1CS-639 Building User Interfaces, Fall 2020, Professor Mutlu

React 2 β (3 Points)

Improving Usability Using Heuristic Evaluation

In this assignment, you will put the ten usability heuristics we learned in class into practice toward improving the usability of your *React 2* α deliverable. You will focus on specific components of your design, identify potential violations of the heuristics, make design recommendations to address these violations, and implement recommendations that are feasible to create a new deliverable. Use this opportunity to make concrete design decisions about your project, to improve your design using the heuristics, and to build a keen eye for identifying usability issues as a UX developer.

Step 1—Identify A Focus. (0.2 Points) Review your *React 2* α deliverable with a critical eye to identify 3–5 "components" that you think are most consequential for user experience.

Step 2—Review the Heuristics. Review the ten usability heuristics we discussed in class from the slides, what principle each heuristic represents, and examples of the violations of the heuristics.

Step 3—Identify Potential Violations. (1.0 Points) Focusing on your components, inspect your design, considering each usability heuristic, for any violations of the heuristics.

Step 4—Develop Design Recommendations. (0.4 Points) For each violation you identified in the previous step, provide a design recommendation for addressing it, assessing its feasibility.

Step 5—Implement Your Recommendations. (1.4 Points) Implement the design recommendations that you identified as "feasible" in the previous step in your prototype, updating your design.

Submission Details

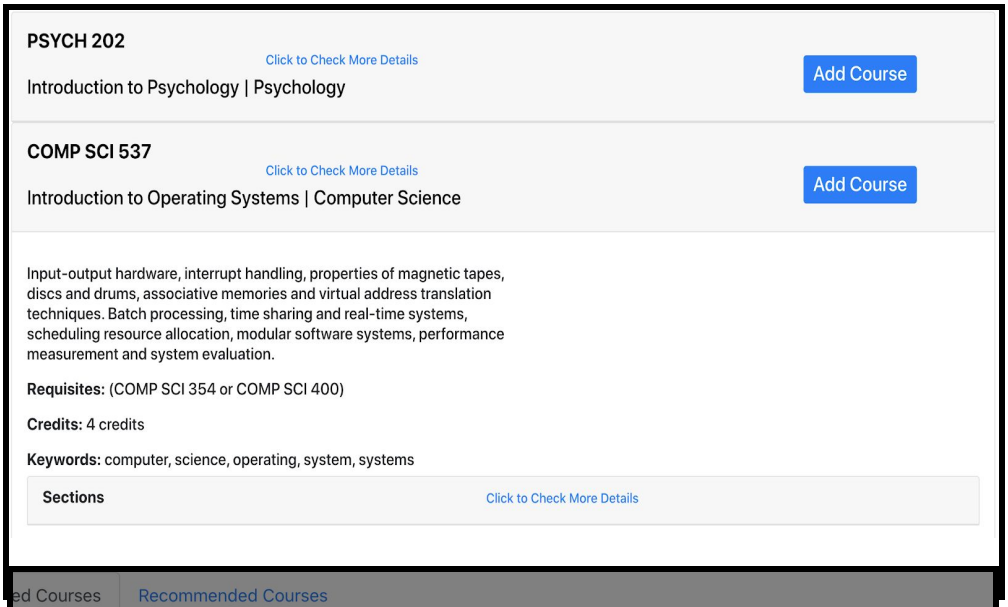
GitHub Classroom Starter Code

React 2 β will build on your implementation of React 2 α. You should copy your code from your React 2 α project to the React 2 β repository linked above, as that will be your starter code. When you commit and push, ensure that you are committing and pushing to the react2-beta repository, not react2-alpha.

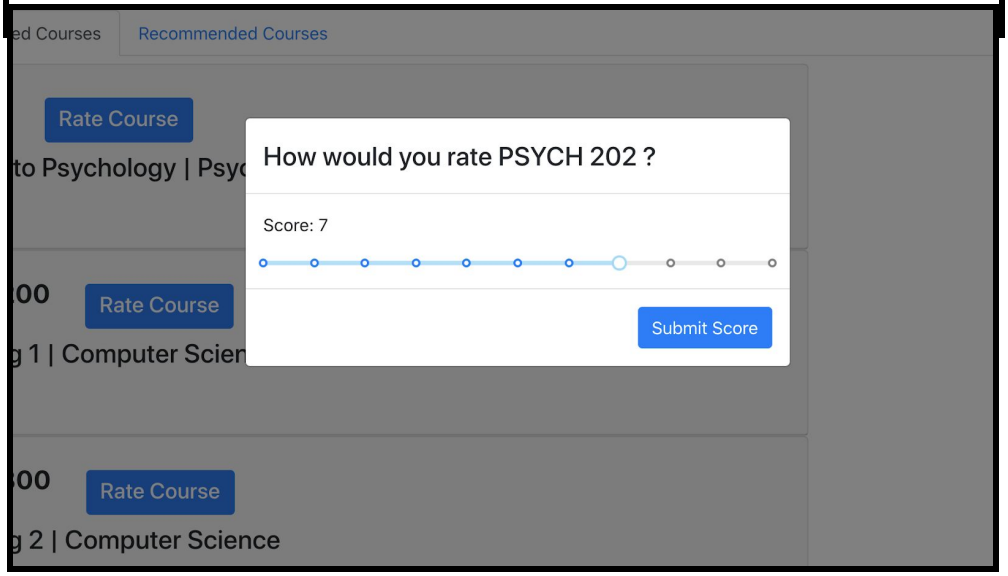
To complete the assignment, you will need to submit a completed version of this document as PDF to Canvas. In addition, you will submit your repository name and latest commit hash from GitHub Classroom, e.g. react2-beta-ctnelson1997, 2b0ef83.

