



React 1 (3 Points)

Reimagining Layout, Structure, & Navigation

In this assignment, you will explore the concepts we learned in the lecture, titled “*Interaction Design: Structure, Layout, & Navigation*.”

Part 1—Analysis: In this part, you will analyze your current solution for the *React 1 * Assignment in terms of its structural, layout, and navigational elements.

Part 2—Redesign: This part will involve using the principles and components covered in class to redesign your solution and describe your design choices.

Part 3—Implementation: In this part, you will implement your new design by extending your implementation for the *React 1 * Assignment using additional React and/or Bootstrap components.

Submission Details

[GitHub Classroom Starter Code](#)

React 1 β will build on your implementation of React 1 α . You should copy your code from your React 1 α project to the React 1 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 react1-beta repository, not react1-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. react1-beta-ctnelson1997, 2b0ef83.

Part 1: Analysis (0.6 Points)

(0.2 Points) **Step 1. Analyze Structures.** What kind of structure(s) (e.g., “Show one single thing”) can you identify in your *React 1* implementation? Does it follow a single structure or combine structures? Take a screenshot of your implementation and annotate it to point at the structure(s) you identify, briefly (2-3 sentences) describing them and explaining why parts or all of your implementation follow these structures.

The screenshot shows a web application for a course catalog. On the left is a 'Search and Filter' sidebar with fields for 'Search', 'Subject' (set to 'All'), and 'Credits' (with 'minimum' and 'maximum' input boxes). The main content area displays a list of courses. At the top, two course cards are visible: 'PSYCH 202: Introduction to Psychology' and 'COMP SCI 537: Introduction to Operating Systems'. Below these is a table with columns: Section, Instructor, Location, Time, and an action column. The table lists several sections, including 'LEC_001' by Andrea Arpaci-Dusseau and several 'DIS' sections. Each row has an 'Add' button (e.g., 'Add Lecture', 'Add Discussion'). Below the table, there is a 'Credits: 4' section, a description of the course content, prerequisites, keywords, and the subject 'Computer Science'. At the bottom, two more course cards are visible: 'COMP SCI 300: Programming 2' and 'CHEM 104: General Chemistry II'. Handwritten blue annotations are present: a bracket on the left sidebar with the text 'show a list of things.', a bracket on the top course cards with the text 'show a list of things', and an arrow pointing to the 'Add Lecture' button with the text 'Facilitating a task'.

Search Cart

Search and Filter

Search

Search

Subject

All

Credits

minimum to maximum

PSYCH 202: Introduction to Psychology

COMP SCI 537: Introduction to Operating Systems

Section	Instructor	Location	Time	
LEC_001	Andrea Arpaci-Dusseau	1125 DeLuca Biochemistry Building	Thursday: 11:00am - 12:15pm; Tuesday: 11:00am - 12:15pm;	Add Lecture
-- DIS_301		2317 Engineering Hall	Wednesday: 11:00am - 11:50am;	Add Discussion
-- DIS_302		1325 Computer Sciences and Statistics	Wednesday: 12:05pm - 12:55pm;	Add Discussion
-- DIS_303		1325 Computer Sciences and Statistics	Wednesday: 1:20pm - 2:10pm;	Add Discussion
-- DIS_304		2255 Engineering Hall	Wednesday: 3:30pm - 4:20pm;	Add Discussion
-- DIS_305		1325 Computer Sciences and Statistics	Wednesday: 4:15pm - 5:25pm;	Add Discussion

Credits: 4

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.

Prerequisites: (COMP SCI 354 OR COMP SCI 400)

Keywords: computer, science, operating, system, systems

Subject: Computer Science

COMP SCI 300: Programming 2

CHEM 104: General Chemistry II

show a list of things.

show a list of things

Facilitating a task

Search

Cart

Welcome to your cart

You can click the sections below to remove the section from your cart, or the Remove All button to remove the entire course.

