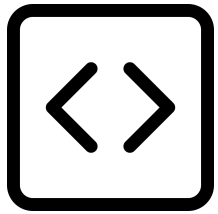
Code Critiquer System for the C Language and Embedded C



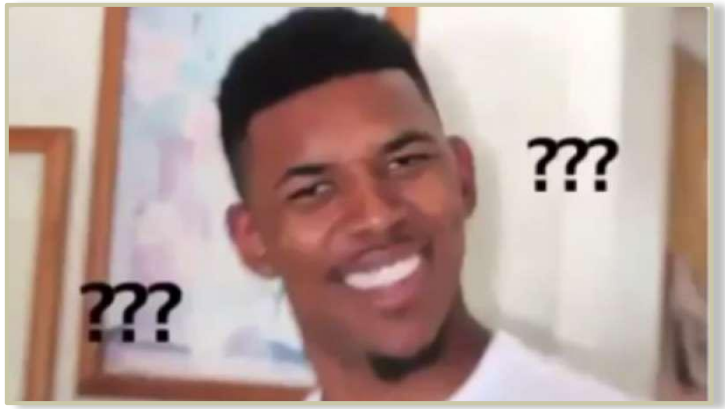
sdmay25-23:

sdmay25-23.sd.ece.iastate.edu/

James Joseph
Samuel Lickteig
Owen Sauser
Andrew Sand
Alix Noble

Client and Advisor: Dr. Diane Rover

```
0.705 /usr/bin/ld: /app/build/objs/static_example.o: in function `main':
0.705 static_example.c:(.text+0x0): multiple definition of `main';
/app/build/objs/blah.o:blah.c:(.text+0x8c): first defined here
0.705 /usr/bin/ld: /app/build/objs/static_example.o: in function `max':
0.705 static_example.c:(.text+0x87): multiple definition of `max';
/app/build/objs/example.o:example.c:(.text+0x0): first defined here
0.736 collect2: error: ld returned 1 exit status
0.737 make: *** [Makefile:60: /app/build/blah.out] Error 1
0.737 make: Target 'test' not remade because of errors.
```

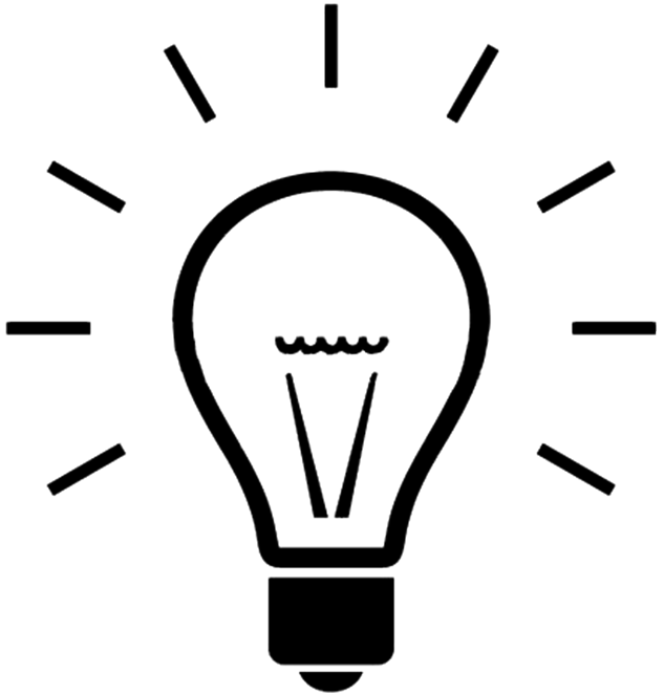


/app/build/objs/static_example.o: in function 'main':
static_example.c: multiple definition of 'main'

Feedback: In static_example.c, press CTRL+f. Type in "main" without the quotes. Look at both sections of the code, find the one you do not want to run, and delete everything between and including: main(...){...}

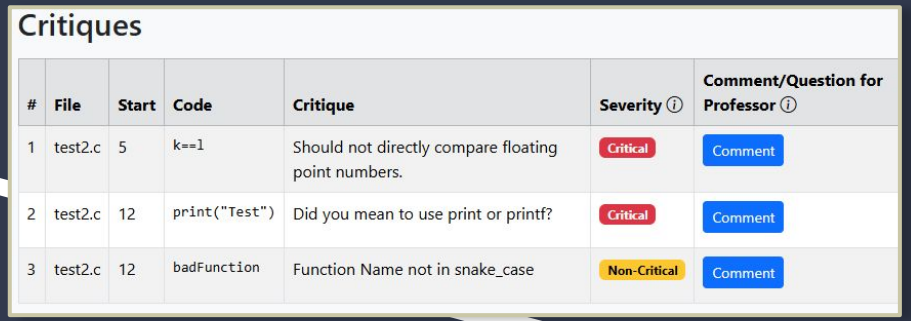
/app/build/objs/static_example.o: in function 'max':
static_example.c: multiple definition of 'max'

Feedback: In static_example.c, press CTRL+f. Type in "max" without the quotes. Look at both sections of the code, find the one you do not want to run, and delete everything between and including: max(...){...}



Problem Statement

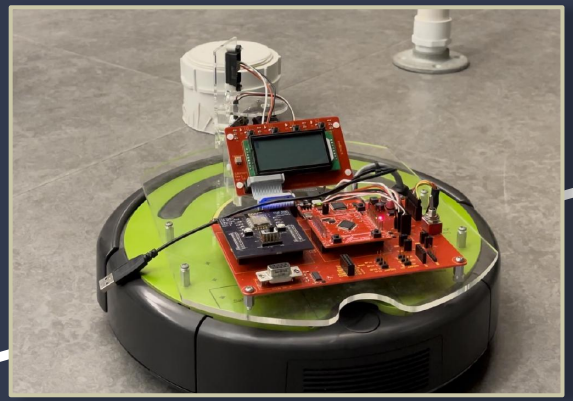
- CPR E 2880 is many students' first experience with C at Iowa State
- Errors can be confusing and overwhelming
- TAs are not always available to help



Critiques

#	File	Start	Code	Critique	Severity ⓘ	Comment/Question for Professor ⓘ
1	test2.c	5	`k==1`	Should not directly compare floating point numbers.	Critical	Comment
2	test2.c	12	`print("Test")`	Did you mean to use print or printf?	Critical	Comment
3	test2.c	12	`badFunction`	Function Name not in snake_case	Non-Critical	Comment