Step 1. Identify A Focus. (0.2 Points)

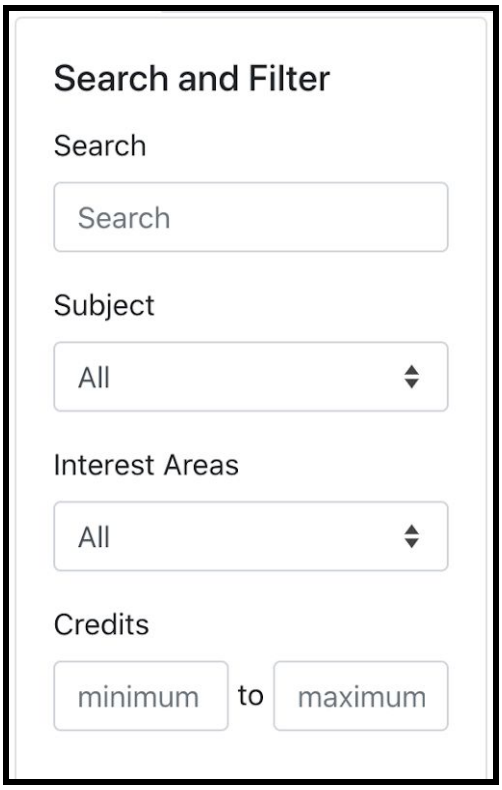
In this step, you will review your *React 2* α deliverable with a critical eye to identify 3–5 "components" that you think are most consequential for user experience and that you will put under the microscope of heuristic evaluation in the next step. In real life, your application might have hundreds of components, screens, or pages, and you will have to focus your efforts on a limited set that will make the most difference in terms of effectiveness and user experience. Similarly, you will review your design and identify 3–5 components to focus on. Here, a "component" can be the entire page/view (e.g., recommended courses) or a reusable component (e.g., the course component, the rating component), but not something as small as a button or label. Provide screenshots of each component below and provide a brief justification (1–2 sentences) of why you think each one is a critical component.



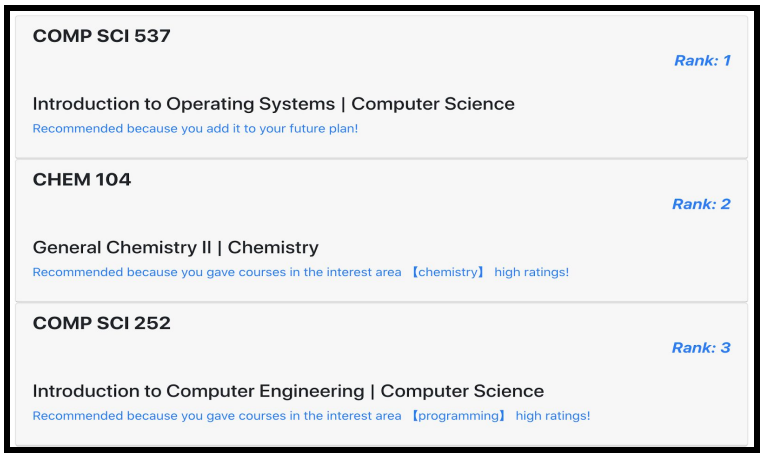
1. Course component is very important for users since it's the basic unit to display all courses' information and enable users to operate on courses



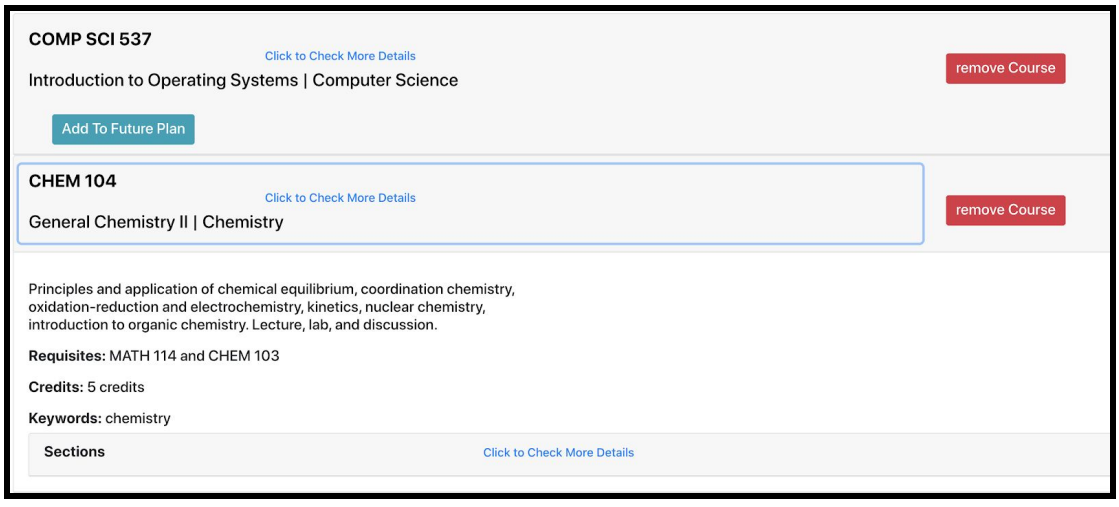
2. Rating component is another consequential part to build up the user experience. It makes it possible for users to rate completed courses. Additionally, it services as the first step(input) of the course recommendation algorithm



3. The next important component is the Search and Filter component, which highly improves the user experience of finding desired courses. It provides users with precise search results of courses they are potentially interested in.



4. The course recommendation component gives users suggestions on untaken courses based on their ratings on the taken courses, showing them courses they may be interested in order.



5. Cart component is also very important for user experience since it stores and shows all users add courses and allows users to operate them

Step 2. Review the Heuristics.

Carefully review the ten usability heuristics we discussed in class from the slides, what principle each heuristic represents, and examples of the designs that violate and support the heuristics. Below is a cheat sheet for Nielsen's ten heuristics that you can use in the next step. (This step does not have any deliverables.)