COMP SCI 537 :
Introduction to Operating Systems

LEC_001

Thursday: 11:00am - 12:15pm;
Tuesday: 11:00am - 12:15pm;
1125 DeLuca Biochemistry Building

DIS_301

Wednesday: 11:00am - 11:50am;
2317 Engineering Hall

DIS_302

Wednesday: 12:05pm - 12:55pm;
1325 Computer Sciences and Statistics

DIS_303

Wednesday: 1:20pm - 2:10pm;
1325 Computer Sciences and Statistics

DIS_304

COMP SCI 300 :
Programming 2

LEC_001

Thursday: 2:30pm - 3:45pm;
Tuesday: 2:30pm - 3:45pm;
AB20 Weeks Hall for Geological Sciences

LEC_002

Thursday: 1:00pm - 2:15pm;
Tuesday: 1:00pm - 2:15pm;
132 Noland Hall

LEC_003

Friday: 11:00am - 11:50pm;
Monday: 11:00am - 11:50pm;
Wednesday: 11:00am - 11:50pm;
AB20 Weeks Hall for Geological Sciences

LEC_004

Friday: 2:25pm - 3:15pm; Monday:
2:25pm - 3:15pm; Wednesday:
2:25pm - 3:15pm;

BIOLOGY 101 : Animal Biology

LEC_001

Friday: 11:00am - 11:50am;
Monday: 11:00am - 11:50am;
Wednesday: 11:00am - 11:50am;
272 Bascom Hall

LEC_002

Friday: 12:05pm - 12:55pm;
Monday: 12:05pm - 12:55pm;
Wednesday: 12:05pm - 12:55pm;
272 Bascom Hall

Remove All

show a list of things

Facilitating a task (delete)

1. Show a list of things (courses) – There are a list of courses user can choose from. Showing a list of these courses allow users to easily glance through the courses that they want to choose. This is to easily let the users glance through the courses they want without much context switching.
2. Show a list of things (sections) – Enable users to glance through the many “options” of the course they want to select. Allow the user to choose a section they want in particular course at a glance without much context switching.
3. Facilitating a task – The buttons are example of facilitating a task, and it allows user to add/remove these courses to the cart.

(0.2 Points) **Step 2. Analyze Layout.** Describe the current layout of your *React 1* implementation, identifying what principles of layout design (e.g., golden proportion, visual hierarchy, visual scan patterns) it currently follows (at least two principles). Use the same (unannotated) screenshot from Step 1 and draw or annotate the principles you identified. Either make additional copies of the screenshots or use different colors for multiple principles.

The screenshot shows a web application for a course catalog. It features a search and filter sidebar on the left, a main content area with a list of courses, and a detailed view of a specific course at the bottom. Handwritten blue annotations highlight three layout principles:

- Gestalt's Theory (Similarity):** An arrow points to the 'Add All' buttons for the top two courses, with the note: "Gestalt's Theory (similarity) - user would know each button behaves the same."
- Creating visual hierarchy:** An arrow points to the 'LEC_001' section header in the course list, with the note: "Creating visual hierarchy".
- Effectively using Grid:** A large arrow points from the left sidebar to the main content area, with the note: "Effectively using Grid".

