WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

CS571: Building User Interfaces | Fall 2025

Section 001 | Tu, Th 9:30 am – 10:45 am | 1524 Morgridge Hall
Section 002 | Tu, Th 2:30 pm – 3:45 pm | 1524 Morgridge Hall

A three-credit undergraduate course in the Department of Computer Sciences at UW-Madison. Lectures are held in-person while recordings are posted online. In-class participation is encouraged.

CS400 is a prerequisite for this course. Students are expected to have a solid foundation in computer science and programming before taking this course.

Course Description

Introduces software development of user interfaces (UIs). Build competence in implementing UIs using state-of-the-art (1) UI paradigms, such as event-driven interfaces, direct-manipulation interfaces, and dialogue-based interaction; (2) methods for capturing, interpreting, and responding to different forms of user input and states, including pointing, text entry, speech, touch, gestures, user activity, context, and physiological states; and (3) platform-specific UI development APIs, frameworks, and toolkits for multiple platforms including web/mobile/desktop interfaces, natural user interfaces, and voice user interfaces. Learn about the fundamental concepts, technologies, algorithms, and methods in building user interfaces, implement UIs using of state-of-the-art UI development tools, and build a UI development portfolio.

Course Learning Outcomes (CLOs):

- Program front-end, user-facing software elements using the state-of-the-art programming languages, frameworks, and libraries.
- Engage in design thinking around user interface needs and problems, ideate and communicate conceptual design solutions
- Prototype and develop user interfaces for the web, mobile, and voice user interfaces (VUIs)
- Follow user-centered design principles, heuristics, and methods to iteratively build, assess, and refine design solutions
- Create visual designs, layouts, and navigation structures, and effectively use design languages, color palettes, and platform-specific design elements

How Credit Hours are Met by the Course

This class meets for two 75-minute class periods each week over the semester and carries the expectation that students will work on course learning activities for about 3 hours out of the classroom for every class period. The syllabus includes more information about meeting times and expectations for student work.

Course Schedule

The course schedule is made up of lectures on both implementation and fundamental design/HCI topics. Lectures on implementation typically happen on Tuesdays, and lectures on fundamental design/HCI topics typically happen on Thursdays. The course schedule below is tentative and subject to change.

Week	Topics	Homework	ICAs
#1	Intro to CS571	HW0 (1 pt)	No ICA
#2	Web Dev Basics 1 & Web Dev Basics 2	HW1	No ICA
#3	Web Dev Basics 3 & Design Thinking	HW2	ICA A
#4	Web Dev 1 & Visual Design	HW3	ICA B
#5	Web Dev 2 & Web Design	HW4	ICA C
#6	Web Dev 3 & Interaction Design	HW5	ICA D
#7	Web Dev 4 & Accessibility	HW6	ICA E
#8	Web Dev 5 & **Midterm Exam (Oct 23rd @ 5:45 PM)**	No HW	No ICA
#9	Mobile Dev 1 & Prototyping	HW7	ICA F
#10	Mobile Dev 2 & Mobile Design	HW8	No ICA
#11	Mobile Dev 3 & VUI Design	HW9	ICA H
#12	Voice Dev 1 & Designing Agents	HW10	ICA I
#13	Voice Dev 2 & Thanksgiving (No Class)	HW11	No ICA
#14	Full Stack Development & Expert Evaluation	No HW	ICA J
#15	User Evaluation & Security (Bonus Lecture)	Bonus Activity	ICA K
#16	**Final Exam (Dec 18th @ 5:05 PM)**	No HW	No ICA

Instructional Staff

Office hours may be held in-person or via Zoom. Please see Canvas for further office hours information.

Instructor	Cole Nelson	ctnelson2@wisc.edu
TA	Hongtao Hao	hongtaoh@cs.wisc.edu
TA	Zhaoyang Liu	zliu934@wisc.edu
TA	Amy Koike	ekoike@wisc.edu
TA	Kassie Povinelli	jpovinelli@wisc.edu
TA	Daniel Killough	dkillough@wisc.edu

Many Peer Mentors and Graders also make this course possible! Their information can be found on Canvas.

