



Dhirubhai Ambani University

Formerly known as

Dhirubhai Ambani Institute of Information and Communication Technology

IT314 Software Engineering
Group 17

Black-box Testing Report



Coding Learning

AI Roadmap Generation Platform

Prof : Surabh Tiwari
Group Mentor : Utsav

INDEX:-

Summary.....	3-4
AUTH	4-9
AI CHATBOT	9-11
ONLINE IDE	12- 16
EVIDENCE	17-23
ACCEPTANCE VERDICT.....	24-24

Authored By :- Harsh Chanchlani (202301152)

Executive Summary

This document summarizes the results of Black-box QA testing conducted on the Online Coding Learning Platform. The evaluation covered the main user workflow, including signup and authentication, learning modules, quizzes, chat assistance, and profile management.

Overall, the platform demonstrates solid progress, and several modules are functioning reliably. However, a few critical issues were identified that must be addressed before the platform can be considered ready for full release.

Modules Functioning Well

The following modules worked as expected during testing:

- **Roadmap Generation:** Input handling and output structure were accurate and consistent.
- **AI Chatbot:** Provided relevant responses, stayed within assigned boundaries, and behaved reliably.
- **Statistics and Timer Features:** Streak tracking, timing tools, progress indicators, and related features operated correctly.

Critical Issues Requiring Fixes

Several high-impact defects were found that affect the user experience and platform reliability:

1. **Quiz System Errors:** Regenerating quizzes triggers automatic selection of previous answers, resulting in incorrect scoring. The system also fails to store and display past quiz responses properly.
2. **Authentication Logic Issue:** An invalid email correctly triggers an error notification, but the user is still redirected to a verification page, which causes confusion.
3. **Password Validation Weaknesses:** Password rules are incomplete, allowing unnecessarily long passwords and weak patterns with low complexity.

Known Limitations and Minor Issues

These issues do not block launch but should be scheduled for future improvement:

- **Online IDE:** The Save shortcut downloads the file instead of saving it in the cloud, and inter-file imports are not supported.
- **User Profile Module:** Profile image uploads fail even when valid formats are used.
- **Notes Feature:** Notes allow empty submissions, have no word limit, and formatting such as lists and bullet points does not always render correctly.

AUTH

Equivalence Class Partitioning (ECP)

To ensure broad coverage without redundant testing, we applied Equivalence Class Partitioning. This method divides input data into valid and invalid classes, assuming that if one test case in a class passes/fails, others in the same class will behave similarly.

Input Field	Valid Class (Positive)	Invalid Class (Negative)
Full Name	E1 : length 3–35, contains alphabets, numbers, underscores, and spaces	E2 : contains special characters other than underscores E3 : Length < 3 E4 : Length > 35 E5 : Contains only whitespace E6 : Empty string
Email Address	E7 : Standard email format (e.g., user@test.com)	E8 : Missing @ symbol E9 : Missing domain (e.g., user.com) E10 : Empty field
Password	E11 : 6-100 characters , contains One upper case letter,number,special character	E12 : Empty String E13 : password < 6 char or password > 100 char E14 : does not contain any uppercase letter E15 : does not contain any number E16 : does not contain any special character

Confirm Password	E17 : matches password exactly and password is valid	E18 : Does not match password E19 : empty string
------------------	--	---

SIGN UP:-

Test Case ID	Test Scenario	Test Data	Expected Result	Actual Result	Status
SP_01	Verify Valid Registration (Happy case)	Name: John Doe Email: valid@test.com Pass: Test@123 Confirm pssword : Test@123	Account created, redirects to Dashboard. Equivalence class coverd : {E1, E7 , E11 , E17, E19 }	Account created, redirected to Dashboard.	PASS
SP_02	Verify Empty Fields	[Left Blank]	Show "Required" errors, stay on page. Equivalence class coverd : {E6, E10 , E12 , E19 }	User Cannot submit the form , Button is freezed till any one feild is empty	PASS
SP_03	Verify Invalid Full name Format	Full name : "ab", "harsh@" "jkqBFJQB EF OUQHFOUHF OQUHFOUFH QOUFHOQUH"	Shows "Required" errors Equivalence class coverd : {E2, E3 , E4 , E5 }	Shows "Required" errors Equivalence class	PASS
SP_04	Verify Invalid Email Format	Email: johndoe.com, Harsh@com	Shows Error : "Inavlid Email format" {E8, E9}	Error : "Invalid Email address"	PASS
SP_05	Verify Invalid Password Format	Password : "ab", " ", "harsh@ch", "harsh123", "Kajbdvjbouw bowvbouvbw uvbouwbvoub owudbvoubvo uwbdvobv"	User cannot Press sign up button till it enters a valid password Equivalence class coverd : {E13, E14 , E15 , E16 }	User cannot Press sign up button till it enters a valid password	PASS

SP_06	Verify Password Mismatch	Pass: Test@123 Confirm: Test@124	Show "Mismatch" error, stay on page. Equivalence class covered : {E18}	Showed error, stayed on page.	PASS
SP_07	Verify Password Visibility	Click Eye Icon	Password text becomes visible.	Password text became visible.	PASS
SP_08	Verify Google Login	Click Google Btn	Redirect to Google Auth.	Redirected to Google Auth.	PASS
SP_09	Verify GitHub Login	Click GitHub Btn	Redirect to GitHub Auth.	Redirected to GitHub Auth.	PASS
SP_10	Verify Sign In Link	Click 'Sign in'	Redirect to Login Page.	Redirected to Login Page.	PASS