Code Critiquer System for the C Language and Embedded C



A CPR E 2880 CyBot

Project Overview

- Project is a web-based critiquer tool
  - Continuation of sdmay24-34
  - Students upload C files to tool
  - Files are statically analyzed to search for antipatterns
  - Tool generates student feedback
  - Instructors view analytics
- Tailored for CPR E 2880
- Targeting a Fall 2025 Prototype

Project Requirements

User Requirements (For Students)

- Easy to use
- Clear, concise, and aesthetically pleasing UI
- Compiler output must be parsed and explained
- Feedback is constructive

User Requirements (For Instructors and TAs)

- Access to statistics/analytics
  - Ability to sort student tests
  - Visual chart interpretations of data
  - Customizable
  - Ability to download information
- Easy to maintain

Functional Requirements

- C files upload successfully
- The program compiles provided code
- Static Analysis
  - Regular Expressions (Antipatterns)
  - Compiler Output
- Interact using GUI

Nonfunctional Requirements

UI

- GUI is simplistic and intuitive
- Quick to navigate
- All feedback is beginner-friendly

Performance

- Feedback provided in < 10 seconds
- File uploads should take no more than a few seconds

Nonfunctional Requirements (cont.)

Maintainability

- Adaptable for:
  - Variety of programming purposes
  - Instructor needs and preferences
- Well-documented
- Modular
- Easy to add antipatterns

Bug of the Week

GPIO Port E	PUSH BUTTONS
Pin 5:	AIN8
Pin 4:	AIN9
Pin 3:	PB_SW4
Pin 2:	PB_SW3
Pin 1:	PB_SW2
Pin 0:	PB_SW1

- Bug of the week is a concept where each week a new potential bug is highlighted
  - Selected on what the students may encounter that week
- We have made antipatterns for each bug of the week given to us by Dr. Rover

MTU Projects (Dr. Ureel)

Michigan Technological University

Pros:

- Multiple critiquers for different languages
- Education-focused
- Canvas integration

Cons:

- Products are just prototypes and not widely available
- Only uses static analysis

Market Gap

- Other market solutions are lacking key features desired by client
- Tailored to CPR E 2880/embedded C
- Reliability and Confidence
- Beginner-friendly explanations
- Customizable antipatterns
- Student analytics and feedback

Key Risks

- Possibly used for cheating
- Uploading malicious code
- Should help, not solve
- Should not mislead students
- Need to be aware of “false positives”

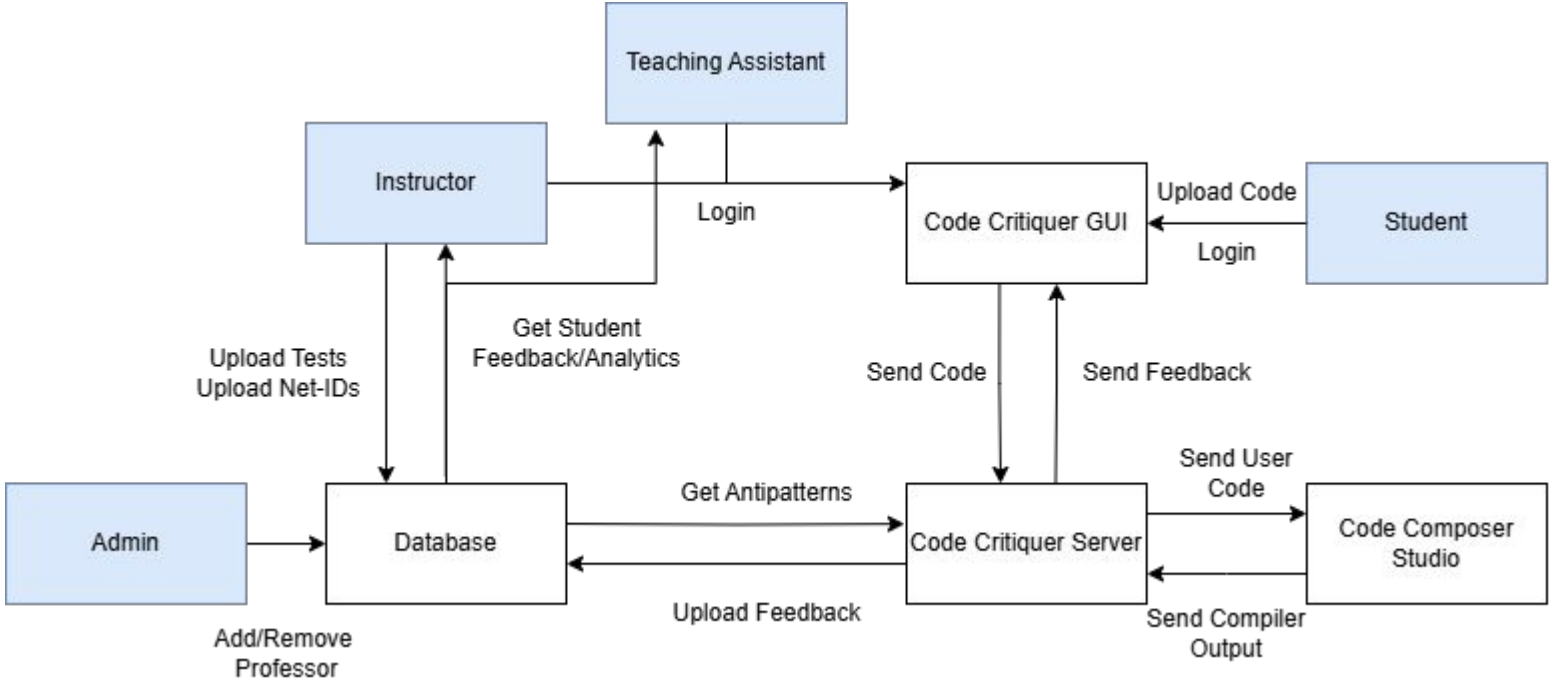
Risk Mitigation Strategies

- Secure Accounts (Hashing and Least Privilege)
- Feedback given is static
- Containerization
- Segmentation
- Contained in Iowa State network