Grading

Students will be graded on a set of individual homeworks, in-class activities (ICAs), a web project, a midterm exam, a final exam, and attendance. The points are distributed as such...

Item	Points	Notes
Homework (HWs)	34	12 HWs, 3 pts each. HW0 worth 1 point. **AI use is restricted.**
In-Class Activities (ICAs)	10	11 ICAs, 1 pt each. Lowest dropped. **AI use is restricted.**
Web Project	10	Develop a web project of your choice. **AI use is encouraged.**
Midterm Exam	20	Students may use a *double-sided* notesheet.
Final Exam	25	Cumulative. Students may use a *double-sided* notesheet.
Attendance	1	Must attend 20 of 26 classes; each worth 0.05 pts, 1 pt max.

All regrade requests must be made *within 1 week* of the grade being returned. No later regrade requests will be accepted. Please email your instructor or TA if you believe a mistake was made in grading.

Students have a total of 10 late days that can be used on HWs throughout the semester. When all 10 late days are depleted, each day late will be 10% off that assignment's grade. No assignment can be turned in >7 days late. Days late are rounded up to the nearest whole number – e.g. an assignment turned in at 12:01 am will use a late day. **ICAs and Web Project deliverables may not be submitted late.**

Grades are assigned using the following scoring system…

Grade	A	AB	B	BC	C	D	F
Range	[100, 94]	(94, 88]	(88, 82]	(82, 76]	(76, 70]	(70, 60]	(60, 0]

There is no rounding of grades, though 1 bonus point will be offered through a bonus activity related to security in the last week of class.

Course Assignments

(HWs and ICAs) Typically, implementation lectures will have a corresponding homework (HW) while design lectures will have a corresponding in-class activity (ICA). These are used as learning activities, and as such *the use of generative AI is restricted*. These assignments are designed to help students engage with and better understand the material.

(Web Project) The web project is a new component of CS571! It is designed to put students' web design and development skills into practice, and thus *the use of generative AI is permitted and encouraged*. More details can be found at the bottom of this syllabus.

(Midterm and Final Exam) The midterm and final exam are designed to assess students' overall knowledge of course content. Students are allowed a *printed* double-sided notesheet for both; no other materials are allowed. The exams are held *in-person*. The final exam is *cumulative*.

(Attendance) Attendance is an important part of students' learning. Each class is worth 0.05 pts, for up to a maximum of 1 pt; this corresponds to 20 of the 26 classes. TopHat will be used to take attendance.

Course Materials

There is no required textbook for this course. The instructional content for the class is curated from different books, articles, and multimedia resources. All reading and multimedia material will be provided through Canvas directly to PDFs or links to archives or multimedia resources (e.g., YouTube). Similarly, information and tutorials on all necessary tools and software will be provided on Canvas.

In-Class Activities will be uploaded via *Canvas*. Assignments will be submitted via *GitHub Classroom*. Questions will be asked via *Piazza*. Personal matters will be handled via *email*.

Students are expected to download & install the following (free) software: TopHat, VSCode, NodeJS 22 with NPM 10, Postman, and Git. For mobile devices, Expo will be used.

Generative AI & Academic Integrity

The use of AI on HWs and ICAs is restricted. Assignment descriptions should be treated like proprietary company information – they should not be pasted into or used by any AI tool. Instead, AI should be used as a personal tutor – a good point of reference to understand a function or explore a concept deeper, but not to complete a student's HW or ICA for them.

These assignments are learning activities specifically designed to help students better understand the course material. *Please be sure to disable any AI assistants (such as Copilot or Claude) while working on HWs and ICAs that read, train, or use any of the original content of the HWs or ICAs.* All content submitted for the HWs and ICAs should be the student's original work; not produced from any other person, repository, or artificial agent.

For example, it is okay to ask an AI agent…

- How does a for loop work in JavaScript?
- Can you give me some examples of how to fetch data from an API?
- What are the primary aspects of visual design?

However, it is *not* okay to ask an AI agent…

- Complete this for me (pasting in the assignment README).
- Fix this for me (pasting in assignment function).
- Respond to these questions (pasting in the ICA prompt).