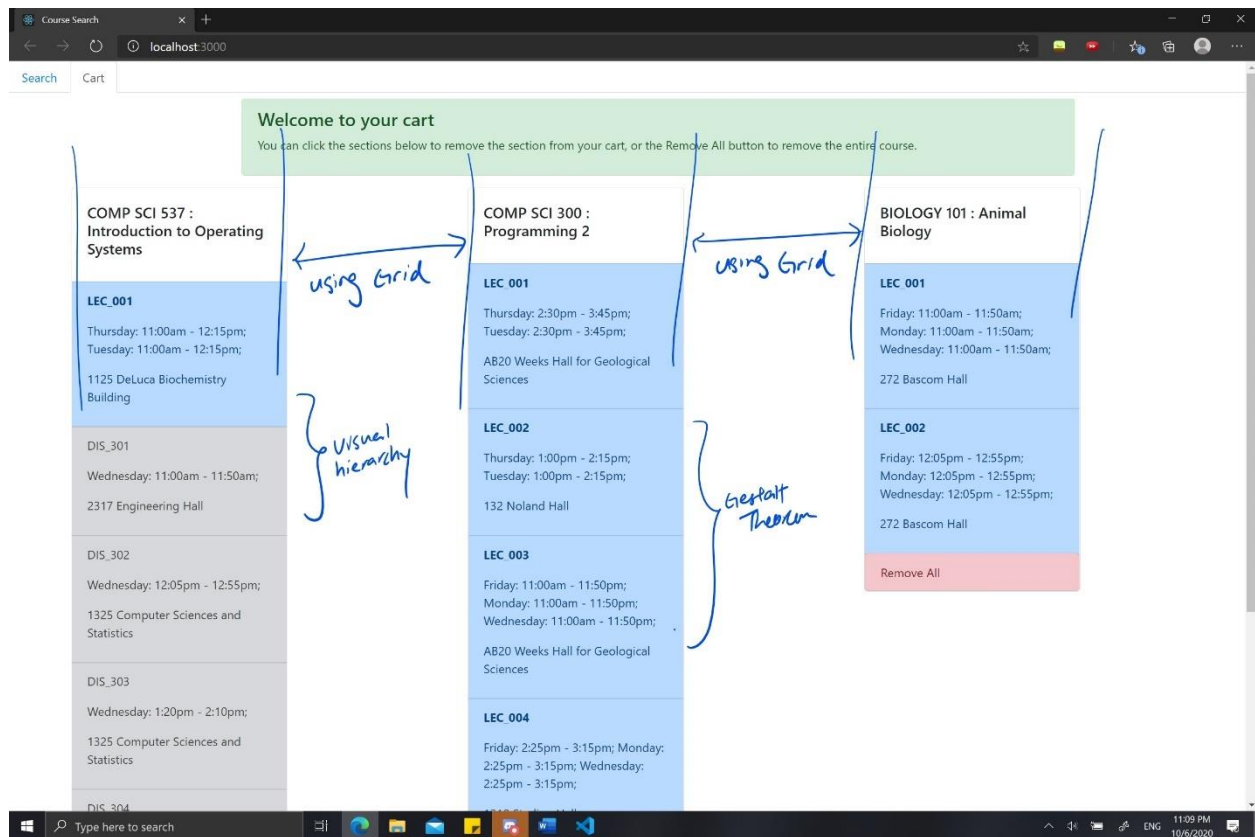
Section	Instructor	Location	Time	Action
LEC_001	Andrea Arpaci-Dusseau	1125 DeLuca Biochemistry Building	Thursday: 11:00am - 12:15pm; Tuesday: 11:00am - 12:15pm;	Add Lecture
-- DIS_301		2317 Engineering Hall	Wednesday: 11:00am - 11:50am;	Add Discussion
-- DIS_302		1325 Computer Sciences and Statistics	Wednesday: 12:05pm - 12:55pm;	Add Discussion
-- DIS_303		1325 Computer Sciences and Statistics	Wednesday: 1:20pm - 2:10pm;	Add Discussion
-- DIS_304		2255 Engineering Hall	Wednesday: 3:30pm - 4:20pm;	Add Discussion
-- DIS_305		1325 Computer Sciences and Statistics	Wednesday: 4:15pm - 5:25pm;	Add Discussion

Credits: 4
 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.
 Prerequisites: (COMP SCI 354 OR COMP SCI 400)
 Keywords: computer, science, operating, system, systems
 Subject: Computer Science

COMP SCI 300: Programming 2 Add All

CHEM 104: General Chemistry II Add All

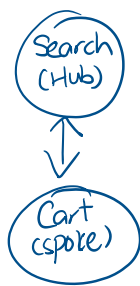
1. Gestalt's Theory (Similarity): User would know that clicking another courses would expand the detailed options since all buttons looked the same.
2. Creating visual hierarchy: The section is bolded, and subsections are not.
3. Effectively using grids: Search and filter panel used grid to better use the SPA space.



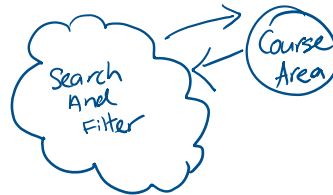
1. Gestalt Theorem: User can easily know elements by look at the color (blue are lectures and greys are discussions).
2. Using Grid: to easily glance through many sections and create organized aesthetic.
3. Visual hierarchy: Different colors (blue, grey) creates hierarchy of the sections (lectures > discussions).

(0.2 Points) **Step 3. Analyze Navigation.** Consider your *React 1* implementation, what navigation model(s) does it use? Below, draw the navigation model that your implementation follows the same way navigation models were described in class.

1. Hub and spoke



2. Flat navigation



Part 2: Redesign (0.8 Points)

(0.4 Points) **Step 1. Conceptual Redesign.** In this step, you will reimagine your *React 1* implementation, such that it uses a different set of structures, navigation models, and/or principles of layout design. Your goal should not be to change your implementation for the sake of changing it, but consider ways in which the structures and layout and navigation principles might improve your implementation. Your conceptual redesign should involve the use of at least one layout principle, make at least one change in the navigation model, and introduce at least one element/aid to improve navigation. The use of additional or a different set of structures is optional. Provide a hand-drawn or digitally created (e.g., in Adobe XD) mock-up of your design below. Annotate your design to describe your design choices, highlighting the specific principles you employed.

Course Search & Enroll

Credits

COMP SCI 639 - Building UI 3 Cr.

COMP SCI 200 - Programming I 3 Cr.