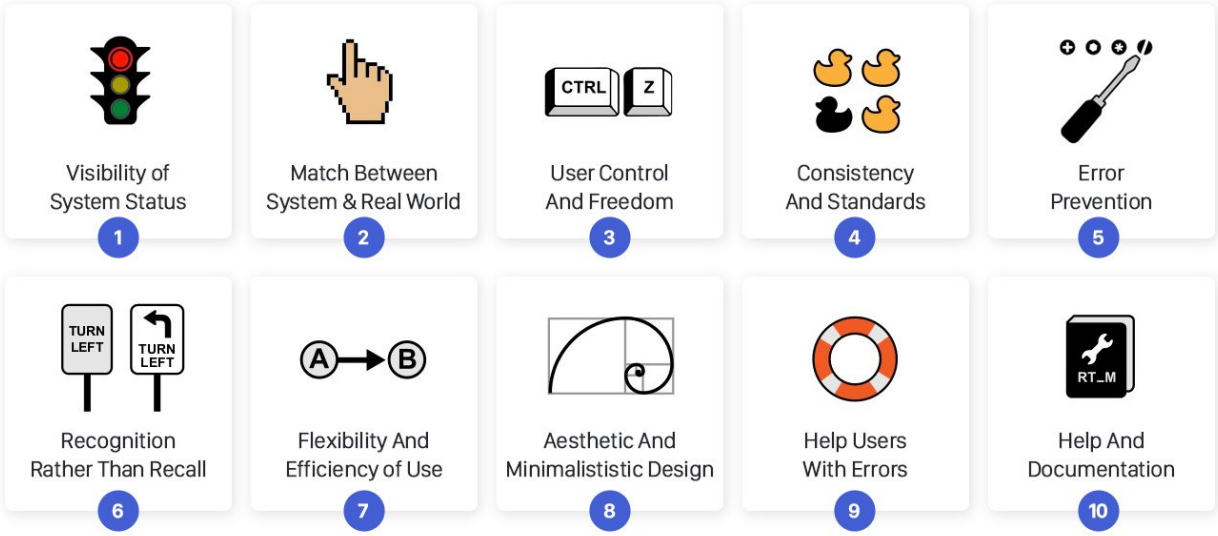


Image source: UX Collective

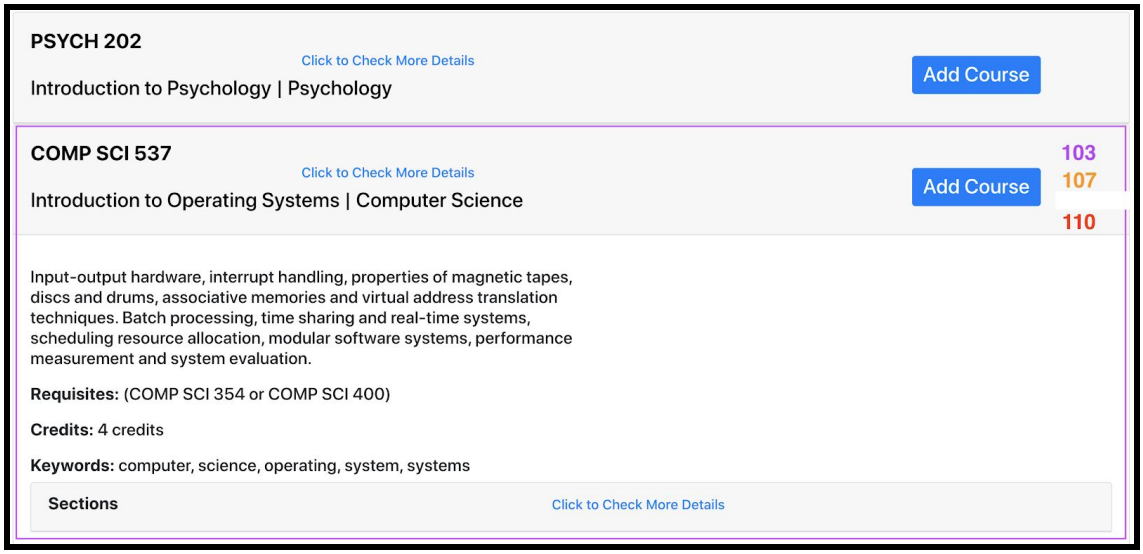
Step 3. Identify Potential Violations. (1.0 Points)

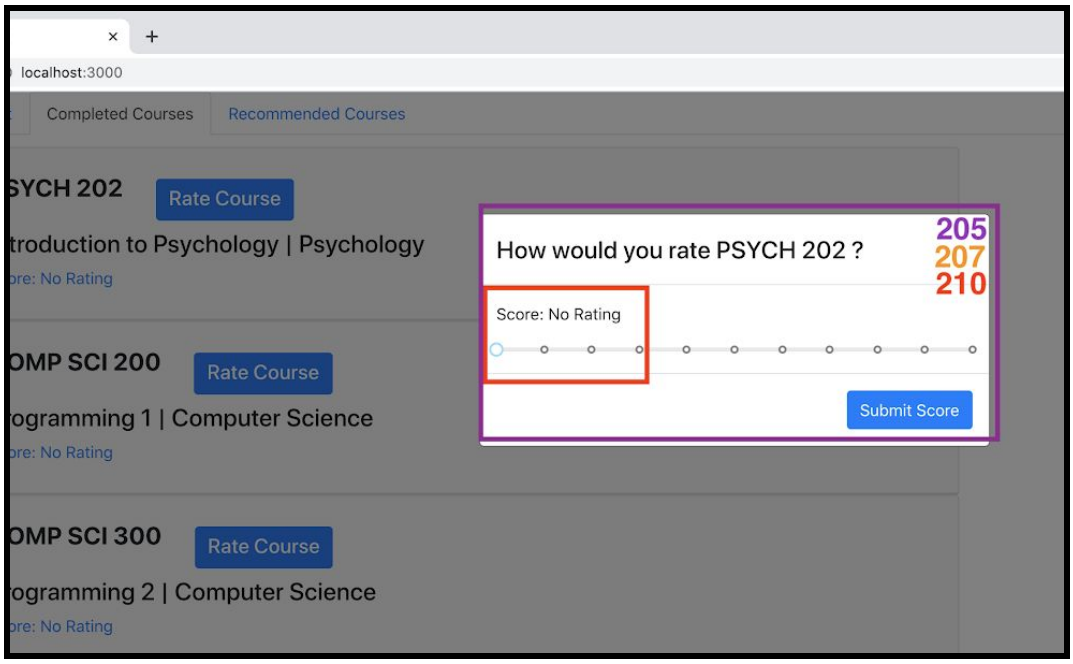
Focusing on your components, inspect your design, considering each usability heuristic, for any violations of the heuristics. For each violation, use the following table to briefly describe the violation and give it a unique number (specified in the # column). Make copies of your screenshots from Step 1, focusing on the design elements you are considering in this step, and mark them with the unique numbers so that the reader of your report can find the location of the violation in the screenshots and read your description in the table below. In addition, color-code the violations for severity, highlighting with red, orange, yellow, green, and gray for the severity-rating scale we covered in class (with red being most severe to gray being a non-issue).