Finally, to reiterate, *please be sure to disable any AI assistants (such as Copilot or Claude) while working on HWs and ICAs that read, train, or use any of the original content of the HWs or ICAs.* These tools, while helpful in practice, stunt a student's learning of course material.

That being said, there is value to using these tools in practice. **The use of AI on the web project is permitted and encouraged.** As software engineers, students *will* be using these tools in practice. As such the web project is an open-ended project where students can use AI with no restrictions. While the university offers a Copilot license, any tool is welcomed and encouraged.

If you have questions about this policy, please reach out to your instructor or TA.

By enrolling in this course, each student assumes the responsibilities of an active participant in UW-Madison's community of scholars in which everyone's academic work and behavior are held to the highest academic integrity standards. Academic misconduct compromises the integrity of the university. Cheating, fabrication, plagiarism, unauthorized collaboration, and helping others commit these acts are examples of academic misconduct, which can result in disciplinary action. This includes but is not limited to failure on the assignment/course, disciplinary probation, or suspension. Substantial or repeated cases of misconduct will be forwarded to the Office of Student Conduct & Community Standards for additional review. For more information, refer to https://conduct.students.wisc.edu/academic-integrity/

Other Course Policies

Students with special needs should inform the instructor immediately via email so that accommodations can be made. Let the instructor know well in advance if an assignment, deadline, or major project milestone interferes with an important religious or cultural observance/event.

Rules, Rights, and Responsibilities

See http://guide.wisc.edu/undergraduate/#rulesrightsandresponsibilitiestext

Diversity & Inclusion

Diversity is a source of strength, creativity, and innovation for UW-Madison. We value the contributions of each person and respect the profound ways their identity, culture, background, experience, status, abilities, and opinion enrich the university community. We commit ourselves to the pursuit of excellence in teaching, research, outreach, and diversity as inextricably linked goals. The University fulfills its public mission by creating a welcoming and inclusive community for people from every background – people who as students, faculty, and staff serve Wisconsin and the world.

Accommodations for Students with Disabilities

McBurney Disability Resource Center syllabus statement: The University of Wisconsin-Madison supports the right of all enrolled students to a full and equal educational opportunity. The Americans with Disabilities Act (ADA), Wisconsin State Statute (36.12), and UW-Madison policy (Faculty Document 1071) require that students with disabilities be reasonably accommodated in instruction and campus life. Reasonable accommodations for students with disabilities is a shared faculty and student responsibility. Students are expected to inform the instructor of their need for instructional accommodations by the end of the third week of the semester, or as soon as possible after a disability has been incurred or recognized. The instructor will work either directly with the student or in coordination with the McBurney Center to identify and provide reasonable instructional accommodations. Disability information, including instructional accommodations as part of a student's educational record, is confidential and protected under FERPA. In addition to completing an electronic Faculty Notification Letter request through McBurney Connect, it is important for students to contact the course instructor directly by the end of the third week of the semester to set up a meeting to discuss implementation of any necessary accommodations. This early communication helps ensure that accommodations can be implemented in a timely manner. For example, if an alternative exam room is needed, arrangements must be made well in advance of an exam date to ensure room availability and to secure a room booking.

Additional Disability Statement

In addition to completing an electronic Faculty Notification Letter request through McBurney Connect, it is important for students to contact the course instructor directly by the end of the third week of the semester to set up a meeting to discuss implementation of any necessary accommodations. This early communication helps ensure that accommodations can be implemented in a timely manner. For example, if an alternative exam room is needed, arrangements must be made well in advance of an exam date to ensure room availability and to secure a room booking.

Course Evaluations

Students will be provided with an opportunity to evaluate their enrolled courses and their learning experience. Student participation is an integral component of course development, and confidential feedback is important to the institution. UW-Madison strongly encourages student participation.

Mental Health

UHS provides no-cost mental health services including individual, couple/partner, group counseling, outreach programming, and stress management. UHS also offers 24/7 crisis services. Psychiatry services are also available for medication management. If at any time you are feeling unsafe, dial 608-265-5600 option 9 or dial 988.