SIGN IN:-

Equivalence Class Partitioning (ECP)

Email Address	E1 : Standard email format (e.g., user@test.com)	E2 : Missing @ symbol E3 : Missing domain (e.g., user.com) E4 : Empty field
Password	E5 : 6-100 characters , contains One upper case letter,number,special character	E6 : Empty String E7 : password < 6 char or password > 100 char E8 : does not contain any uppercase letter E9 : does not contain any number E10 : does not contain any special character

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
SI_01	Verify Valid Login	1. Enter registered Email. 2. Enter correct Password. 3. Click 'Sign In'.	Email: student@test.com Pass: Test@123	System redirects user to the Dashboard/Home page.	System redirected user to the Dashboard/Home page.	PASS
SI_02	Verify Invalid Email Format	Email: johndoe.com, Harsh@com,	Show error, block submission, stay on page. Equivalence class coverd : {E2, E3 }	Error : "Invalid Email address"	Error : "Invalid Email address"	PASS
SI_03	Verify Invalid Password Format	Password : "ab", " ", "harsh@ch", "harsh123", "Kajbdvjbowbow vbouvbwouvbowv bvoubowudbvoub vouwbdvobv"	User cannot Press sign up button till it enters a valid password Equivalence class coverd : {E7, E8 , E9}	User cannot Press sign up button till it enters a valid password	User cannot Press sign up button till it enters a valid password	PASS
SI_04	Verify Unregistered Email	1. Enter an email that is not in the database. 2. Enter any password. 3. Click 'Sign In'.	Email: unknown@test.com Pass: 123456	Shows error message: "User not found" or "Invalid credentials".	Showed error message: "User not found".	PASS
SI_05	Verify Empty Fields	1. Leave Email & Password fields blank. 2. Click 'Sign In'.	[Blank]	Error messages appear: "Email is required" / "Password is required". Equivalence class coverd : {E4, E6 }	Error messages appeared correctly.	PASS
SI_06	Verify Password Visibility	1. Type password. 2. Click the 'Eye' icon.	Pass: secret123	Password masking is removed, text becomes visible.	Password masking was removed, text became visible.	PASS

SI_07	Verify 'Forgot Password' Link	1. enter a email address 2. Click on Forgot password Button	Email : "harshc2710@gmail.com"	User is given a message that reset password link is send on his(entered) email id	User is given a message that reset password link is send on his(entered) email id	PASS
SI_08	Verify Google Login	1. Click 'Continue with Google'.	N/A	Opens Google authentication window/pop-up.	Opened Google authentication window.	PASS
SI_09	Verify GitHub Login	1. Click 'Continue with GitHub'.	N/A	Opens GitHub authentication window/pop-up.	Opened GitHub authentication window.	PASS
SI_10	Verify 'Sign Up' Link	1. Click 'Sign up' link at the bottom.	N/A	User is redirected to the Registration (Sign Up) page.	User was redirected to the Registration page.	PASS
SI_11	Verify Forgot password with unregistered email	1. enter a email address 2. Click on Forgot password Button	Email : "Test123@example.com"	Error message should be displayed "User not found "	Error message is displayed "User not found "	PASS

Reset Password :-

EQUIVALENCE CLASS :-

Password	E1 : 6-100 characters , contains One upper case letter,number,special character	E2 : Empty String E3 : password < 6 char or password > 100 char E4 : does not contain any uppercase letter E5 : does not contain any number E6 : does not contain any special character
Confirm Password	E7 : matches password exactly and password is valid	E8 : Does not match password E9 : empty string

TestCases:-

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
RP_01	Verify Valid Reset Password (Happy case)	Pass: Test@123 Confirm password :	Account created, redirects to Dashboard. Equivalence class coverd : {E1, E7 , E11 , E17, E19 }	Account created, redirected to Dashboard.	PASS	SP_0_1
RP_02	Verify Invalid Password Format	Password : “ab”, ” ”, ”harsh@ch”, ”harsh123”, ”Kajbdvjbouwb owvbouvbwou vbouvbvoubo wudvboubvou wbdvobv”	User cannot Press sign up button till it enters a valid password Equivalence class coverd : {E13, E14 , E15 , E16 }	User cannot Press sign up button till it enters a valid password	User cannot Press sign up button till it enters a valid password	PASS
PR_03	Verify Password Mismatch	Pass: Test@123 Confirm: Test@124	Show "Mismatch" error, stay on page. Equivalence class coverd : {E18}	Showed error, stayed on page.	Showed error, stayed on page.	PASS

AI ChatBot

EQUIVALENCE CLASS :-

Input Type	Valid Class	Invalid Class
Prompt Content	Technical/Coding queries	<ul style="list-style-type: none"> Non-technical/General queries (e.g., "Pizza recipe") Empty or whitespace only

TESTCASES:-

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
CH_BOT_01	Verify Valid Tech Question	<ol style="list-style-type: none"> 1. Open Chat. 2. Type a coding question. 3. Send. 	Input: "How do I reverse a string in Python?"	AI responds with a Python code example and explanation.	AI responded with Python code and explanation.	PASS
CH_BOT_02	Verify Chat History	<ol style="list-style-type: none"> 1. Send a message. 2. Click "New Chat". 3. Click the previous chat in the sidebar. 	N/A	The previous conversation and messages reload correctly.	Previous conversation reloaded correctly.	PASS
CH_BOT_03	Verify 'New Chat' Button	<ol style="list-style-type: none"> 1. Be in an active chat. 2. Click "+ New Chat" button. 	N/A	The message area clears, ready for a fresh start.	Message area cleared successfully.	PASS
CH_BOT_04	Verify Code Formatting	<ol style="list-style-type: none"> 1. Ask for code. 2. Check the response style. 	Input: "Write a React component."	Response should be inside a dark/styled code block, not plain text.	Response contained properly styled code blocks.	PASS