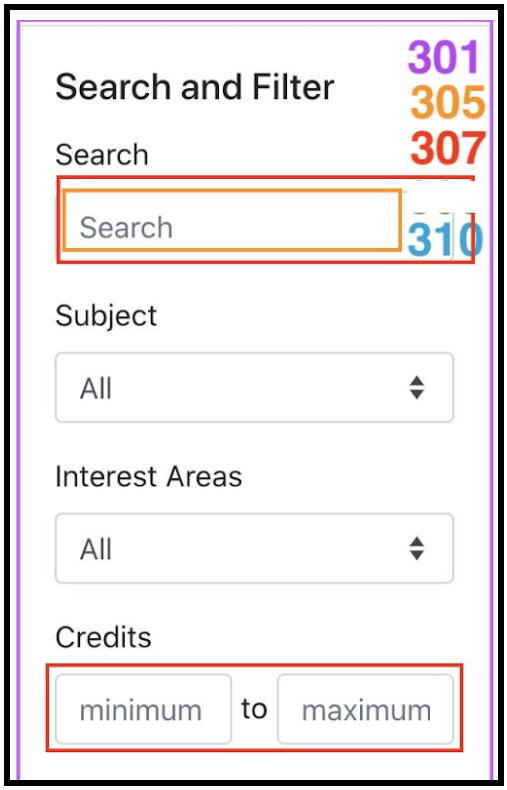
Heuristic	#	Component 1	#	Component 2	#	Component 3
Visibility of system status						301 no guideline shows after user input or select
Match between real world & system						

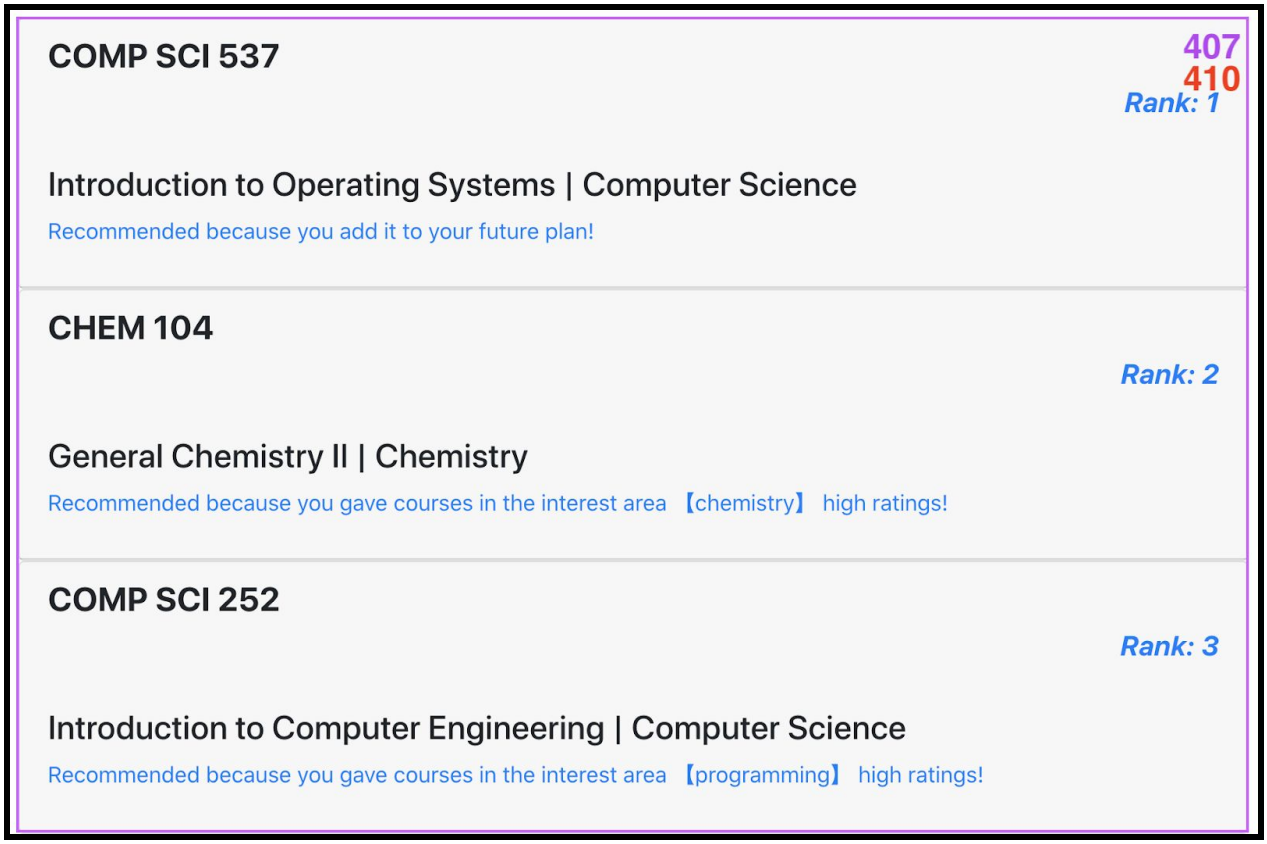
User control & freedom	103 cannot undo after add course action		
Consistency & standards			
Error prevention		205 show no rating if user doesn't move and leave slider at zero index	305 No auto prediction of user search input
Recognition rather than recall			
Flexibility & efficiency of use	107 No shortcut	207 No shortcut	307 No shortcut
Aesthetic & minimalist design			
Help users with errors			
Help & documentation	110 No documentation	210 No documentation	310 No documentation

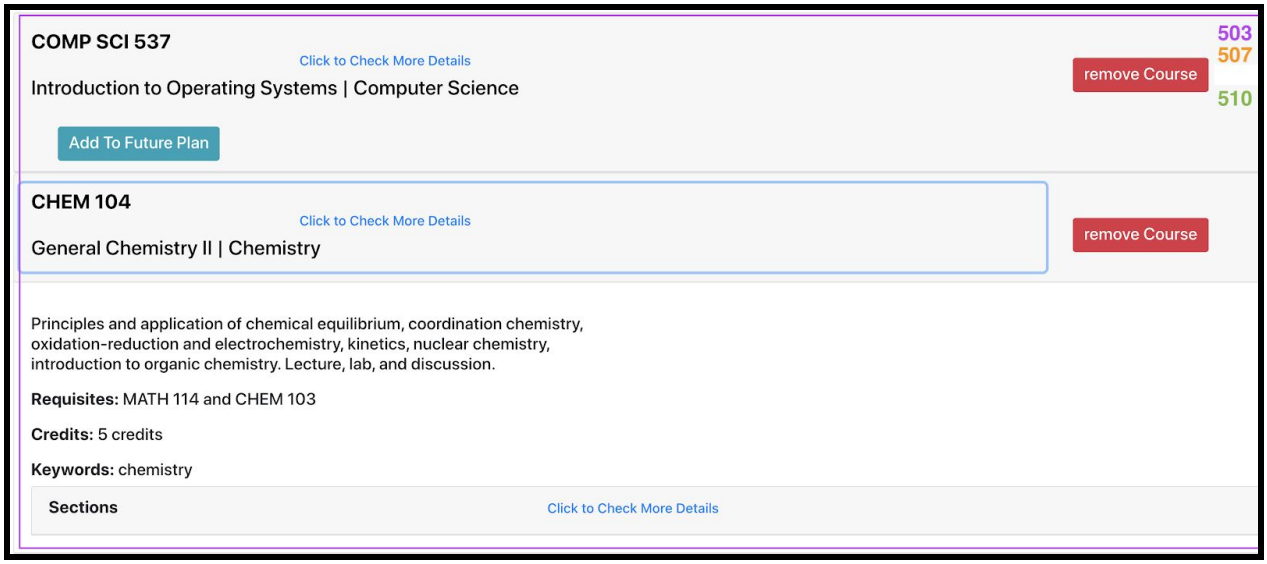
Heuristic	**#**	**Component 4**	**#**	**Component 5**	**#**	**Component 6**
Visibility of system status						
Match between real world & system						
User control & freedom				503 cannot undo after remove course action		
Consistency & standards						
Error prevention						
Recognition rather than recall						
Flexibility & efficiency of use		407 No shortcut		507 No shortcut		
Aesthetic & minimalist design						
Help users with errors						
Help & documentation		410 No documentation		510 No documentation		