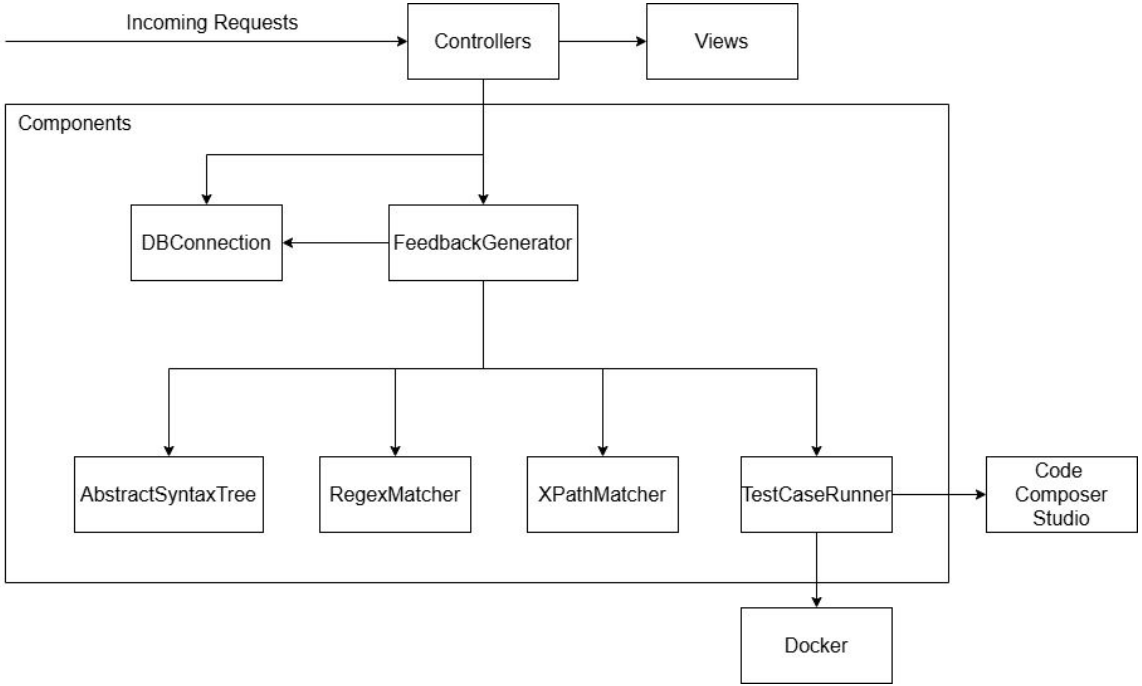
Resource/Cost Estimate

- Technical cost:
  - Uses available university resources
  - Minimal energy costs
  - Open-source software
- Human cost:
  - Maintenance once project is completed
  - Training to maintain system