CH_BOT_05	Verify Scroll Behavior	1. Send a message that generates a long response. 2. Wait for generation to finish.	N/A	Chat window should auto-scroll to show the latest text.	Chat window auto-scrolled to the bottom.	PASS
CH_BOT_06	Verify Empty Input	1. Leave input box empty. 2. Try to click Send or press Enter.	[Blank]	The Send button should be disabled or nothing should happen.	Button was disabled/ Nothing happened.	PASS
CH_BOT_07	Verify System Scope (Guardrail)	1. Ask a non-tech question.	Input: "What is the best pizza recipe?"	AI should either answer politely or redirect to tech topics (depending on your prompt).	AI answered/redirected correctly based on system prompt.	PASS
CH_BOT_08	Verify Markdown Rendering	1. Ask AI to use bold/ lists.	Input: "List 3 items in bold."	Output should show Bold text, not **Bold**.	Text was rendered as Bold, not raw markdown.	PASS

ONLINE IDE

EQUIVALENCE CLASS:-

Input Condition	Valid Class	Invalid Class
Code Syntax	Valid Python/C++ syntax	<ul style="list-style-type: none"> Syntax errors (missing parentheses, typos)
Execution Time	Finite execution (e.g., <code>print("Hi")</code>)	<ul style="list-style-type: none"> Infinite execution (e.g., <code>while(1)</code>) [Critical Boundary]
File Naming	Alphanumeric names with valid extensions (EX: <code>cpp</code> , <code>py</code> , <code>jsx</code> , <code>tsx</code> etc),	<ul style="list-style-type: none"> Special characters (e.g., <code>/</code>, <code>\</code>) Empty filenames
Folder Naming	Alphanumeric names	<ul style="list-style-type: none"> Special characters (e.g., <code>/</code>, <code>\</code>) Empty filenames

TESTCASES:-

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Status
IDE_01	Verify Basic Execution (Hello World)	<ol style="list-style-type: none"> Select Python language. Type print code. Click 'Run'. 	Code: <pre>print("Hello World")</pre>	Output console displays Hello World.	Output console displayed Hello World.	PASS
IDE_02	Verify User Input Handling	<ol style="list-style-type: none"> Type input code. Enter text in 'Input' box first. Click 'Run'. 	Code: <code>x = input()</code> <pre>print(x)</pre> Input Box: Testing123	Output console displays Testing123.	Output console displayed Testing123.	PASS
IDE_03	Verify Syntax Error Handling	<ol style="list-style-type: none"> Type broken code. Click 'Run'. 	Code: <pre>print("Missing paren"</pre>	Console shows a red error/traceback message (SyntaxError).	Console showed Python SyntaxError trace.	PASS

IDE_04	Verify Console 'Clear' Button	1. Run code to generate output. 2. Click 'Clear' button.	N/A	The Output console text disappears.	The Output console remained in the same state	FAIL
IDE_05	Verify File Creation with valid name and extension	1. Click the 'New File' (+) icon. 2. Name the file.	Filename: test.py	New file appears in the sidebar list.	New file appeared in the sidebar.	PASS
IDE_06	Verify file and folder creation using empty file name	1. Click the 'New File' (+) icon. 2. Name the file.	Filename: [blank]	Error Message : "File/ Folder name cannot be empty "	Error Message : "File/ Folder name cannot be empty "	PASS
IDE_07	Verify File Switching	1. Create two files. 2. Click on File A, then File B.	File A content: print("A") File B content: print("B")	Editor content changes to match the selected file.	Editor content updated correctly.	PASS
IDE_08	Verify 'Download' Code	1. Click the Download icon (top right).	N/A	A file (e.g., file1.py) is downloaded to the local computer.	File downloaded successfully.	PASS
IDE_09	Verify 'Copy' Output	1. Generate output. 2. Click 'Copy' button on console.	N/A	Output text is copied to clipboard. (Verify by pasting).	Text copied to clipboard successfully.	PASS
IDE_10	Verify Infinite Loop Protection	1. Run an infinite loop. 2. Check system response.	Code: while True: pass	System should timeout or allow 'Stop/Cancel', not crash the browser.	stopped execution safely after a finite number of loops.	PASS
IDE_11	Verify Language Selector	1.create a file with any familiar extension .	N/A	The file is created and the language is automatically selected	Correct language selected automatically	PASS

