Welcome to Class!

CS 390 – Web Application Development

J. Setpal

August 21, 2023

About Us

Jinen Setpal
Major: Data Science
Interests: Racing, Chess
`jsetpal@purdue.edu`

Mikail Khan
Major: Computer Science
Interest: Programming Languages
`mikail@purdue.edu`

About Us

Jinen Setpal
Major: Data Science
Interests: Racing, Chess
`jsetpal@purdue.edu`

Mikail Khan
Major: Computer Science
Interest: Programming Languages
`mikail@purdue.edu`

Outline

1. Why it's Worth Your Time
2. Logistics
3. Wat
4. ETC

Outline

1. Why it's Worth Your Time
2. Logistics
3. Wat
4. ETC

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development.

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development. Through this course, we aim to cover:

- **HTML: Structure your webpage.**
- **CSS: Style your webpage.**
- JavaScript: Script functionality into your webpage.
- Web Assembly: Build binaries that run in the browser sandbox.
- Node: Run JavaScript server-side.
- Express: Middleware and Routing for server-side applications.
- React: Design *react*-ive front-end applications.
- Databases: Full-stack data management.
- Web Hosting: Putting your website on the internet.
- Web Security: Breaking and hardening everything we use.
- PSOs: Git / Browser DevTools

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development. Through this course, we aim to cover:

- HTML: Structure your webpage.
- CSS: Style your webpage.
- **JavaScript: Script functionality into your webpage.**
- **Web Assembly: Build binaries that run in the browser sandbox.**
- Node: Run JavaScript server-side.
- Express: Middleware and Routing for server-side applications.
- React: Design *react*-ive front-end applications.
- Databases: Full-stack data management.
- Web Hosting: Putting your website on the internet.
- Web Security: Breaking and hardening everything we use.
- PSOs: Git / Browser DevTools

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development. Through this course, we aim to cover:

- HTML: Structure your webpage.
- CSS: Style your webpage.
- JavaScript: Script functionality into your webpage.
- Web Assembly: Build binaries that run in the browser sandbox.
- **Node: Run JavaScript server-side.**
- **Express: Middleware and Routing for server-side applications.**
- React: Design *react*-ive front-end applications.
- Databases: Full-stack data management.
- Web Hosting: Putting your website on the internet.
- Web Security: Breaking and hardening everything we use.
- PSOs: Git / Browser DevTools

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development. Through this course, we aim to cover:

- HTML: Structure your webpage.
- CSS: Style your webpage.
- JavaScript: Script functionality into your webpage.
- Web Assembly: Build binaries that run in the browser sandbox.
- Node: Run JavaScript server-side.
- Express: Middleware and Routing for server-side applications.
- **React: Design react-ive front-end applications.**
- Databases: Full-stack data management.
- Web Hosting: Putting your website on the internet.
- Web Security: Breaking and hardening everything we use.
- PSOs: Git / Browser DevTools

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development. Through this course, we aim to cover:

- HTML: Structure your webpage.
- CSS: Style your webpage.
- JavaScript: Script functionality into your webpage.
- Web Assembly: Build binaries that run in the browser sandbox.
- Node: Run JavaScript server-side.
- Express: Middleware and Routing for server-side applications.
- React: Design *react*-ive front-end applications.
- **Databases: Full-stack data management.**
- Web Hosting: Putting your website on the internet.
- Web Security: Breaking and hardening everything we use.
- PSOs: Git / Browser DevTools

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development. Through this course, we aim to cover:

- HTML: Structure your webpage.
- CSS: Style your webpage.
- JavaScript: Script functionality into your webpage.
- Web Assembly: Build binaries that run in the browser sandbox.
- Node: Run JavaScript server-side.
- Express: Middleware and Routing for server-side applications.
- React: Design *react*-ive front-end applications.
- Databases: Full-stack data management.
- **Web Hosting: Putting your website on the internet.**
- Web Security: Breaking and hardening everything we use.
- PSOs: Git / Browser DevTools

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development. Through this course, we aim to cover:

- HTML: Structure your webpage.
- CSS: Style your webpage.
- JavaScript: Script functionality into your webpage.
- Web Assembly: Build binaries that run in the browser sandbox.
- Node: Run JavaScript server-side.
- Express: Middleware and Routing for server-side applications.
- React: Design *react*-ive front-end applications.
- Databases: Full-stack data management.
- Web Hosting: Putting your website on the internet.
- **Web Security: Breaking and hardening everything we use.**
- PSOs: Git / Browser DevTools

WIWYT – Web Application Programming

This course attempts to be a practical primer towards web development. Through this course, we aim to cover:

- HTML: Structure your webpage.
- CSS: Style your webpage.
- JavaScript: Script functionality into your webpage.
- Web Assembly: Build binaries that run in the browser sandbox.
- Node: Run JavaScript server-side.
- Express: Middleware and Routing for server-side applications.
- React: Design *react*-ive front-end applications.
- Databases: Full-stack data management.
- Web Hosting: Putting your website on the internet.
- Web Security: Breaking and hardening everything we use.
- **PSOs: Git / Browser DevTools**