Laptop Lending Program

The CDIS IT Department now has Windows laptops available for lending to students who do not have an appropriate computer for a CS class. Loans can be for a short duration (while the student's laptop is being repaired, for example) or for the entire semester. Students who need to borrow a laptop should send an email to **it@cdis.wisc.edu** to initiate the loan.

Change Log

None yet.

Web Project Details

The web project is designed to allow students to design a web application to tackle a problem of their choosing. Additionally, this project may be completed in groups of *up to* 3 students, though this is *not* required – it is solely to your benefit. More details will be given at the end of the “Visual Design” lecture on Sep 25, 2025 to kick off the project. Briefly, the project will follow the schedule below…

Deliverable	**Points**	**Due**	**Expectation**
Proposal	1	Oct 15, 2025	Establish project domain and (optionally) form group.
Initial Publish	2	Nov 5, 2025	Begin work and publish the website to GitHub Pages.
Check-In	3	Nov 19, 2025	Show substantial progress since the last milestone and leave feedback on another person/group’s project.
Final Website & Presentation	4	Dec 10, 2025	Complete work on the website. The final website should be a polished product showcasing your knowledge of both design and implementation. Leave feedback on another person/group’s project.

Proposal

The proposal document will define your group members (if applicable) and project scope through a 200-300 word summary. The project scope is up to you! You could build…

- A promotional website that focuses on a specific event (e.g., a concert, charity run, or conference). This may feature information about the event, an event schedule, ticket purchase links, and a countdown timer!
- A club website that conveys information about an organization you are a part of. This may feature information about the club, an upcoming events list, set of beliefs and values, and membership inquiry form!
- A travel website that shows the experiences that you and your group members have embarked on. This may feature information about your favorite destinations, reviews, and a chance for interaction with visitors via the comment section.

Whatever your proposed project is, it is expected to have an *interactive* element. That is, beyond just displaying information to the user, the user is able to interact in some way (e.g. register for an event, sign up for membership, or bookmark favorite locations). Please be sure to think carefully about this requirement if you plan on creating a personal website.

Following your proposal, a member of course staff will reach out with either an “acceptance”, “acceptance with revisions”, or “needs revisions”. Furthermore, a GitHub repository will be created for you and your team members (if applicable). This repository will be used to both host and deploy your web application.

Initial Publish

The only deliverable for the initial publish is a website deployed to GitHub.io; these next steps will guide you through the process of creating and deploying your website.

You will build your website using React + Vite. Only one person from your group needs to initialize the repository with a React app as described below. **Additionally, we walk through this process of setting up your project together in a lecture video here (will be updated for Fall 2025).**

1. Create a React App

In a folder of your choosing, run the following commands from a terminal…

```
npm create vite@latest p0-project -- --template react
cd p0-project
npm install
```

You should replace `p0-project` *with the name of your group project; e.g.* `p1-project`*.*

If you are prompted, you should choose the “React” and “JavaScript” templates.

2. Remove StrictMode

`StrictMode` introduces behavior that we did not cover in class (such as double-fetching). While you are welcome to keep it, it is not included in any of our in-class exercises and HW assignments. You can “disable” StrictMode by removing the `StrictMode` tags from your `main.jsx` file. Your `main.jsx` file should simply insert the `App` component into the `root` of the DOM.

3. Install React-Bootstrap

To install React-Bootstrap, you’ll run…

```
npm install react-bootstrap bootstrap
```

You’ll also need to include the Bootstrap CSS. To do this, you will include…

a) A `link` to the CSS file within the `head` of your `index.html`.

```
<link
  rel="stylesheet"
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.5/dist/css/bootstrap.min.css"
  integrity="sha384-SgOJa3DmI69IUzQ2PVdRZhwQ+dy64/BUtbMJw1MZ8t5HZApcHrRKUc4W0kG879m7"
  crossorigin="anonymous"
/>
```

b) An `import` at the top of your `main.jsx`.

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

4. Install React Router

To install React Router, you’ll simply run…

```
npm install react-router
```

Be sure to use `HashRouter`; not `BrowserRouter`! We will use the Declarative mode.