ID	Feature	Action/Steps	Expected Result	Actual Result	Status
File/Folder Management					
IDE_FM_01	Create Folder	<ol style="list-style-type: none"> Right-click on the root (or any folder) in the File Tree. Select "Create Folder". Enter a valid name (e.g., src). Click "Create". 	A new folder named src/ appears in the File Tree. Success toast message is displayed.	Succes message ; "Created Succesfully"	PASS
IDE_FM-02	Create File in Folder	<ol style="list-style-type: none"> Right-click on the newly created src folder. Select "Create File". Enter a valid name (e.g., utils.js). Click "Create". 	A new file named src/utils.js appears under the src/ folder. Success toast.	Succes Message ; "Script.js Created Succesfully"	PASS
IDE_FM-03	Rename Folder	<ol style="list-style-type: none"> Right-click on the src folder. Select "Rename". Change the name (e.g., to source). Click "Rename". 	The folder and all its contents are renamed to source/. Success toast.	Succes Message : "Src renamed Succesfully"	PASS
IDE_FM-04	Rename File	<ol style="list-style-type: none"> Right-click on file2.cpp. Select "Rename". Change the name to main.cpp. Click "Rename". 	The file is renamed to main.cpp. The open tab updates its name. Success toast.	Succes Message : "File 2 renamed succesfully"	PASS
IDE_FM-05	Delete File	<ol style="list-style-type: none"> Right-click on source/utils.js. Select "Delete". Confirm deletion in the modal. 	The file is removed from the File Tree. If the file was open, its tab closes, and focus moves to another open file or an empty editor. Success toast.	Succes Message : "utils.js deleted succesfully "	PASS
IDE_FM-06	Delete Non-Empty Folder	<ol style="list-style-type: none"> Create a file inside a test folder. Right-click the test folder. Select "Delete" Confirm deletion. 	The folder and all its contents (the nested file) are removed. Success toast.	Succes Message : "Source deleted succesfully "	PASS

IDE_FM-07	Create File - Invalid Name/ Empty	Attempt to create a file with illegal characters (e.g., file*.txt) or a reserved system name.	An error toast message is displayed. The file is not created.	An error toast message is displayed : "File Cannot contain / or \ " or "File name cannot be empty"	PASS
Code Execution & State					
IDE_CE-01	Execution Error	Write code that causes a runtime error (e.g., divide by zero). Click Run.	The Output panel displays the error message (Runtime Error, stack trace, etc.).	Output:'Floating point exception '	PASS
IDE_CE-02	Save File (Ctrl+S)	1. Modify the code in the editor. 2. Press Ctrl+S (or Cmd+S on Mac).	The file content is saved to the backend (Redux state saved flag should be true). Success toast is displayed.	The file is directly going in save as in host computer , button is working completly fine	FAIL
AI Analysis					
IDE_AI-01	Analyze Valid Code	1. Write non-trivial code (e.g., a sorting function). 2. Click Analyze.	The Analysis Modal opens. A loading spinner appears, and then the structured analysis data is displayed (e.g., Complexity, Suggestions).	The Analysis Modal opens. A loading spinner appears, and then the structured analysis data is displayed (e.g., Complexity, Suggestions).	PASS
AI-02	Analyze Empty Code	1. Ensure the current file is empty or only contains comments. 2. Click Analyze.	An error toast pops up: "File is empty. Write some code to analyze." The Analysis Modal does not open.	Error Message : "File is empty Write some code to analyze"	PASS
AI-03	Analyze Complex Code	Write a piece of code with poor structure (e.g., high cyclomatic complexity, deeply nested loops). Click Analyze.	The analysis output correctly highlights the issues in the "Suggestions" or "Complexity" fields.	The analysis will tell if there is any syntax or Compilaton error, ad it wil suggest better techniques of implementation	PASS
Inter-File Dependencies					
ID-01	Import Another File	1. Create fileA and fileB. 2. In fileA, write code to import/include fileB (e.g., Python import fileB). 3. Click Run on fileA.	The code Should execute the function in file B and give the corresponding output	Error Message : "No Module named File B"	FAIL

ID-02	Saving and Switching	1. Open fileA and modify its content. 2. Without saving, switch to fileB by clicking its tab. 3. Switch back to fileA.	The unsaved changes from Step 1 are still present in the editor for fileA.	The unsaved changes from Step 1 are still present in the editor for fileA.	PASS
-------	----------------------	--	--	--	------

EVIDENCE

AUTH :- (SIGN IN/ SIGN UP)

CodeLearn

Create Account

Join thousands of developers learning today

FULL NAME

John Doe

EMAIL ADDRESS

test123exmaple.com

Please include an '@' in the email address. 'test123exmaple.com' is missing an '@.'

Test@123

REQUIREMENTS:

- ✓ At least 6 characters
- ✓ One uppercase letter
- ✓ One lowercase letter
- ✓ One number
- ✓ One special character