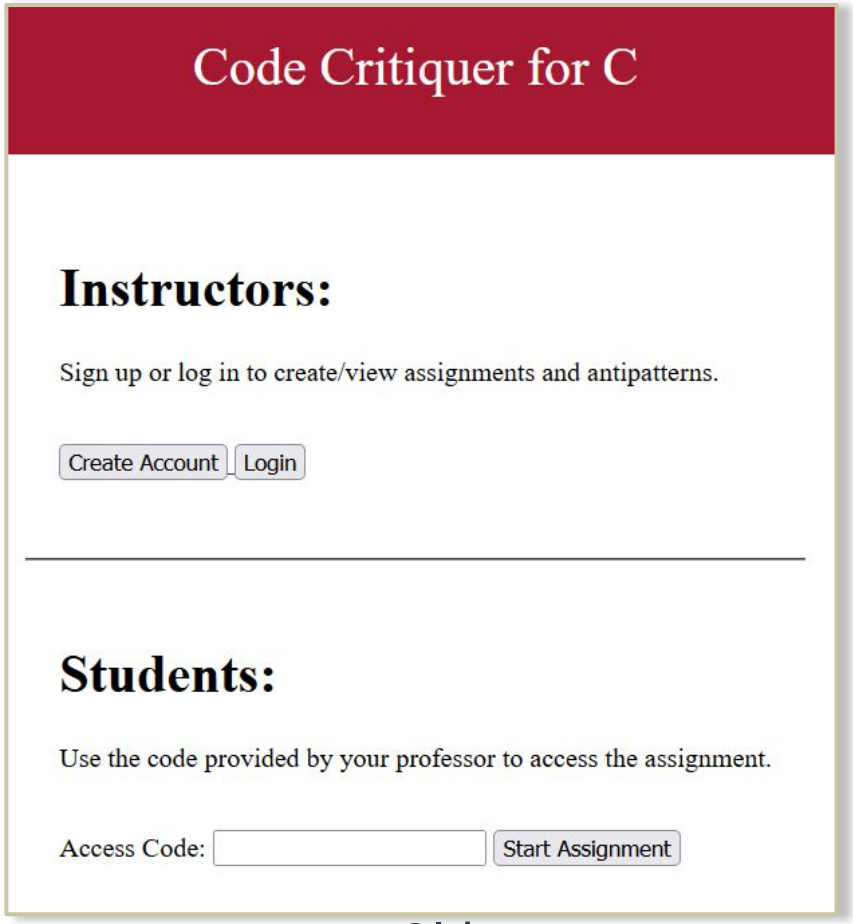
System Sketch

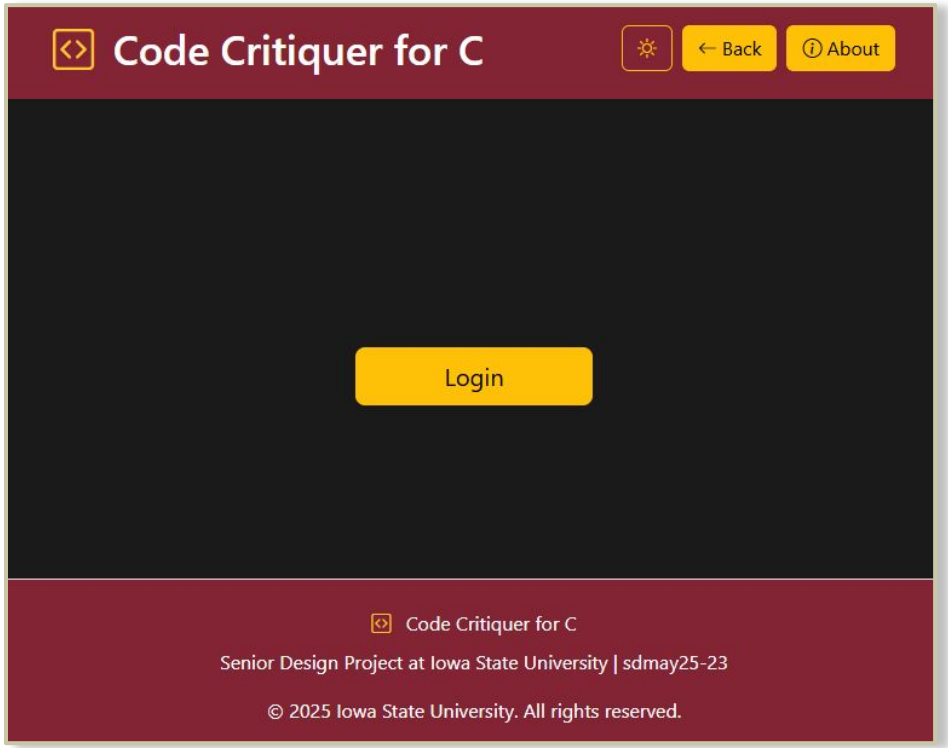


System Block Diagram

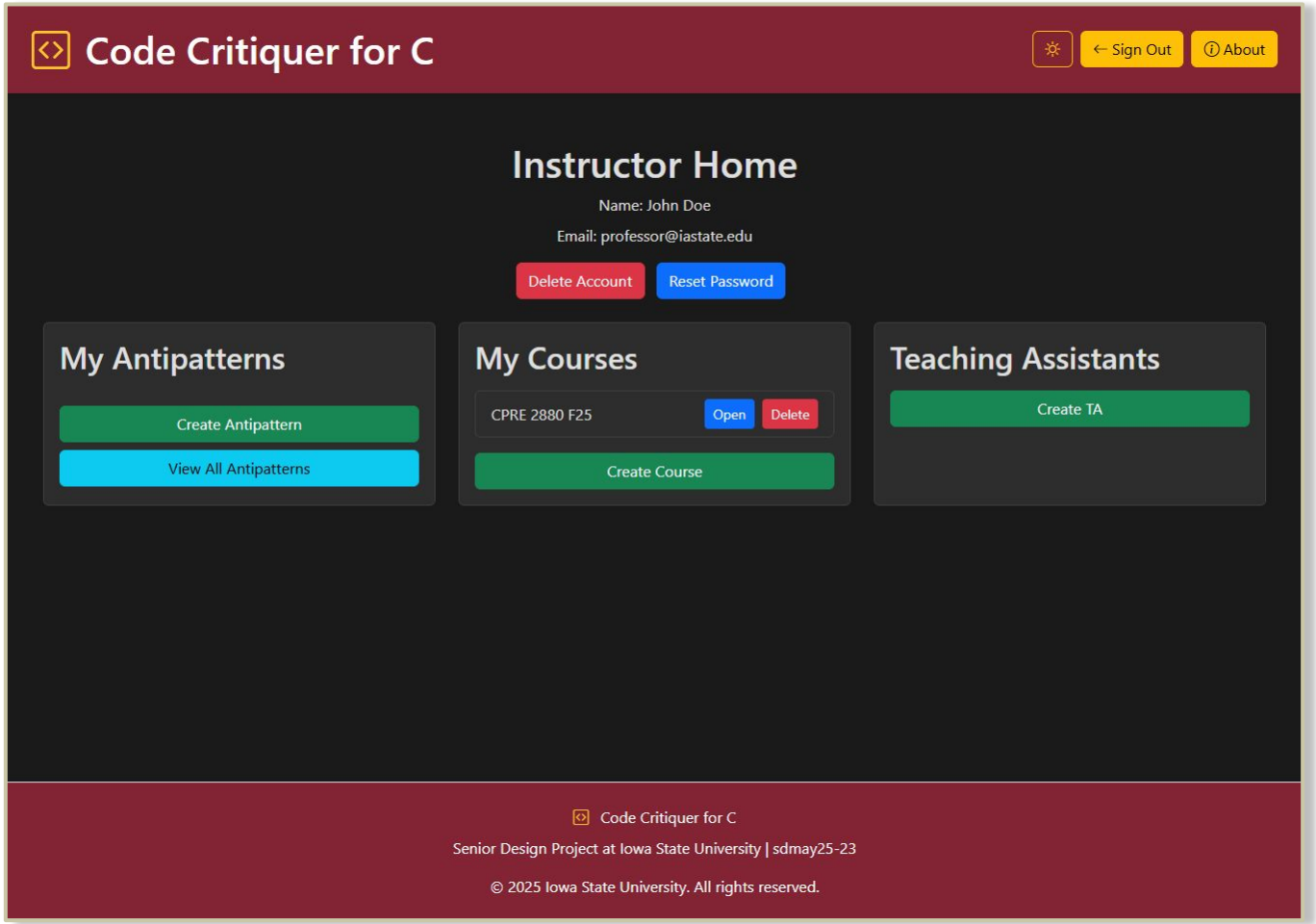


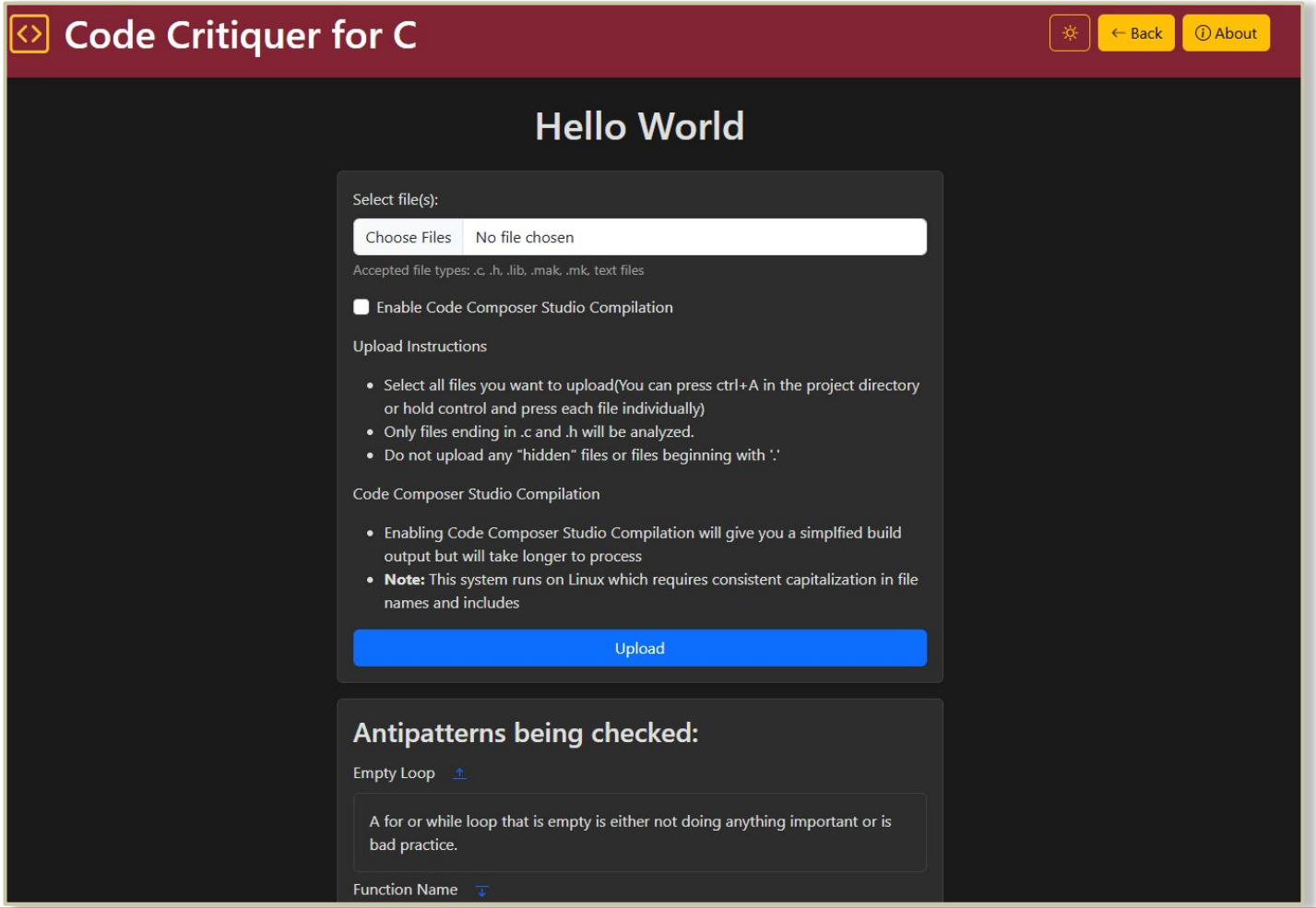
UI Design





Old → New (With Bootstrap)


Code Critiquer for C
Sign Out
About
Instructor Home
Name: John Doe
Email: professor@iastate.edu
Delete Account
Reset Password
My Antipatterns
Create Antipattern
View All Antipatterns
My Courses
CPRE 2880 F25
Open
Delete
Create Course
Teaching Assistants
Create TA
Code Critiquer for C
Senior Design Project at Iowa State University | sdmay25-23
© 2025 Iowa State University. All rights reserved.


Code Critiquer for C
← Back
About
Hello World
Select file(s):
Choose Files
No file chosen
Accepted file types: .c, .h, .lib, .mak, .mk, text files
Enable Code Composer Studio Compilation
Upload Instructions
Select all files you want to upload(You can press ctrl+A in the project directory or hold control and press each file individually)
Only files ending in .c and .h will be analyzed.
Do not upload any "hidden" files or files beginning with '.'
Code Composer Studio Compilation
Enabling Code Composer Studio Compilation will give you a simplfied build output but will take longer to process
Note: This system runs on Linux which requires consistent capitalization in file names and includes
Upload
Antipatterns being checked:
Empty Loop
A for or while loop that is empty is either not doing anything important or is bad practice.
Function Name

Code Critiquer Feedback

Assignment:	Hello World
Instructor:	John Doe
Critique Created:	05/04/2025, 13:30:37
Critiqued Files:	test.c
Excluded Files:	