4. Configure Vite Build

To deploy to GitHub pages, you'll need to replace your `vite.config.js` with the following…

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

export default defineConfig({
  plugins: [react()],
  base: '/p0/',
  build: {
    outDir: 'docs'
  }
})
```

Be sure to replace p0 with your project group number.

Then, to build your production bundle, run…

```
npm run build
```

5. Configure GitHub Pages

To allow deployment to GitHub Pages, visit your group's GitHub repository, go to "Settings" > "Pages" and select "Deploy from Branch" as the source, "main" as the branch, and "docs" as the folder. You are welcome to configure these; however, this guide is designed for these settings.

6. Push to GitHub

You'll need to add all of your work to your group's GitHub repository. To do this, you will need to run the following commands in the same directory as your web project…

```
npm run build
git init
git remote add origin https://github.com/CS571-F25/p0.git
git pull origin main
git add -A
git commit -m "initial commit"
git branch -M main
git push -u origin main
```

Be sure to replace `p0` *with your project number.*

7. Deployed!

Your website should be accessible via github.io, e.g. http://cs571-f25.github.io/p0.

After completing this initial setup, be sure to run `npm run build` before every push to the `main` branch in order for your changes to appear on github.io. If you are working with others, it may be beneficial to develop a Git branching strategy to avoid conflicts. However, how you choose to complete your work is up to you!

Check-In

At the check-in, you should have shown substantial progress since the last milestone. This means, at a minimum, your website should…

- Be committed and pushed to GitHub.
- Be live and functional on GitHub.io.
- Have *some* use of React Bootstrap or some other design library.
- Have a primary navigation bar present and functional.
- Have at least 3 pages at least *partially* developed using React Router or some other routing library.
- Have at least 5 components defined and *meaningfully* used.

Following your check-in, you will be invited to peer review another group's progress. In your peer review, in 50-150 words, please give at least one piece of feedback. Your peer review will be anonymous to the receiver.

Final Website & Recorded Presentation

The final deliverable is a finished website and recorded presentation.

The final website should…

- Be committed and pushed to GitHub.
- Be live and functional on GitHub.io.
- Have *consistent* use of React Bootstrap or some other design library.
- Have a primary navigation bar present and functional.
- Have at least 3 pages fully developed using React Router or some other routing library.
- Have at least 12 components defined and *meaningfully* used.
- Include a meaningfully interactable element (see "Proposal" for details).
- Include the thoughtful use of design principles.
- Be accessible, including…
  - No skipped heading levels.
  - Having appropriate alt text on all images.
  - Having sufficient color contrast meeting WCAG AA standards.
  - Having all inputs appropriately labeled.
  - Having all forms completable via keyboard.

The final presentation should…

- Be between 4 - 7 minutes of length.
- Showcase the main features and functionalities of your website.
- Describe the design decisions made.

A template for the presentation will be provided in the Canvas portal. How you record your presentation is up to you, but any easy solution is to use Zoom local recordings.

Following your final website and presentation submission, you will be invited to peer review another group's website and presentation. In your peer review, in 50-150 words, please identify at least one positive aspect of their presentation or website, and one piece of constructive criticism. Your peer review will be anonymous to the receiver.

Other Details

1. Group Work

Working in a group setting can be challenging but rewarding! As software engineers, you *will* be working in a group setting, so I'd encourage (but not require) you to get some practice in this assignment. If you choose to work in a group, make expectations clear and make communication easy. If issues arise, reach out to course staff earlier rather than later so we can help address these issues.

2. AI Usage

You should have developed foundational implementation and design skills through the HWs and ICAs. As such, there is value to using AI tools in practice. **The use of AI on the Web Project is permitted and encouraged.** As software engineers, you *will* be using these tools in practice; the Web Project is an open-ended project where you can use AI with no restrictions. While the university offers a Copilot license, any tool is welcomed and encouraged.

3. Expectations

This project is open-ended and may leave questions about what is “acceptable” and “unacceptable” work. The requirements listed above do their best to define the expectations of this project, but a good rule of thumb is “Would I be proud enough of this work to use it in my personal portfolio?” The final product should be a well-executed, professional website that demonstrates your ability to build user interfaces. If you have questions specific to your project, please reach out to course staff.