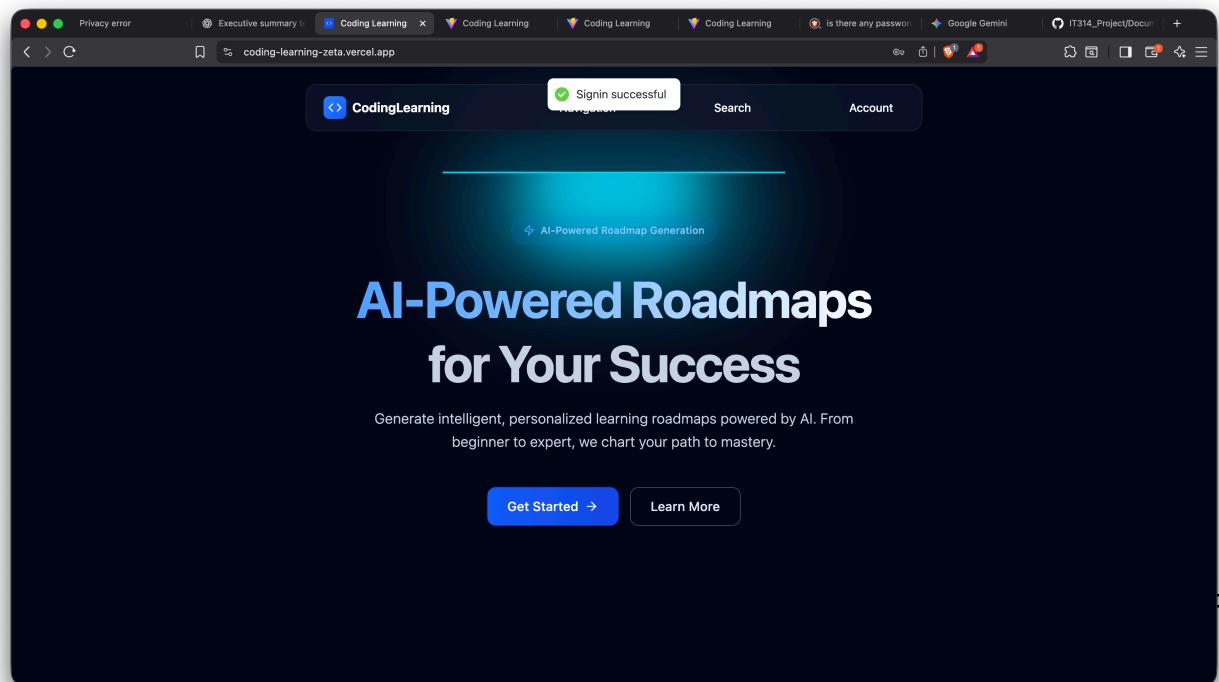
CONFIRM PASSWORD