Summary:	**Code Critiquer in C found 6 issues with your code. (See below.)** - There are 3 critical issues in your code. These issues must be fixed before your code will work as intended. - There are 3 non-critical concerns about your code. These issues should be addressed to make sure your code is more robust and maintainable.
AST Generation Diagnostics:	Error test.c redefinition of 'main'
Compiler Output:	Errors for project '05-04-2025--13-30-21-1' (5): gmake: *** [test.obj] Error 1 gmake: Target 'all' not remade because of errors. test.c [line 13]: #249 function "main" has already been defined test.c [line 15]: #29 expected an expression test.c [line 15]: #20 identifier "j" is undefined Warnings for project '05-04-2025--13-30-21-1' (2): test.c [line 5]: #179-D variable "i" was declared but never referenced test.c [line 29]: #1-D last line of file ends without a newline
Instructor Tests:	File not found

Critiques

#	File	Start	Code	Critique	Severity ⓘ	Comment/Question for Professor ⓘ
1	test.c	3	`main`	The main function has been declared twice in the provided code. This prevents the compiler from knowing where to start the program. Please ensure that you are only using the desired main and delete or rename the other	Critical	Comment
2	test.c	13	`main`	The main function has been declared twice in the provided code. This prevents the compiler from knowing where to start the program. Please ensure that you are only using the desired main and delete or rename the other	Critical	Comment
6	test.c	25	`k==1`	Should not directly compare floating point numbers.	Critical	Comment
3	test.c	15	`for(int j = 0; j < 10; j++){}`	Loop that contains nothing inside its body - {}	Non-Critical	Comment
4	test.c	15	`for(int j = 0; j < 10; j++){}`	Loop that contains nothing inside its body - {}	Non-Critical	Comment
5	test.c	20	`badFunction`	Function Name not in snake_case	Non-Critical	Comment

