

# CpS 404: Internet Application Development

*Spring Semester 2024-2025*

Instructor: Stephen Schaub                      Email: sschaub@bju.edu  
Office: SermonAudio Suite (MB 2<sup>nd</sup> floor)  
Office Hours: By Appointment

## Course Information

A survey of current Internet application development technologies including server database technologies, language choices for both Windows and Linux environments, and security considerations. Each student will build an Internet-based application using the technologies presented. *Prerequisite: CpS 301.*

**Overview:** This course tackles the technologies used and issues involved in the development of distributed applications that run on the Internet. The course covers technologies, tools, and techniques used to build client/server web applications that are efficient, scalable, secure, and maintainable, using the Node.js platform. The instructor assumes that students have a basic knowledge of SQL and HTML and are proficient programmers.

This is an advanced class in our computer science program. Expect to spend several hours a week writing code. The instructor will present key concepts and cover some coding techniques in class, but students are expected to "fill in the gaps" by doing their own research and reading in order to successfully complete assignments.

## Course Resources

**Textbook:** There is no textbook for this class. A variety of online readings will be assigned. See the course schedule.

**Announcements:** Students are expected to use the Microsoft Teams CpS 404 team to receive course announcements and to engage in the online experiences of this course

**Website:** The course website contains links to assignments and required readings: <https://cs.bju.edu/cps404>

## Grading

Grading:			
Qty	Item	Points	Total
3	Labs	5-10	25
4	Exercises	50-75	225
3	Projects	100-125	350
2	Written Tests	150	300
1	Final Exam	100	100
<b>Total Points:</b>		1000	

Scale:	
A	90-100%
B	80-90%
C	70-80%
D	60-70%
F	<60%

**Programming Assignments:** There are three types of assignments in the class: labs, exercises and projects. Labs are warmups that should take an hour or so. Exercises are small-scale assignments that typically take a few hours to complete. Projects are larger-scale assignments that will likely take many hours (10-20). Projects are graded as follows:

- **60% Correctness:** Program produces correct results; runs according to specification. Attention to the program specification is very important here.

- **20% Style:** Code is written according to style guidelines and instructor's requirements. Consistency and attention to detail are important. The goal is clear, easily understandable code, thoughtfully commented. **Programs which do not receive at least 20 of the 60 correctness points will receive no style credit.**
- **15% Reports:** Each program and project submission is accompanied with a written report.
- **5% Presentation:** Program assignment submitted according to instructions.

## Course Policies

---

**Assignments can receive full credit only if submitted by 11:59 p.m. on the day due.** A 25% penalty will be applied if the assignment is not turned in on time. No credit is possible after one week. I waive one late penalty for students who are punctual in their class attendance. If you anticipate trouble on an assignment, see me as soon as possible for assistance.

Students are required to submit a reasonable attempt for all programming assignments, even if the attempt is too late to receive credit. Failure to submit a reasonable attempt for one or more assignments may result in a penalty of up to one letter grade on the final course grade.

**Programs may be submitted late only by approval of the instructor.** I will allow this only for students who formally request permission to submit the program late. The request must be made by email, and should report the number of hours invested and include a description of problems encountered. Your current program effort should be uploaded to the submission system. The request must be submitted by the original assignment deadline. Requests indicating little effort invested will be denied.

**Electronic devices may not be used during class** except by special arrangement with the instructor.

**Gum chewing** in professional settings is inappropriate and therefore not permitted in class. Compliance with student handbook policies is expected during class.

## University Policies

---

### Attendance Policy

You are expected to attend class and be on time per the standard University attendance policy: <https://home.bju.edu/bju-policies/>. If you come late or leave early, I will record a partial attendance mark if you missed at most 15 minutes of class. If you miss more than 15 minutes of class, you will be marked absent. If you exceed the 3 allowed absences for this class, you may be withdrawn from class.

For planned absences, notify me a week ahead of time by e-mail. Written assignments and scheduled tests should be completed before your planned absence; please contact me to make arrangements for doing so. It is your responsibility to check in advance of a planned absence to verify what is due.

For absences due to illness or emergency, contact me by the end of the day of your absence to indicate the reason for your absence and to arrange for making up any graded work without penalty. In these situations, you will be able to make arrangements for making up tests without penalty for the first occurrence. Each subsequent time a test is missed because of incapacitating illness or emergency, an additional 10 percent grade penalty for that test will be incurred.

### Accommodations for Students with Disabilities

If you have a documented learning disability or if you are impaired in some way (auditory, visual, cognitive, neurological, or physical), please let the instructor know this within the first week of the course so that any necessary adjustments can be made before you get behind.

### Academic Honesty and Integrity Policy

See the Computer Science Department's Academic Integrity Policy:

<https://cs.bju.edu/academics/policies/academic-integrity-policy/>

*Copyright 2025 Stephen Schaub. Students are prohibited from selling (or being paid for taking) notes during the course to, or by any person, or commercial firm without the express written permission of the professor teaching the course. Students are prohibited from distributing course materials. This course uses copyrighted material from other sources. This material is being used by permission.*

<b>Schedule</b>			
Date	Day	Class	Assignment
Jan. 16	Th	HTML and HTTP	
Jan. 21	T	JavaScript and Node.js intro	
Jan. 23	Th	More JavaScript and Node.js	Exercise 1
Jan. 28	T	Node.js execution model	
Jan. 30	Th	Consuming web services	
Feb. 4	T	Node.js web services	
Feb. 6	Th	Node.js database apps	
Feb. 11	T	Unicode	
Feb. 13	Th	Intro to Docker	Project 1
Feb. 18	T	Test 1	
Feb. 20	T-F	<b>Bible Conference</b>	
Feb. 25	T	Concurrency; Express web apps	
Feb. 27	Th	Express Form Processing	Exercise 2
Mar. 4	T	State Management	
Mar. 6	Th	Multiuser Considerations	
Mar. 11	T	Mobile UI	
Mar. 13	Th	Promises and Async	
Mar. 18	T	Review	Project 2
Mar. 20	Th	Test 2	
Mar. 25-28	M-F	<b>Spring Break</b>	
Apr. 1	T	Client JavaScript	
Apr. 3	Th	Intro to Vue	
Apr. 8	T	Client State Management	
Apr. 10	Th	Cookie Security	Exercise 3
Apr. 15	T	Web App Security	
Apr. 17	Th	Password Management	
Apr. 22	T	Code Security Reviews	
Apr. 24	Th	Peer Code Reviews	Project 3
Apr. 29	T	Public Key	
May 1	Th	Final Review	
May 8	Th	<b>Final Exam</b>	

## Curriculum Information

---

### Context

This course supports the following objectives of the Computer Science and Information Technologies programs:

CS 1. Design and implement solutions to practical problems

CS 2. Use appropriate technology as a tool to solve problems in various domains

CS 6. Demonstrate an ability to acquire new knowledge in the computing discipline

Objective	Content	Assessment
Construct small to medium-sized web applications	All Lectures	Programs 1-3
Discuss techniques to maintain state using the stateless HTTP protocol	Java State Mgmt Lectures	Test 1
Design layered web applications that maintain appropriate separation between user interface and processing	Discussion around Programs 1-3	Programs 1-3
Discuss strategies to handle concurrency and multiple users in web applications	Lectures on Multiuser Issues and Database Concurrency	Test 2
Discuss common vulnerabilities in web applications, common attacks malicious users launch against web applications, and means of defense	Security lectures	Test 3