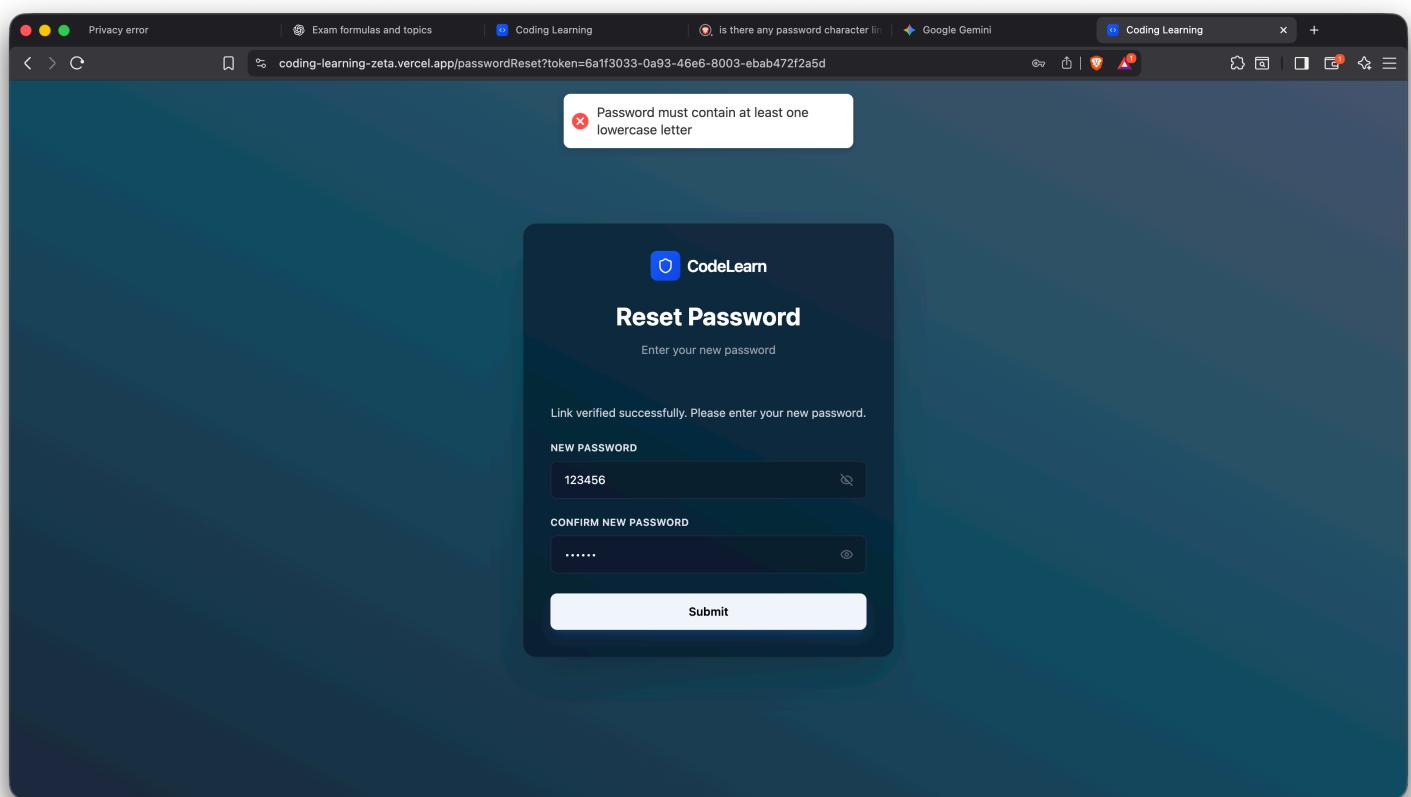
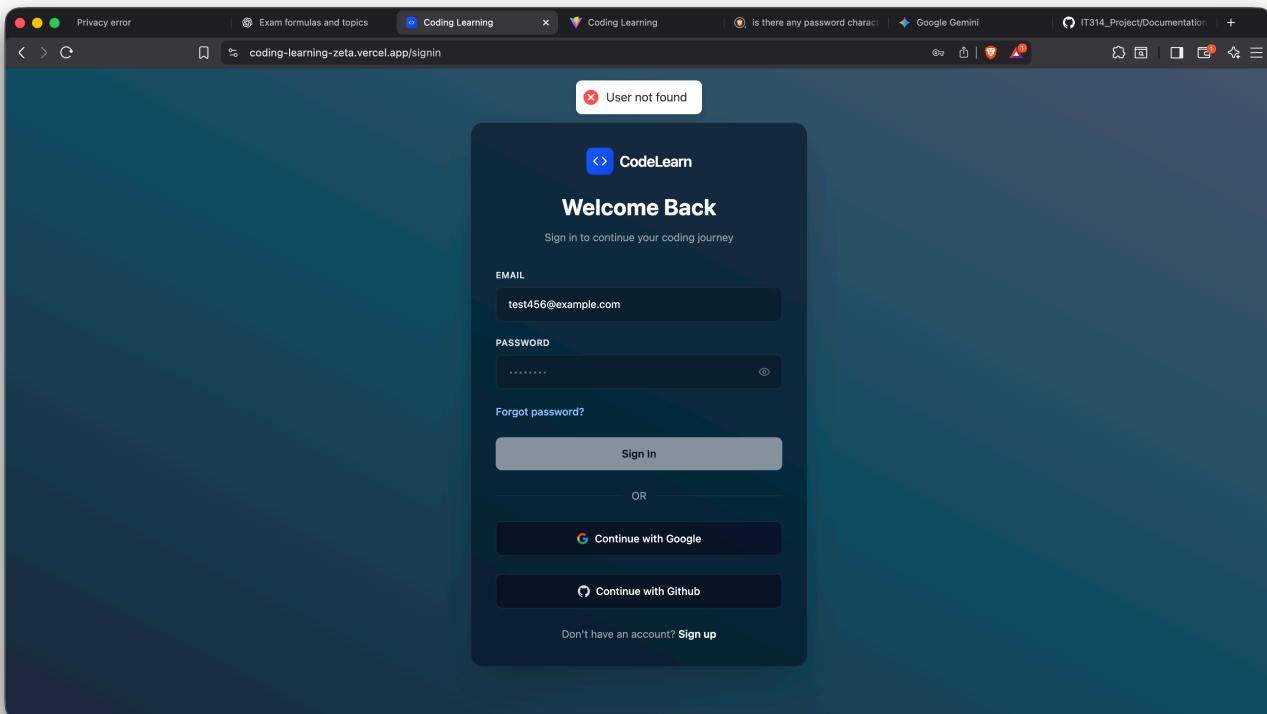
.....