PSYCH 202
Click to Check More Details
Introduction to Psychology | Psychology
Add Course
COMP SCI 537
Click to Check More Details
Introduction to Operating Systems | Computer Science
Add Course
103
107
110
Input-output hardware, interrupt handling, properties of magnetic tapes, discs and drums, associative memories and virtual address translation techniques. Batch processing, time sharing and real-time systems, scheduling resource allocation, modular software systems, performance measurement and system evaluation.
Requisites: (COMP SCI 354 or COMP SCI 400)
Credits: 4 credits
Keywords: computer, science, operating, system, systems
Sections
Click to Check More Details


localhost:3000
Completed Courses
Recommended Courses
SYCH 202
Rate Course
troduction to Psychology | Psychology
re: No Rating
OMP SCI 200
Rate Course
ogramming 1 | Computer Science
re: No Rating
OMP SCI 300
Rate Course
ogramming 2 | Computer Science
re: No Rating
How would you rate PSYCH 202 ?
205
207
210
Score: No Rating
Submit Score


Search and Filter
301
305
307
Search
Search
310
Subject
All
Interest Areas
All
Credits
minimum
to
maximum


COMP SCI 537
407
410
Rank: 1
Introduction to Operating Systems | Computer Science
Recommended because you add it to your future plan!
CHEM 104
Rank: 2
General Chemistry II | Chemistry
Recommended because you gave courses in the interest area 【chemistry】 high ratings!
COMP SCI 252
Rank: 3
Introduction to Computer Engineering | Computer Science
Recommended because you gave courses in the interest area 【programming】 high ratings!



Step 4. Develop Design Recommendations. (0.4 Points)

For each violation you identified in the previous step, provide a design recommendation for addressing it along with an indication of whether or not it is feasible to implement the recommendation as an extension of your *React 2* α deliverable. (Only recommendations that are beyond the capabilities we learned in class or beyond the scope of the project should be marked as not being feasible.) Order the table of recommendations based on the severity of the usability problem from most severe to least severe. Use the table below to describe your recommendations, adding additional rows as needed, and follow the same color-coding from the previous step for severity ratings.

#	Recommendation	Feasibility (Yes/No)
	103 users can select undo after add course action	Yes
	107 add shortcut to select and add course and select search tab as well as course components	Yes
	110 add documentation of how to add courses	Yes
	205 show rate of zero if user doesn't move and leave slider at zero index	Yes
	207 add shortcut to select and rate course and select rate tab	Yes
	210 add documentation of how to rate a taken course	Yes
	301 tells user search done and check right side after inputs and selections	Yes
	305 Auto predicting search words	Yes
	307 add shortcut to select inputs and enter number or keywords and select search component	Yes

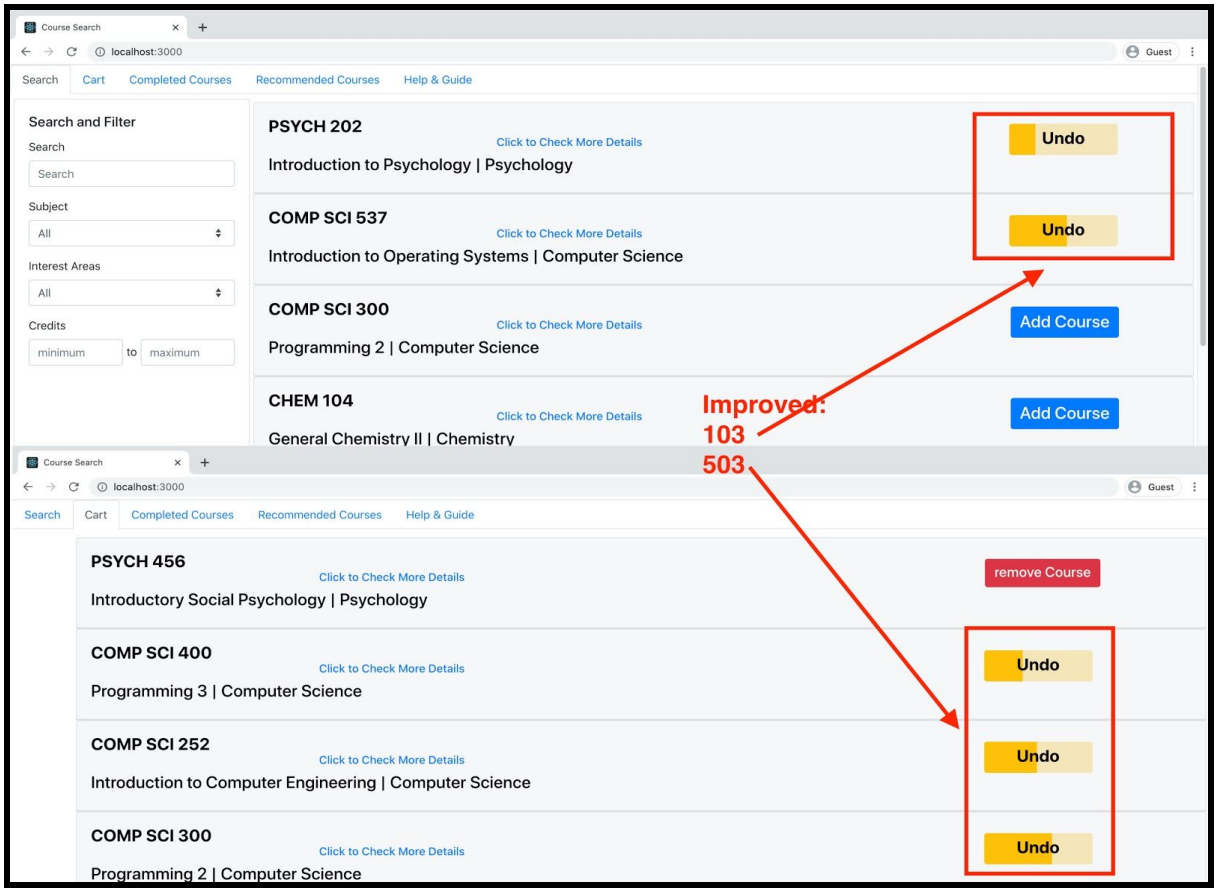
310 add documentation of how to search courses	Yes
407 add shortcut to select recommendation tab	Yes
410 add documentation of how does the course recommendation works	Yes
503 users can select undo after remove course action	Yes
507 add shortcut to select and remove course and select recommendation tab	Yes
510 add documentation of how to remove courses	Yes