Hello World

Download Assignment Data | Download Student Comments

Number of submissions: 9

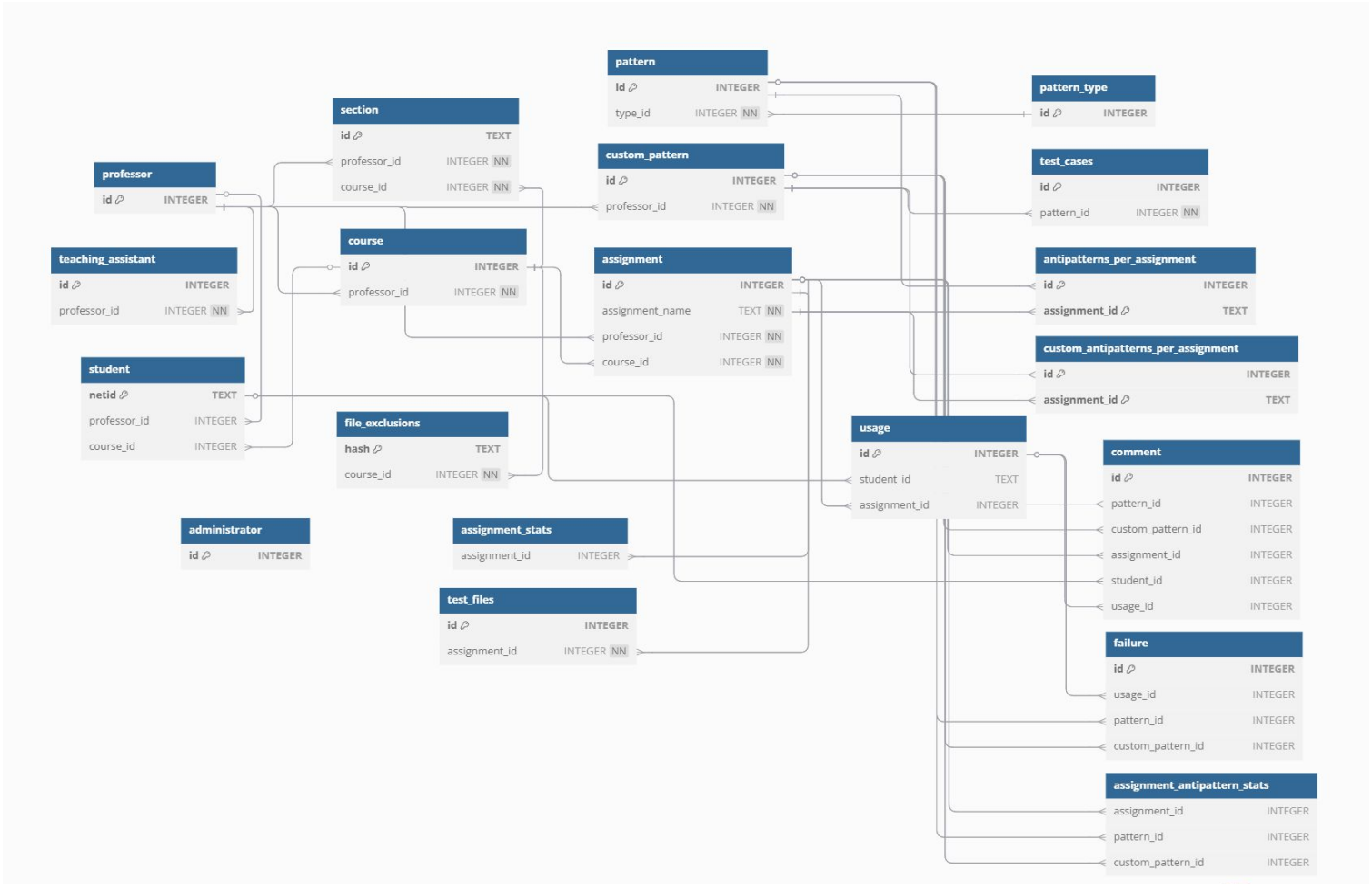
Number of students: 2

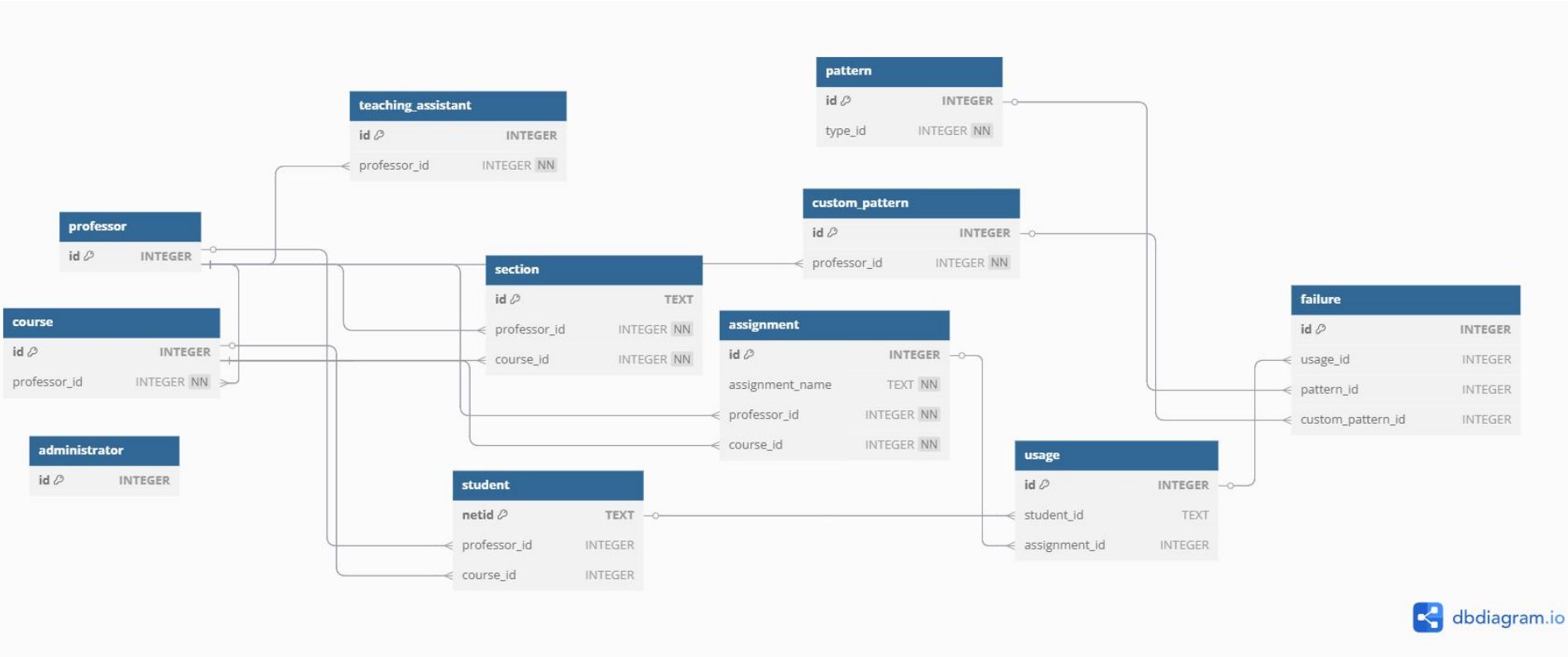
Average number of submissions per student: 4.5