Create Account

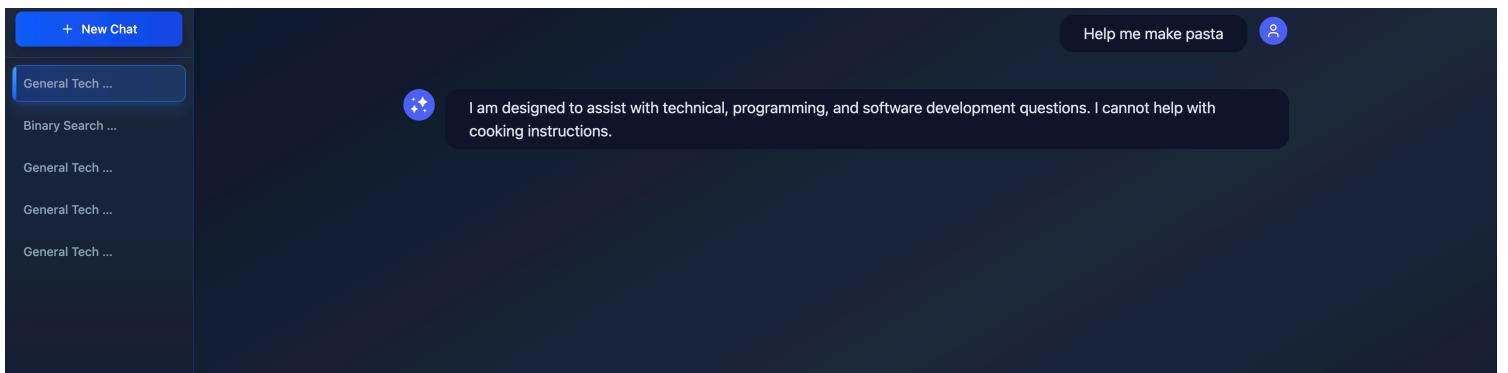
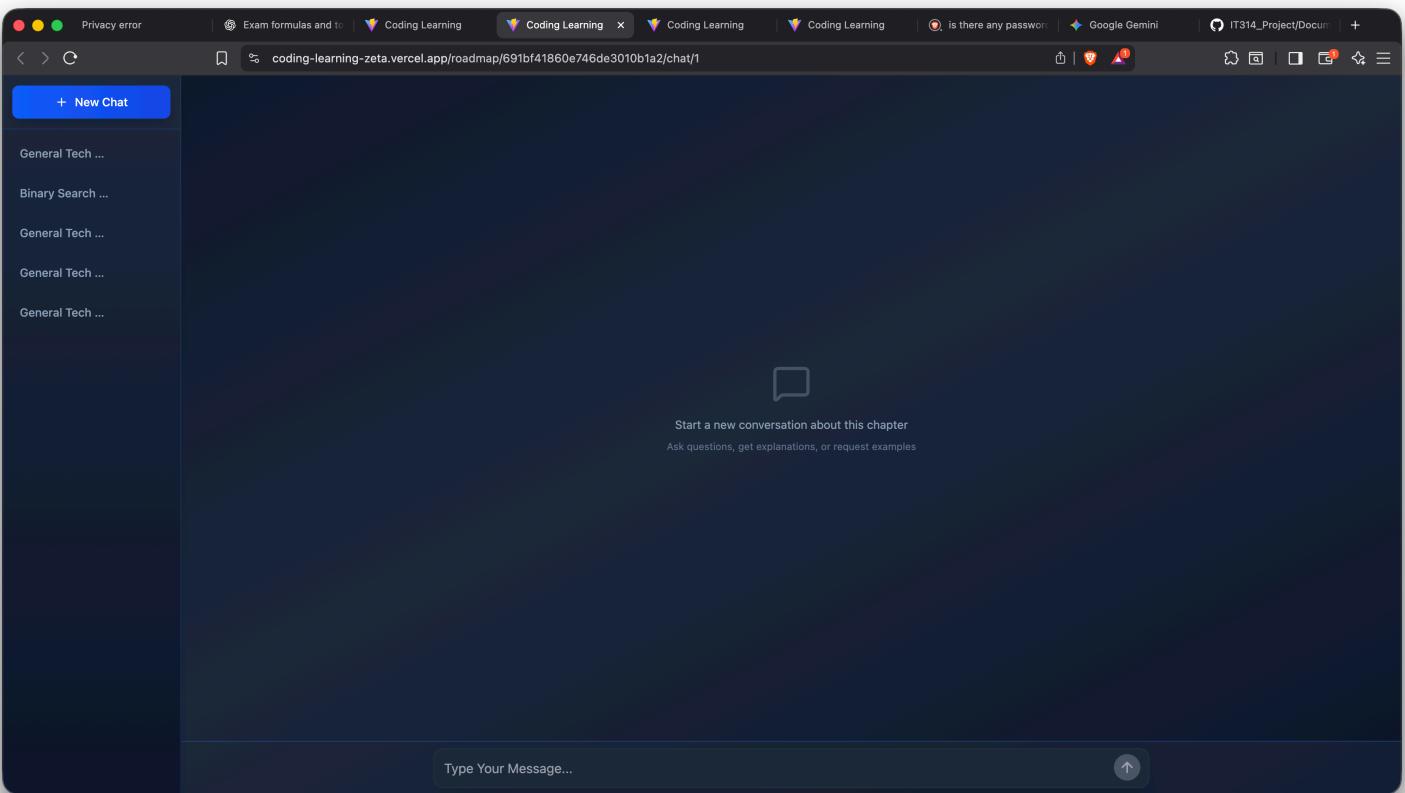
OR

Continue with Google





AI Chatbot :-



The screenshot shows a web-based coding environment. On the left, there's a sidebar with tabs for "General Tech ...". In the main area, a code editor displays Python code for binary search:

```
low = 0
high = len(arr) - 1

while low <= high:
    mid = low + (high - low) // 2 # Avoid potential overflow

    if arr[mid] == target:
        return mid
    elif arr[mid] < target:
        low = mid + 1
    else:
        high = mid - 1

return -1 # Target not found

# Example usage
arr = [2, 5, 7, 8, 11, 12]
target = 13

result = binary_search(arr, target)

if result != -1:
    print(f"Element is present at index {result}")
else:
    print("Element is not present in array")
```

Below the code, there are performance notes:

- Time Complexity:** $O(\log n)$, where n is the number of elements in the array. This is because the search interval is halved in each iteration.
- Space Complexity:** $O(1)$ for iterative implementation (as shown above) as it uses a constant amount of extra space. Recursive implementations can have a space complexity of $O(\log n)$ due to the call stack.

Prerequisites: The array *must* be sorted for binary search to work correctly. Also, it generally works best on arrays with random access (e.g., contiguous arrays).

At the bottom right, there are buttons for "Explain me segment Tree" and a user icon. A message input field with placeholder "Type Your Message..." is also visible.

Online IDE:-

The screenshot shows a web-based IDE interface. On the left, a sidebar titled "Web IDE" shows a file named "file1.py". The main area has two tabs: "Editor" and "Console".

In the "Editor" tab, the code "1" is displayed in the file "File2.py".

In the "Console" tab, there are two sections: "Input" and "Output". The "Input" section contains the placeholder "Enter input here...". The "Output" section displays the message "(no output yet)".

At the top, a message says "file1.py created successfully.". Below the tabs, there are buttons for "python", "Save", "Run", "Analyse", and a download icon.

A screenshot of a web-based IDE interface. The top navigation bar includes links for Brave, File, Edit, View, History, Bookmarks, Profiles, Tab, Window, and Help. The main window shows a dark-themed editor with a file named /file1.py containing the following Python code:

```
1 x = input("Enter Your Name:-")
2
3 print("\nHello", x)
```

The right side of the interface features a "Console" section with two tabs: "Input" and "Output". In the "Input" tab, the user has entered "Harsh". In the "Output" tab, the application has printed "Hello Harsh".