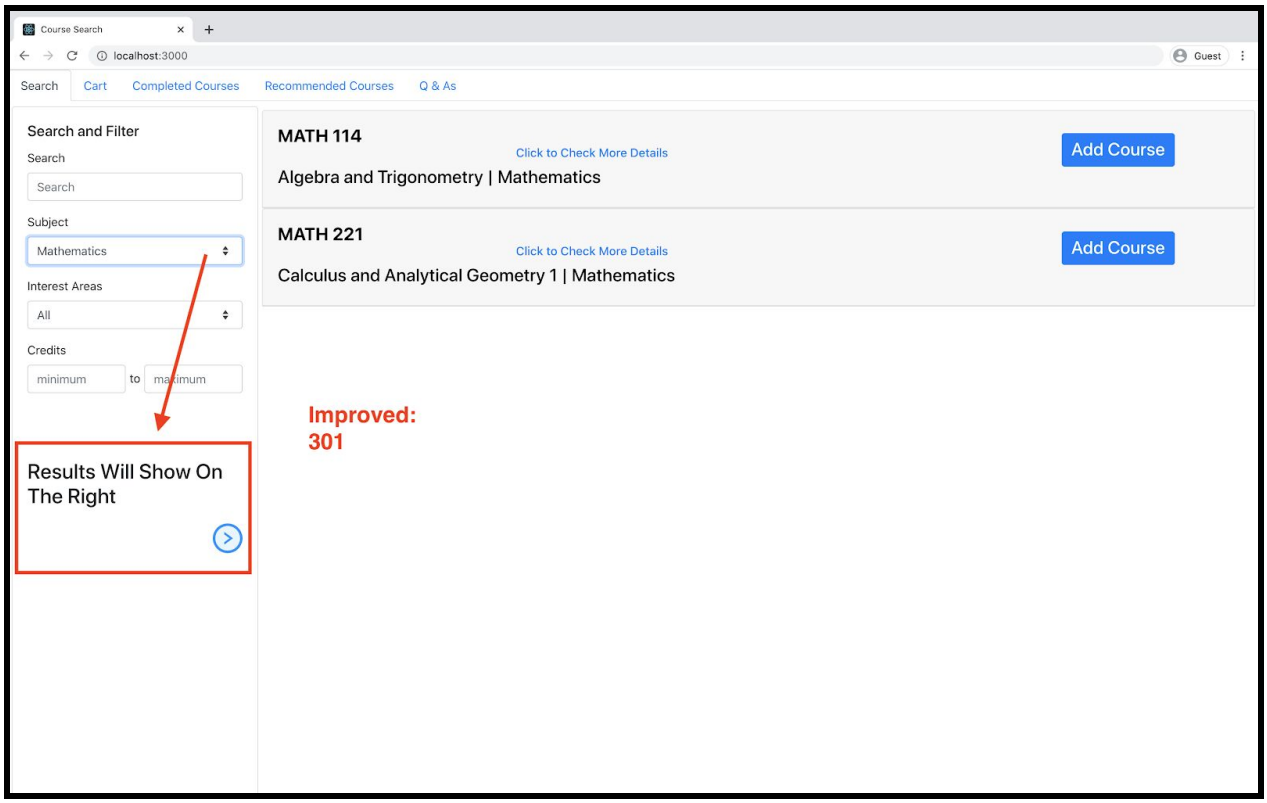
Step 5. Implement Your Recommendations. (1.4 Points)