Outline

1. Why it's Worth Your Time
2. Logistics
3. Wat
4. ETC

Grading

The points distribution is as follows:

Category	**Points**
Homework	70%
Project	30%
Attendance	10%
Total	**110%**

Grading

The points distribution is as follows:

Category	**Points**
Homework	70%
Project	30%
Attendance	10%
Total	**110%**

Tentative thresholds are as follows:

Grade	**Threshold**
A+	$> 95\%$
A	$94\% - 90\%$
A-	$89\% - 85\%$
B+	$84\% - 80\%$
B	$79\% - 75\%$
B-	$74\% - 70\%$
C+	$69\% - 65\%$
C	$64\% - 60\%$
C-	$59\% - 55\%$
F	$< 55\%$

Homework

Assignments will be released after each module upto React.

Homework

Assignments will be released after each module upto React. Both Node / Express and React will have two homeworks each, bringing the total to 6.

Homework

Assignments will be released after each module upto React. Both Node / Express and React will have two homeworks each, bringing the total to 6.

In addition, we'll have one optional assignment for extra credit.

Homework

Assignments will be released after each module upto React. Both Node / Express and React will have two homeworks each, bringing the total to 6.

In addition, we'll have one optional assignment for extra credit.

The homeworks (for the most part) are structured such that one assignment follows the other in building a resume website.

Homework

Assignments will be released after each module upto React. Both Node / Express and React will have two homeworks each, bringing the total to 6.

In addition, we'll have one optional assignment for extra credit.

The homeworks (for the most part) are structured such that one assignment follows the other in building a resume website.

Note that choosing to skip one of the non-tangential assignments for the extra credit one will likely affect your assignment after that!

Homework

Assignments will be released after each module upto React. Both Node / Express and React will have two homeworks each, bringing the total to 6.

In addition, we'll have one optional assignment for extra credit.

The homeworks (for the most part) are structured such that one assignment follows the other in building a resume website.

Note that choosing to skip one of the non-tangential assignments for the extra credit one will likely affect your assignment after that!

Late Policy: Everyone gets 5 total late days for any assignment, no questions asked.

Homework

Assignments will be released after each module upto React. Both Node / Express and React will have two homeworks each, bringing the total to 6.

In addition, we'll have one optional assignment for extra credit.

The homeworks (for the most part) are structured such that one assignment follows the other in building a resume website.

Note that choosing to skip one of the non-tangential assignments for the extra credit one will likely affect your assignment after that!

Late Policy: Everyone gets 5 total late days for any assignment, no questions asked. Beyond this, we will accept assignments upto a week late for half credit.

Homework

Assignments will be released after each module upto React. Both Node / Express and React will have two homeworks each, bringing the total to 6.

In addition, we'll have one optional assignment for extra credit.

The homeworks (for the most part) are structured such that one assignment follows the other in building a resume website.

Note that choosing to skip one of the non-tangential assignments for the extra credit one will likely affect your assignment after that!

Late Policy: Everyone gets 5 total late days for any assignment, no questions asked. Beyond this, we will accept assignments upto a week late for half credit. If more late days are required with legitimate cause, please reach out to ODOS.

Project

The final project for this class is a group project to build a **full-stack web application**, using the tools and frameworks covered through the semester.

Project

The final project for this class is a group project to build a **full-stack web application**, using the tools and frameworks covered through the semester.

The applications will then be hosted symposium style on a common domain.

Project

The final project for this class is a group project to build a **full-stack web application**, using the tools and frameworks covered through the semester.

The applications will then be hosted symposium style on a common domain. The last few classes are allotted for project presentations.

Project

The final project for this class is a group project to build a **full-stack web application**, using the tools and frameworks covered through the semester.

The applications will then be hosted symposium style on a common domain. The last few classes are allotted for project presentations.

Each project (one submission per team) must be proposed by **September 30, 2023**. We will provide general feedback, and evaluate it for complexity (generally: members $\propto$ complexity).

Project

The final project for this class is a group project to build a **full-stack web application**, using the tools and frameworks covered through the semester.

The applications will then be hosted symposium style on a common domain. The last few classes are allotted for project presentations.

Each project (one submission per team) must be proposed by **September 30, 2023**. We will provide general feedback, and evaluate it for complexity (generally: members $\propto$ complexity).

Details will be added to the Project Section under Brightspace.