A screenshot of a web-based IDE interface. The top navigation bar includes links for Brave, File, Edit, View, History, Bookmarks, Profiles, Tab, Window, and Help. The main window shows a dark-themed editor with a file named /file2.cpp containing the following C++ code:

```
5 int main(){
6     while(1){
7         cout<<"Hi Harsh"<<endl;
8     }
9     return 0;
10 }
```

An error message "File name cannot contain / or \\" is displayed above the editor. A "Create New File" dialog box is overlaid on the editor, prompting the user to enter "file*txt". The "Create" button is highlighted in blue.

A screenshot of a web-based IDE interface, likely Om1201/IT314_Project, running in a Brave browser window. The IDE has a dark theme and includes a sidebar, an editor, a console, and a file manager.

File Manager: Shows files `file.cpp` and `file2.cpp`. `file2.cpp` is currently selected.

Editor: Displays the following C++ code in `file2.cpp`:

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main(){
6
7     while(1){
8         cout<<"Hi Harsh"<<endl;
9     }
10    return 0;
11 }
```

A message box at the top right indicates "source delete successfully."

Console:

- Input:** Placeholder "Enter input here..."
- Output:** The output window shows the string "Hi Harsh" repeated 15 times, followed by an ellipsis (...).

The browser's address bar shows the URL `localhost:5173/ide/691371fa039cd2f40155ef6c`.

The system tray at the bottom shows various application icons, including Mail, Safari, and Finder.

A screenshot of a web-based IDE interface titled "Web IDE". The interface is dark-themed and includes a navigation bar with "Brave", "File", "Edit", "View", "History", "Bookmarks", "Profiles", "Tab", "Window", and "Help" options. The top status bar shows battery level at 66%, signal strength, and the date "Wed Nov 26 12:44 AM".

The main workspace consists of several panels:

- FILES** panel on the left with a search bar and file list containing "file1.py" and "File2.py" (which is currently selected).
- Editor** panel showing two files: "file1.py" and "File2.py". "file1.py" contains the following code:

```
1 import file1
2
3 print(analyze_prime_chaos)
```
- Console** panel on the right with "Input" and "Output" sections. The "Input" section has a placeholder "Enter input here...". The "Output" section displays a traceback:

```
Traceback (most recent call last):
  File "/piston/jobs/9c8a77c2-3476-406d-bcf1-c0cd37a36729/File2.py", line 1,
    import file1
ModuleNotFoundError: No module named 'file1'
```


The bottom dock contains various application icons, including Mail, Safari, News, Wallet, iMessage, Home, Calendar, Stocks, Notes, Reminders, Stocks, Maps, Wallet, App Store, Safari, Spotify, Xcode, and a trash can.

Brave File Edit View History Bookmarks Profiles Tab Window Help

Om1201/IT314_Project Coding Learning Coding Learning Google Gemini New Tab

localhost:5173/de/691371fa039cd2f40155ef6c

AI Code Analysis
Powered by Gemini

TIME EXPLANATION
The binarySearch function operates by repeatedly dividing the search interval in half. In each step, the middle element is compared with the target value 'x'. If 'x' is found, the index is returned. If 'x' is greater than the middle element, the search continues in the right half; otherwise, the search continues in the left half. This halving of the search interval results in a logarithmic time complexity, where n is the size of the input array.

SPACE COMPLEXITY
 $O(1)$

SPACE EXPLANATION
The binarySearch function uses a constant amount of extra space, regardless of the input size. It only uses a few integer variables (low, high, mid, result, x, n) for its operations. Therefore, the space complexity is $O(1)$, indicating constant space usage.

SUGGESTIONS
No optimizations possible. (no suggestions)

Done



Brave File Edit View History Bookmarks Profiles Tab Window Help

Om1201/IT314_Project Coding Learning Coding Learning Google Gemini New Tab

localhost:5173/de/691371fa039cd2f40155ef6c

AI Code Analysis
Powered by Gemini

COMPILE ERROR
 No

ERROR EXPLANATION
No syntax or logical errors.

TIME COMPLEXITY
 $O(\log n)$

TIME EXPLANATION
The binarySearch function operates by repeatedly dividing the search interval in half. In each step, the middle element is compared with the target value 'x'. If 'x' is found, the index is returned. If 'x' is greater than the middle element, the search continues in the right half; otherwise, the search continues in the left half. This halving of the search interval results in a logarithmic time complexity, where n is the size of the input array.

SPACE COMPLEXITY
 $O(1)$

Done



Acceptance Testing Verdict

Here is a acceptance level verdict for each core feature

Module	Test Cases Executed (Total)	Fails Identified	Overall Status	Key Verdict / Next Steps
Authentication (AUTH)	32 (Sign-Up: 10, Sign-In: 11, Reset Password: 11)	0	Mostly Stable (Pending Fix)	Stable, but still has the critical logic flow defect where an invalid email incorrectly redirects to the "Verify Link" page ¹ . Fix is mandatory for production sign-off.
AI ChatBot	8	0	Stable & Production-Ready	All core functional test cases passed ² . The "Placeholder Triggers" were correctly descoped (removed per your request) ³ .
Online IDE	16	3	Ready with Known Limitations	The IDE is now considered ready, with the following known and acceptable limitations : 1. Inter-File Support is technically blocked by the Piston API and is accepted ⁴ . 2. Clear Console is a minor fix/implementation pending ⁵⁵⁵ . 3. Save (Ctrl+S) functionality is not working , but user can save the file using save file button on IDE page

The webiste is almost production ready and has no big failues/ bugs requires some minor bug fixing