Number of comments: 1

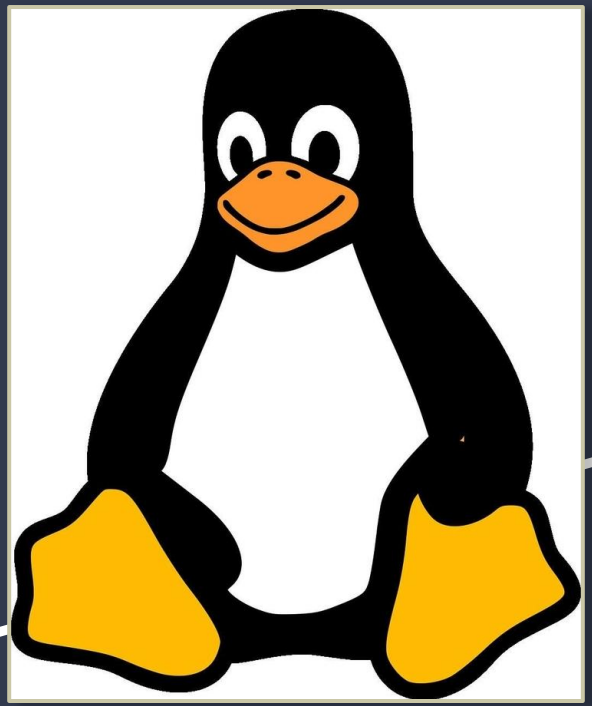
Antipattern	Number of Occurrences	Percent of Submissions
Empty Loop	9	100.0
Function Name	0	0.0
Incorrect Print Function	0	0.0
Direct Floating Point Comparison - Type 1	0	0.0
Assignment in an if statement	0	0.0
Usage of true of false	9	100.0
Direct Floating Point Comparison - Type 2	0	0.0
Recursive Functions Need Base Case	0	0.0
Two main functions declared	0	0.0
Timer Init not called before other timer functions	0	0.0

Database Table Diagram


pattern
id INTEGER
type_id INTEGER NN
pattern_type
id INTEGER
section
id TEXT
professor_id INTEGER NN
course_id INTEGER NN
custom_pattern
id INTEGER
professor_id INTEGER NN
test_cases
id INTEGER
pattern_id INTEGER NN
professor
id INTEGER
course
id INTEGER
professor_id INTEGER NN
teaching_assistant
id INTEGER
professor_id INTEGER NN
assignment
id INTEGER
assignment_name TEXT NN
professor_id INTEGER NN
course_id INTEGER NN
antipatterns_per_assignment
id INTEGER
assignment_id TEXT
custom_antipatterns_per_assignment
id INTEGER
assignment_id TEXT
student
netid TEXT
professor_id INTEGER
course_id INTEGER
file_exclusions
hash TEXT
course_id INTEGER NN
usage
id INTEGER
student_id TEXT
assignment_id INTEGER
comment
id INTEGER
pattern_id INTEGER
custom_pattern_id INTEGER
assignment_id INTEGER
student_id INTEGER
usage_id INTEGER
administrator
id INTEGER
assignment_stats
assignment_id INTEGER
test_files
id INTEGER
assignment_id INTEGER NN
failure
id INTEGER
usage_id INTEGER
pattern_id INTEGER
custom_pattern_id INTEGER
assignment_antipattern_stats
assignment_id INTEGER
pattern_id INTEGER
custom_pattern_id INTEGER

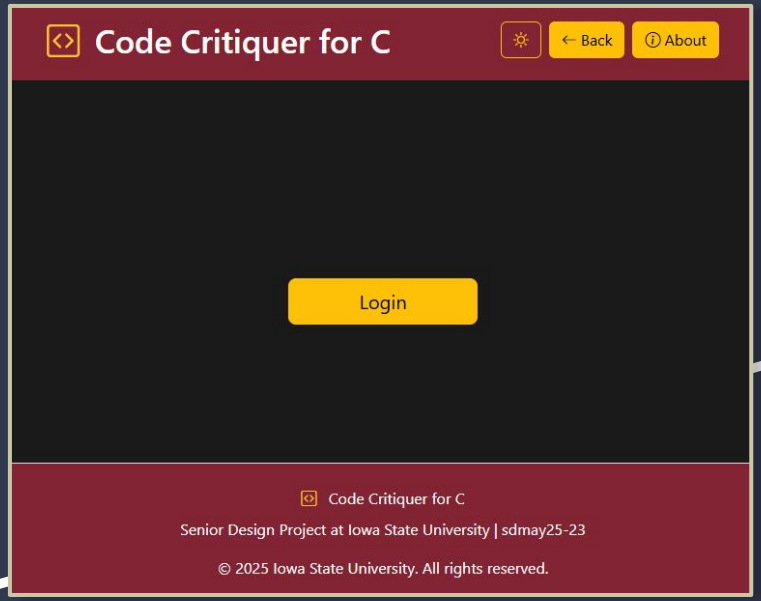

pattern
id INTEGER
type_id INTEGER NN
teaching_assistant
id INTEGER
professor_id INTEGER NN
professor
id INTEGER
custom_pattern
id INTEGER
professor_id INTEGER NN
section
id TEXT
professor_id INTEGER NN
course_id INTEGER NN
failure
id INTEGER
usage_id INTEGER
pattern_id INTEGER
custom_pattern_id INTEGER
course
id INTEGER
professor_id INTEGER NN
assignment
id INTEGER
assignment_name TEXT NN
professor_id INTEGER NN
course_id INTEGER NN
administrator
id INTEGER
usage
id INTEGER
student_id TEXT
assignment_id INTEGER
student
netid TEXT
professor_id INTEGER
course_id INTEGER
dbdiagram.io

Hardware Platform



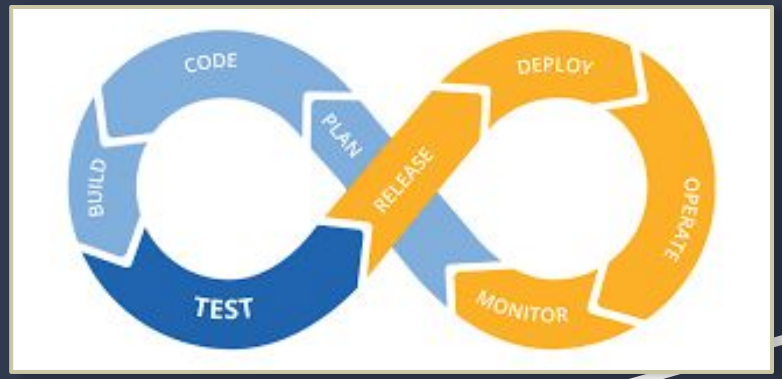
- Linux server
  - Used to host a website for the users to access
  - Holds the database containing system data