Introduces students to Object-Oriented Programming using classes and objects to solve more complex problems. Introduces array-based and linked data structures: including lists, stacks, and queues. Programming assignments require writing and developing multi-class (file) programs using interfaces, generics, and exception handling to solve challenging real world problems. Topics reviewed include reading/writing data and objects from/to files and exception handling and command line arguments. Topics introduced: object-oriented design: class vs. object; create and define interfaces and iterators; searching and sorting; abstract data types (List, Stack, Queue, PriorityQueue, Heap); Binary Search Tree; generic interfaces (parametric polymorphism); how to design and write test methods and classes; array based vs. linked node implementations; introduction to complexity analysis; recursion.

Prerequisites: (COMP SCI 200)

Keywords: computer, science, programming, java

Subject: Computer Science

Section	Instructor	Location	Time
LEC 001	Jim Williams	132 Noland Hall	Thursday: 8:00am - 9:15am; Tuesday: 8:00am - 9:15am;
LAB 311		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 312		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 313		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 314		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LEC 002	Jim Williams	132 Noland Hall	Thursday: 11:00am - 12:15pm; Tuesday: 11:00am - 12:15pm;
LAB 321		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 323		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 325		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;

CHEM 103 - General Chemistry I 3 Cr.

Change to flat navigation

Show "one thing" (A cart)

Show a list of things

Cart

CS 639 - Building UI

LEC 001
Monday 4:00 - 5:00pm
Wednesday 4:00 - 5:00pm

CS 200 - Programming I

LEC 001
Thursday 8:00 - 9:15am
Tuesday 8:00 - 9:15am

LAB 311
Wednesday 9:30 - 10:45am

LAB 314
Wednesday 2:30 - 3:45pm

LEC 001
Thursday 8:00 - 9:15am
Tuesday 8:00 - 9:15am

LAB 311
Wednesday 9:30 - 10:45am

LAB 314
Wednesday 2:30 - 3:45pm

CHEM 103 - General Chemistry I

LEC 001
Thursday 8:00 - 9:15am
Tuesday 8:00 - 9:15am

LAB 311
Wednesday 9:30 - 10:45am

LAB 314
Wednesday 2:30 - 3:45pm

Show a list of things

navigation aid, group bunch of similar data to a table for ease of navigation

(0.4 Points) **Step 2. Detailed Redesign.** In this step, you will build on your mock-up to create a detailed design, determining image, color (for background and elements), type, size, icons, and so on (as we also did, to some extent, in the Javascript β Assignment). Provide a digitally created mock-up (e.g., in Adobe XD) that shows your design choices. Annotate your mock-up to describe your design choices.

Course Search & Enroll (Franklin Gothic, 35)

shapes as text field

Keywords Subject Credits

Franklin Gothic, 20

COMP SCI 639 - Building UI 3 Cr.

COMP SCI 200 - Programming I 3 Cr.

Introduces students to Object-Oriented Programming using classes and objects to solve more complex problems. Introduces array-based and linked data structures including lists, stacks, and queues. Programming assignments require writing and developing multi-class (file) programs using interfaces, generics, and exception handling to solve challenging real world problems. Topics reviewed include reading/writing data and objects from/to files and exception handling, and command line arguments. Topics introduced: object-oriented design; class vs. object; create and define interfaces and iterators; searching and sorting; abstract data types (List, Stack, Queue, Priority Queue, Heap); Binary Search Trees; generic interfaces (parametric polymorphism); how to design and write test methods and classes; array based vs. linked node implementations; introduction to complexity analysis; recursion.

Prerequisites: (COMP SCI 200)

Keywords: computer, science, programming, java

Subject: Computer Science

Section	Instructor	Location	Time
LEC 001	Jim Williams	132 Noland Hall	Thursday: 8:00am - 9:15am; Tuesday: 8:00am - 9:15am;
LAB 311		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 312		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 313		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 314		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LEC 002	Jim Williams	132 Noland Hall	Thursday: 11:00am - 12:15pm; Tuesday: 11:00am - 12:15pm;
LAB 321		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 323		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;
LAB 325		1350 Computer Sciences and Statistics	Wednesday: 9:30am - 10:45am;

Franklin Gothic, 15

Franklin Gothic Book, 15

CHEM 103 - General Chemistry I 3 Cr.