Tools We'll Use

- **For Assignment / Project Submission:** GitHub Classroom

Tools We'll Use

- **For Assignment / Project Submission:** GitHub Classroom
- **For Development:** Text Editor (e.g. vim, emacs, VSC, Sublime)

Tools We'll Use

- **For Assignment / Project Submission:** GitHub Classroom
- **For Development:** Text Editor (e.g. vim, emacs, VSC, Sublime), Node.js, NPM, React.js, Express.js

Tools We'll Use

- **For Assignment / Project Submission:** GitHub Classroom
- **For Development:** Text Editor (e.g. vim, emacs, VSC, Sublime), Node.js, NPM, React.js, Express.js
- **For Communication:** Brightspace + TBA (from today's poll)

We don't have any required reading or materials for this course.

Academic Integrity

- Don't copy any code **without attribution**. It's okay to use resources like Stack OverFlow / MDN[1], provided you cite your sources.

[1] We will share a complete list of approved resources on Brightspace.

Academic Integrity

- Don't copy any code **without attribution**. It's okay to use resources like Stack OverFlow / MDN[1], provided you cite your sources.
- **Don't copy each others code** under any circumstance.

[1] We will share a complete list of approved resources on Brightspace.

Academic Integrity

- Don't copy any code **without attribution**. It's okay to use resources like Stack OverFlow / MDN[1], provided you cite your sources.
- **Don't copy each others code** under any circumstance.
- Use resources responsibly; to debug, not solve the challenge entirely.

[1] We will share a complete list of approved resources on Brightspace.

Academic Integrity

- Don't copy any code **without attribution**. It's okay to use resources like Stack OverFlow / MDN[1], provided you cite your sources.
- **Don't copy each others code** under any circumstance.
- Use resources responsibly; to debug, not solve the challenge entirely.
- If you're unsure about if an action is acceptable, ask first!

[1] We will share a complete list of approved resources on Brightspace.

Academic Integrity

- Don't copy any code **without attribution**. It's okay to use resources like Stack OverFlow / MDN[1], provided you cite your sources.
- **Don't copy each others code** under any circumstance.
- Use resources responsibly; to debug, not solve the challenge entirely.
- If you're unsure about if an action is acceptable, ask first!

Violations of the above policy necessitate us to report it to ODOS.

[1] We will share a complete list of approved resources on Brightspace.

Accommodations

To make classes as student-friendly as possible:

- All lectures are recorded and uploaded to Brightspace.

Accommodations

To make classes as student-friendly as possible:

- All lectures are recorded and uploaded to Brightspace.
- You can take this class as distance learning, with express approval from your advisor.

Accommodations

To make classes as student-friendly as possible:

- All lectures are recorded and uploaded to Brightspace.
- You can take this class as distance learning, with express approval from your advisor.
- There are no exams or timed assignments.

Accommodations

To make classes as student-friendly as possible:

- All lectures are recorded and uploaded to Brightspace.
- You can take this class as distance learning, with express approval from your advisor.
- There are no exams or timed assignments.
- No restrictions will be placed on seating, note-taking, etc.

Accommodations

To make classes as student-friendly as possible:

- All lectures are recorded and uploaded to Brightspace.
- You can take this class as distance learning, with express approval from your advisor.
- There are no exams or timed assignments.
- No restrictions will be placed on seating, note-taking, etc.

If you are a student requiring accommodations like the change of grade schema, distance learning, preferred seating, or anything we do not have by default, please notify us and we'll work together to resolve it.

Outline

1. Why it's Worth Your Time

2. Logistics

3. Wat

4. ETC

Some Useful Background

JavaScript boasts two interesting behaviors: **type coercion**

Some Useful Background

JavaScript boasts two interesting behaviors: **type coercion** and **ASI**.

Some Useful Background

JavaScript boasts two interesting behaviors: **type coercion** and **ASI**.

Today, we'll look at the first nuance.

Some Useful Background

JavaScript boasts two interesting behaviors: **type coercion** and **ASI**.

Today, we'll look at the first nuance. Being weakly typed results a lot of unintuitive behaviors; let's uncover some of them!

This awesome excerpt is taken from a lightning talk by Gary Bernhardt from CodeMash 2012: `https://www.destroyallsoftware.com/talks/wat`

JavaScript's Wats

If you can view this screen, I am making a mistake.

Outline

1. Why it's Worth Your Time

2. Logistics

3. Wat

4. ETC

An Incomplete List of Great Resources

- The ODIN Project: `https://theodinproject.com/`
- MDN Docs: `https://developer.mozilla.org/en-US/docs/Web`
- Learn X in Y Minutes: `https://learnxinyminutes.com/`
- React Documentation: `https://react.dev/learn`
- freeCodeCamp: `https://www.freecodecamp.org/learn/`
- edX: `https://www.edx.org/`
- The Modern JavaScript Tutorial: `https://javascript.info/`

Thank you!

Have an awesome rest of your day!

Slides: `https://cs.purdue.edu/homes/jsetpal/slides/intro.pdf`

If anything's incorrect or unclear, please ping `jsetpal@purdue.edu`
I'll patch it ASAP.