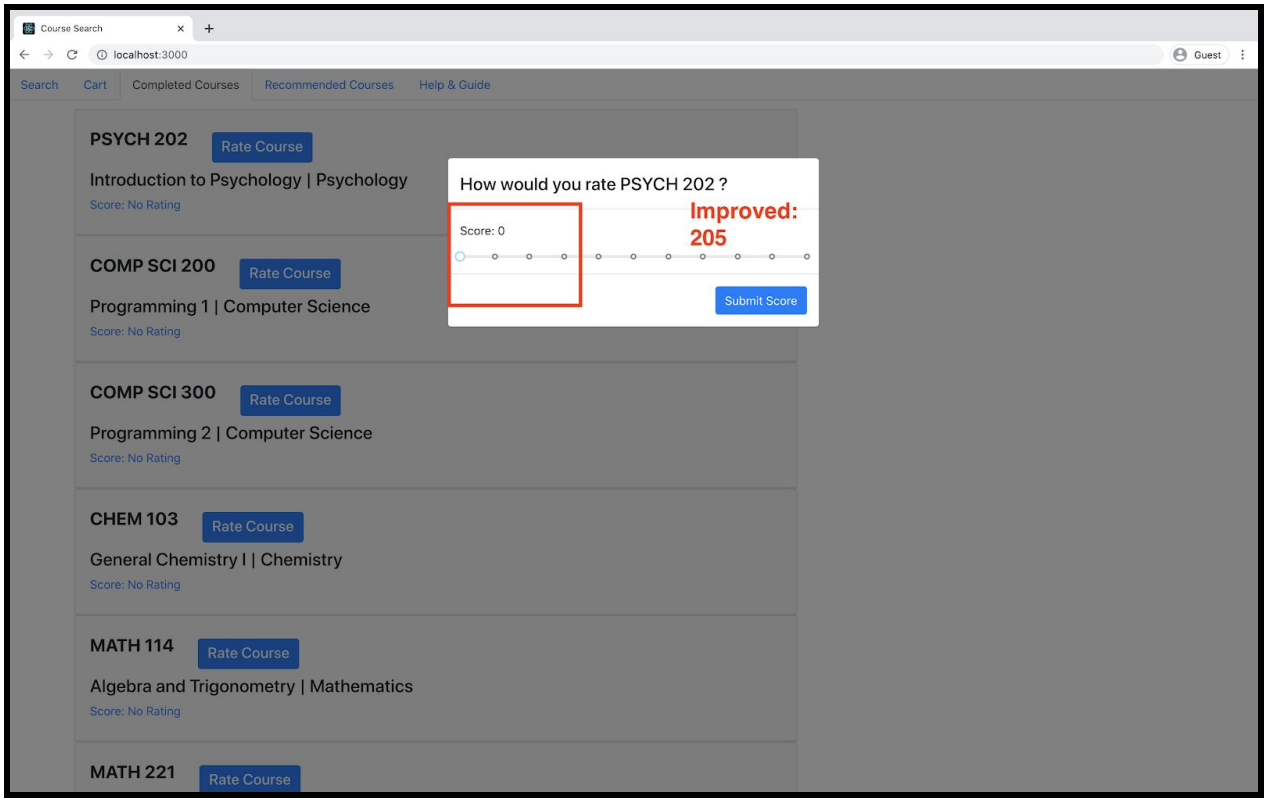
In this step, you will implement the design recommendations that you identified as "feasible" in the previous step in your prototype, updating your design. To receive full points, you will implement at least three design recommendations that can improve one or more of the components you focused on. Submit your improved React project based on instructions below and provide a paragraph that summarizes the outcome of the heuristic evaluation. In this paragraph, reflect on how your design improved, what you learned about usability in the process of applying the heuristics, and whether you gained any unexpected insights about your design.

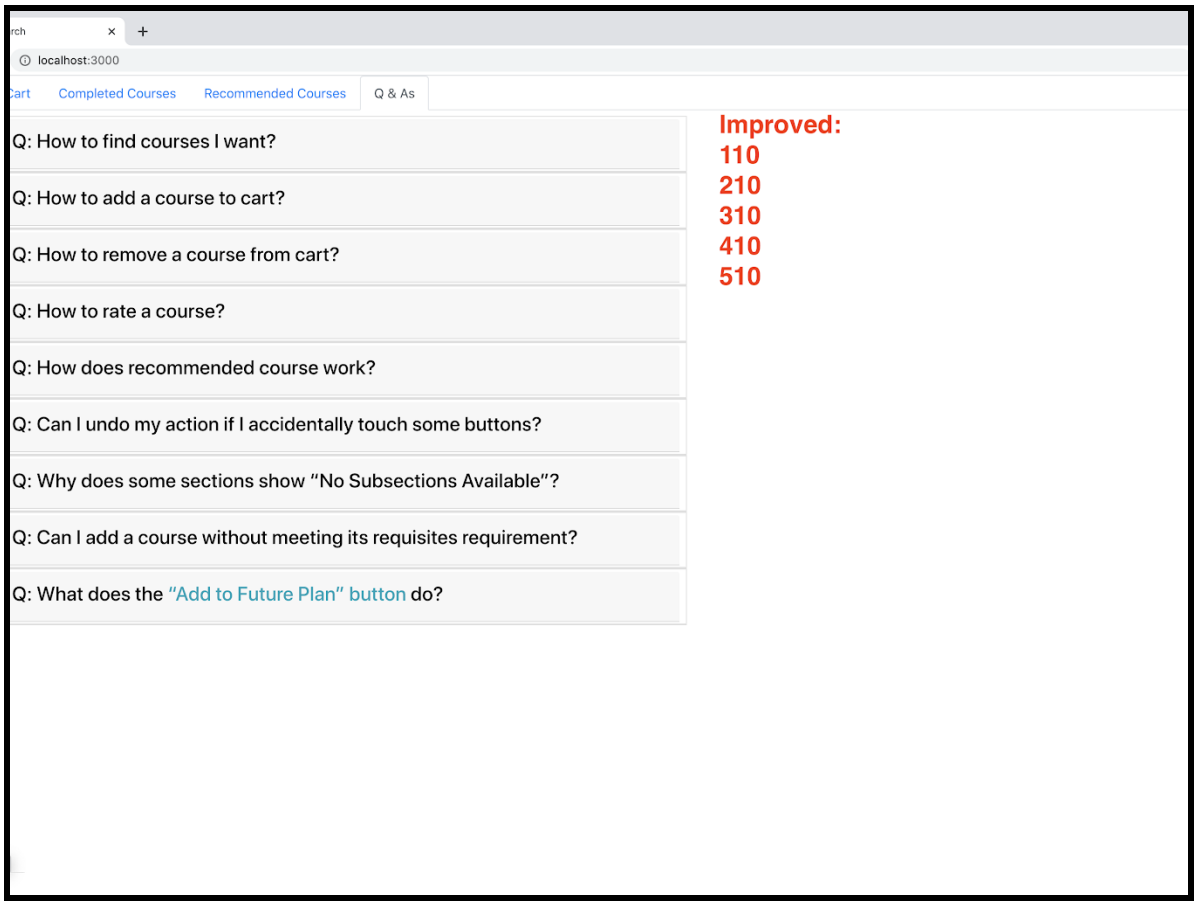
Your deliverable will be a completed version of this document, attached to the canvas assignment as a PDF, and the GitHub Classroom repository name and latest commit hash.