Software Platform



- HTML, CSS, Bootstrap, and Jinja
  - Website Design
- Flask
  - Handles routing of pages in the web application
- Regular Expressions/XPath
  - Used for static analysis
- Docker

Testing



- Continuous Integration and Continuous Delivery (CI/CD)
  - Automated tests that run before updating production code
- Electronics and Technology Group Server
  - Manual testing
- User Feedback
  - Fix issues discovered by users

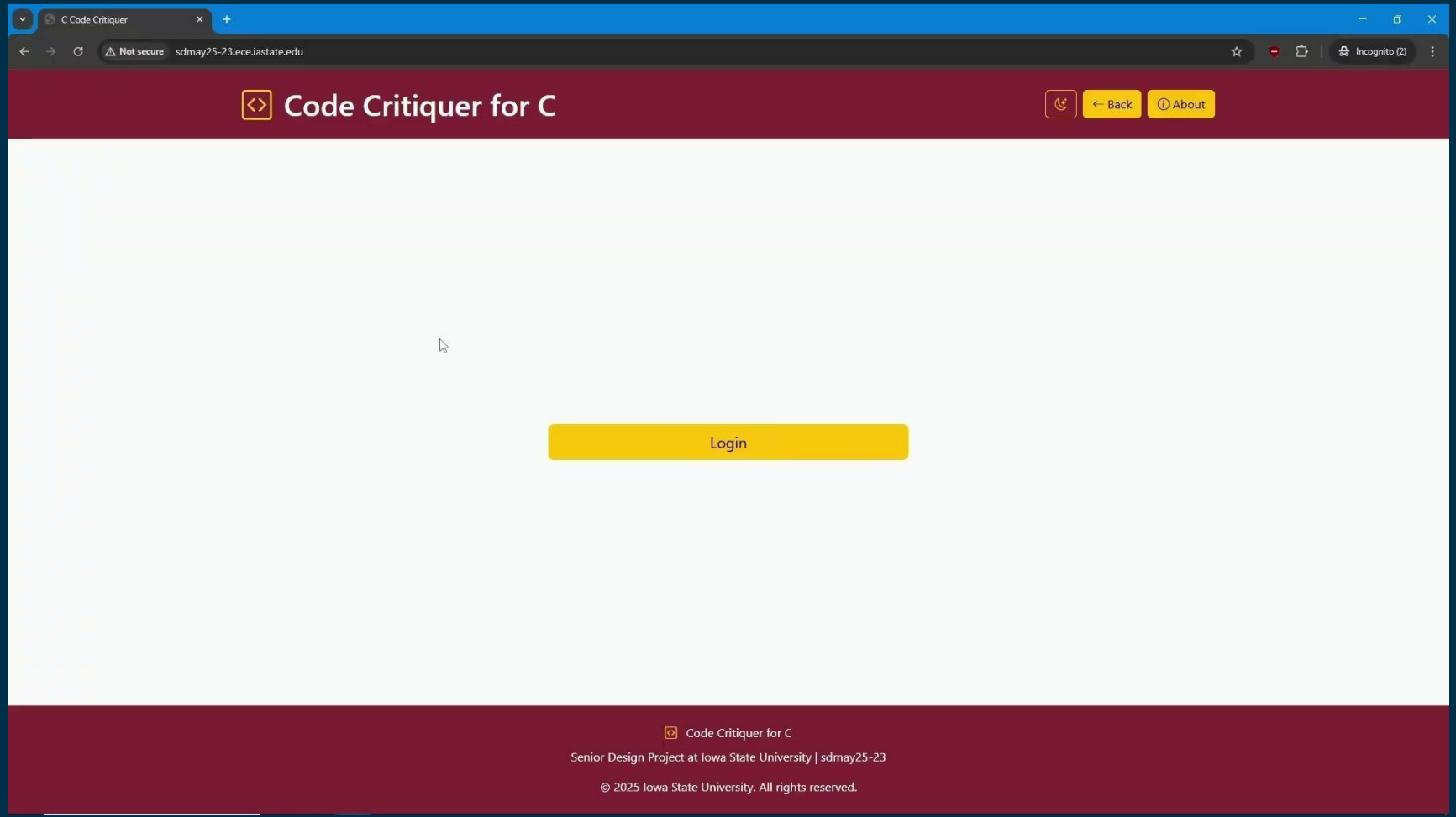
Project Accomplishments

- Got previous project running
- Overhauled UI
- Improved static analysis + Code Composer Studio Integration
- Added more antipatterns
- User account hierarchy (Student, TA, Professor, Admin)
- Analytics and Downloadable Reports
- Added courses
- Account registration

Next Steps

- First trial run
- More bug fixes
- Frontend framework
- Dynamic Analysis
- More analytics
- Microsoft Authentication Library
- Generalization

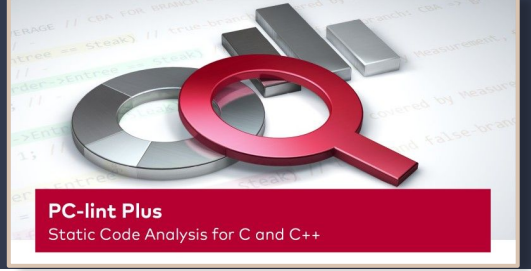
Any Questions, Comments, or Suggestions?


C Code Critiquer
Not secure
sdmay25-23.ece.iastate.edu
Incognito (2)
Code Critiquer for C
Back
About
Login
Code Critiquer for C
Senior Design Project at Iowa State University | sdmay25-23
© 2025 Iowa State University. All rights reserved.

Link to Team Website


Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14
Week 15
Week 16
Week 17
Collect User Feedback
New roles
Instructor Invitations
TA Role
Class Sections
Static Analysis
Lab-based Antipatterns
Code Composer Studio Integration
Analytics Page
Dashboard
Improved Feedback
Filtering
Downloadable Reports
UI Update
Create Screen Sketches
Prototype new UI
Implement new UI design
Finish Design
Clean up code
Document for future teams

PC-lint Plus



Pros:

- Professional grade tool
- Identifies security issues as well as bad coding practices
- Undergone rigorous auditing and testing

Cons:

- Costs money
- Restrictive licensing

CodePal.AI

Pros:

- It gives detailed feedback on:
  - Syntax/structure
  - Code readability
  - Functionality
- Visually appealing feedback

Cons:

- AI can be unreliable
- Advanced features cost money
- Limited number of queries at free level