Cart

Franklin Gothic, 30

Franklin Gothic, 20

Franklin Gothic, 15

Franklin Gothic Book, 12

Colors to convert hierarchy and use of Gestalt Theory

CS 639 - Building UI

LEC 001
Monday 4:00 - 5:00pm
Tuesday 4:00 - 5:00pm

CS 200 - Programming I

LEC 001
Thursday 8:00 - 9:15am
Tuesday 8:00 - 9:15am

LAB 311
Wednesday 9:30 - 10:45am

LAB 314
Wednesday 2:30 - 3:45pm

LEC 001
Thursday 8:00 - 9:15am
Tuesday 8:00 - 9:15am

LAB 311
Wednesday 9:30 - 10:45am

LAB 314
Wednesday 2:30 - 3:45pm

CHEM 103 - General Chemistry I

LEC 001
Thursday 8:00 - 9:15am
Tuesday 8:00 - 9:15am

LAB 311
Wednesday 9:30 - 10:45am

LAB 314
Wednesday 2:30 - 3:45pm

LEC 001
Thursday 8:00 - 9:15am
Tuesday 8:00 - 9:15am

1. Font – Franklin Gothic for bold/emphasized text, and Franklin Gothic Book for normal text
2. Font size: Depends on the hierarchy which follows (Title, course name, sections, details)
3. Black and white color scheme for course (unity)
4. Clickable items are elevated to show affordances and direct user attention.
5. Contrast (of hierarchy) are created using font-weight.

Part 3: Implementation (1.6 Points)

(0.3 Points) **Step 1. Inspect Library Elements.** In this step, you will inspect the standard React component library, the [Bootstrap](#) component library, and/or an alternative that you are comfortable working with to see how you can realize detailed design you created in the previous part using these components. You are not expected to change the library components to exactly match your design choices, but to identify which component elements might best meet your design goals. Below, copy the design and the choices you generated in Part 2 and annotate them to describe which components from the library you will use to accomplish your design goals.

(1.3 Points) **Step 2. Implement Redesign.** The last step of this part will involve implementing the design improvements you described in Part 2, using the layout and components you described in the previous step. You can use standard React components, Bootstrap components, and/or an alternative library in your implementation. You do not have to implement new *functionality*; focus on implementing 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.

Search & Enroll Search Subjects Interests Credits

Subjects: All, Psychology, Computer Science, Chemistry, Mathematics, Biology, Statistics

Course List:

- PSYCH 202: Introduction to Psychology 3 Cr.
- COMP SCI 537: Introduction to Operating Systems 4 Cr.
Input-output hardware, interrupt handling, properties of magnetic tapes, discs and d...
Batch processing, time sharing and real-time systems, scheduling resource allocation...
evaluation, ...
ress translation techniques, ...
measurement and system...
Pre-requisites: (COMP SCI 354 OR COMP SCI 400)
- COMP SCI 300: Programming 2 3 Cr.
- CHEM 104: General Chemistry II 5 Cr.
- COMP SCI 200: Programming 1 3 Cr.
- MATH 114: Algebra and Trigonometry 5 Cr.
- PSYCH 456: Introductory Social Psychology 4 Cr.
- COMP SCI 252: Introduction to Computer Engineering 2 Cr.

Table for COMP SCI 537:

Section	Instructor	Location	Time
LEC 001	Andrea Arpaci-Dusseau	1125 DeLuca Biochemistry Building	Thursday: 11:00am - 12:15pm Tuesday: 11:00am - 12:15pm
DIS 301		2317 Engineering Hall	Wednesday: 11:00am - 11:50am
DIS 302		1325 Computer Sciences and Statistics	Wednesday: 12:05pm - 12:55pm
DIS 303		1325 Computer Sciences and Statistics	Wednesday: 1:20pm - 2:10pm
DIS 304		2255 Engineering Hall	Wednesday: 3:30pm - 4:20pm
DIS 305		1325 Computer Sciences and Statistics	Wednesday: 4:15pm - 5:25pm

Cart:

- COMP SCI 537: Introduction to Operating Systems
- LEC 001
1125 DeLuca Biochemistry Building
Thursday: 11:00am - 12:15pm
Tuesday: 11:00am - 12:15pm
- DIS 301
2317 Engineering Hall
Wednesday: 11:00am - 11:50am
- DIS 302
1325 Computer Sciences and Statistics
Wednesday: 12:05pm - 12:55pm
- DIS 303
1325 Computer Sciences and Statistics
Wednesday: 1:20pm - 2:10pm
- DIS 304
2255 Engineering Hall
Wednesday: 3:30pm - 4:20pm
- DIS 305
1325 Computer Sciences and Statistics
Wednesday: 4:15pm - 5:25pm

Prerequisite Warning: Prerequisite not met. You have not met the pre-requisites required for this course. Make sure you completed the pre-requisites before enrolling.

CLEAR