfully. The problem was due to inconsistencies in data processing on the front-end. I resolved this issue by transforming and validating the data with the correct format. I also used React's state and effect hooks to handle asynchronous data fetching properly, which ultimately resolved the display errors.

iv. Tianze Yang: As the one responsible for the database, the recommendation function's implementation causes some trouble in efficiency. Since the function will generate a complete PC design of all 7 components and also satisfy compatibility requirements, the function's query will have to cross reference entries of all seven tables and one by one verifying their compatibility with one another, so the process is very inefficient and takes long. I modify the logic of joining and selecting piece by piece and made multiple changes to optimize the query in order to shorten the time for generating results as must as possible, and in the end the process would take

about 10 seconds, still much longer than the other operations, but considering the database server we use and the amount of data we have, I would say this is an fairly acceptable waiting time.

- Are there other things that changed comparing the final application with the original proposal?

  There were no other changes between the final application and the original proposal.

- Describe future work that you think, other than the interface, that the application can improve on
  i. Enhancing the Manufacturer Ranking System : The manufacturer ranking system could be made more sophisticated by incorporating multiple evaluation criteria to provide users with a more comprehensive view of manufacturer performance. This includes integrating user reviews and ratings sourced from web scraping or popular tech forums, tracking product popularity based on the frequency of user selections in PC builds, and incorporating product performance metrics such as speed, efficiency, and lifespan. Additionally, users could be given the ability to filter manufacturers based on specific criteria like product type, price range, or user rating, making it easier for them to identify the top-performing brands and make more informed purchasing decisions.
  ii. Improving Data Visualization : Advanced data visualization could be introduced to make complex information more digestible and visually appealing. The platform could offer interactive visualizations, such as bar charts, heatmaps, and comparison tables, to display manufacturer rankings, product price trends, and performance metrics for various PC components. These visualizations would allow users to spot patterns, analyze trends, and compare options at a glance, leading to better purchasing decisions. By incorporating interactive elements, users would be able to explore and manipulate data in real-time, resulting in a more intuitive, dynamic, and engaging experience.

- Describe the final division of labor and how well you managed teamwork.
- Encheng Xie(encheng2):
    - Developed entire backend server with apis connecting database for web functionality
        - querySQL() to perform query to database
        - Store user data using Google OAuth2.0
        - 8 major routes to handle requests sent by frontend
    - Configured GCP API, firewall and VM to ensure a consistent environment for development
    - Deployed both ends to GCP for serverless run
- Tianze Yang(tianzey2): Built and maintain the content and relations of the database, including tables, attributes, constraints, relations, triggers, procedures to use by the backend and frontend, transactions. Write procedures for implementing functionalities the application needs. Modify relationships, indexes, keys and constraints as changes in the project need.
- Yanxin Jiang(yanxinj2): Built the front-end interfaces for the components page, with a name filter and price adjustment options, an advanced PC customization with specified GPU, CPU, and expected budget, and an overall dashboard layout.
- Jin Fan(jinf2): Designing the front-end interfaces for user-customized PC configurations, displaying saved PC builds, and showcasing the Manufacturer Ranking page.