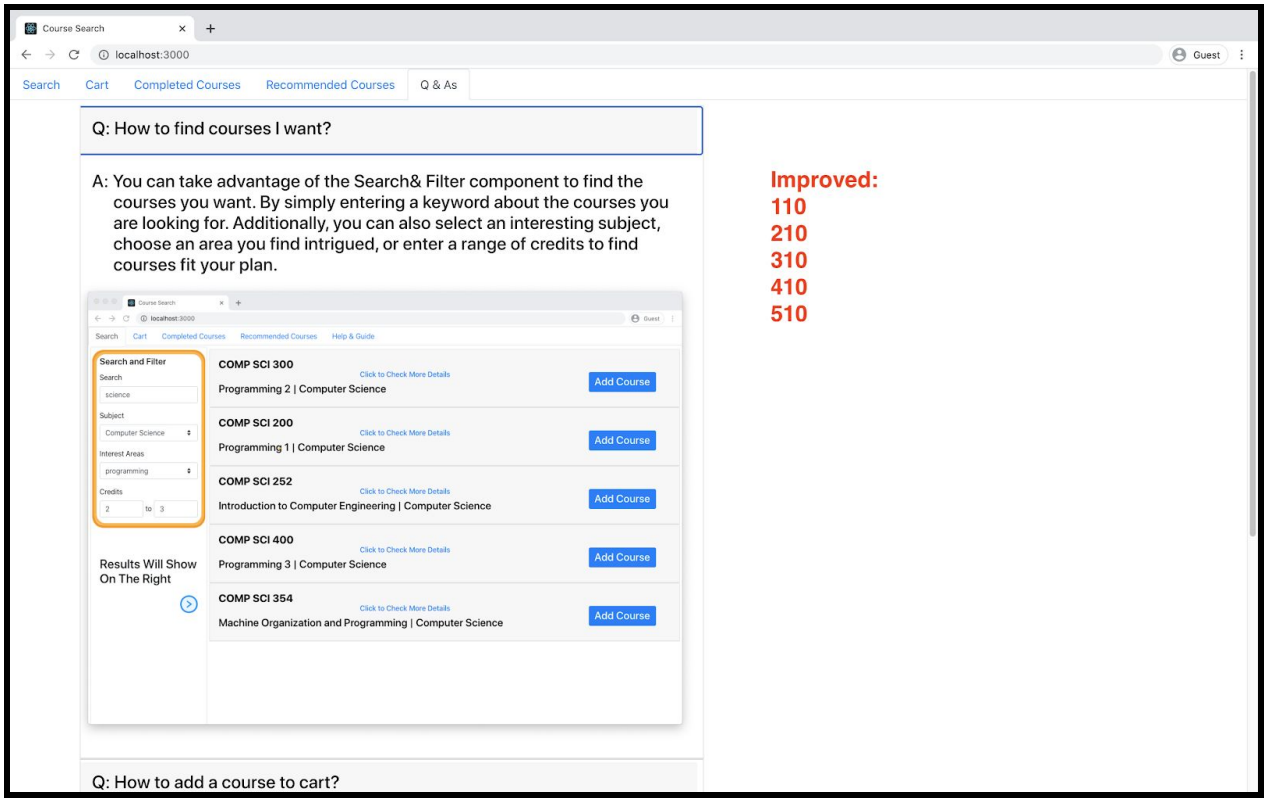
I improved 9 recommendations mentioned in Step 4, which are the 103, 503(*add, remove button undo*), 205 (*show rate of zero if user doesn't move and leave slider at zero index*), 301(*tells user search done and check right side after inputs and selections*), 110, 210, 310, 410, and 510(*add documentation of how to add courses, how to rate a taken course, how to search courses, how does the course recommendation works, and how to remove courses*). My design now can provide users with the ability to undo mistakenly add/remove courses operations, improving its User control & freedom heuristic. Moreover, my design is equipped with a Q&A style documentation with annotated screenshots, which tells users how each function works in this application, improving its Help & documentation heuristic. Furthermore, its rating component's Error prevention heuristic and its search & filter component's Visibility of system status heuristic are also improved. From the process of applying these heuristics, I learned that there are many aspects I missed to think about, which may create obstacles for my users to understand and use my application. For example, documentation is necessary since designers can be biased when using their apps, and they can seem some hard-to-understand functionalities to users as obvious. Additionally, in this experience, I found heuristic very helpful to guide me through points I can miss. Throughout the process of reviewing my design, I surprisingly noticed that the Visibility of system status heuristic of my design is not thoughtful as I thought. In the future, I should pay more attention to the Visibility of system status heuristic of my design by using more elements to show users how their actions have interacted with and affected the application.


Course Search
localhost:3000
Guest
Search
Cart
Completed Courses
Recommended Courses
Help & Guide
Search and Filter
Search
Search
Subject
All
Interest Areas
All
Credits
minimum
to
maximum
PSYCH 202
Click to Check More Details
Introduction to Psychology | Psychology
Undo
COMP SCI 537
Click to Check More Details
Introduction to Operating Systems | Computer Science
Undo
COMP SCI 300
Click to Check More Details
Programming 2 | Computer Science
Add Course
CHEM 104
Click to Check More Details
General Chemistry II | Chemistry
Add Course
Improved:
103
503
Course Search
localhost:3000
Guest
Search
Cart
Completed Courses
Recommended Courses
Help & Guide
PSYCH 456
Click to Check More Details
Introductory Social Psychology | Psychology
remove Course
COMP SCI 400
Click to Check More Details
Programming 3 | Computer Science
Undo
COMP SCI 252
Click to Check More Details
Introduction to Computer Engineering | Computer Science
Undo
COMP SCI 300
Click to Check More Details
Programming 2 | Computer Science
Undo


Course Search
localhost:3000
Guest
Search
Cart
Completed Courses
Recommended Courses
Q & As
Search and Filter
Search
Search
Subject
Mathematics
Interest Areas
All
Credits
minimum
to
maximum
Results Will Show On The Right
MATH 114
Click to Check More Details
Algebra and Trigonometry | Mathematics
Add Course
MATH 221
Click to Check More Details
Calculus and Analytical Geometry 1 | Mathematics
Add Course
Improved:
301


Course Search
localhost:3000
Guest
Search
Cart
Completed Courses
Recommended Courses
Help & Guide
PSYCH 202
Rate Course
Introduction to Psychology | Psychology
Score: No Rating
COMP SCI 200
Rate Course
Programming 1 | Computer Science
Score: No Rating
COMP SCI 300
Rate Course
Programming 2 | Computer Science
Score: No Rating
CHEM 103
Rate Course
General Chemistry I | Chemistry
Score: No Rating
MATH 114
Rate Course
Algebra and Trigonometry | Mathematics
Score: No Rating
MATH 221
Rate Course
How would you rate PSYCH 202 ?
Score: 0
Submit Score
Improved:
205


localhost:3000
Completed Courses
Recommended Courses
Q & As
Q: How to find courses I want?
Q: How to add a course to cart?
Q: How to remove a course from cart?
Q: How to rate a course?
Q: How does recommended course work?
Q: Can I undo my action if I accidentally touch some buttons?
Q: Why does some sections show "No Subsections Available"?
Q: Can I add a course without meeting its requisites requirement?
Q: What does the "Add to Future Plan" button do?
Improved:
110
210
310
410
510



Improved:
110
210
310
410
510