

WEB TECH MINI PROJECT

Murder Mystery Game

SHRREYA NAGARAJ - PES1UG23CS569

SHRUJANNA M - PES1UG23CS570

SPURTHI RAGHOTHAMA - PES1UG23CS590

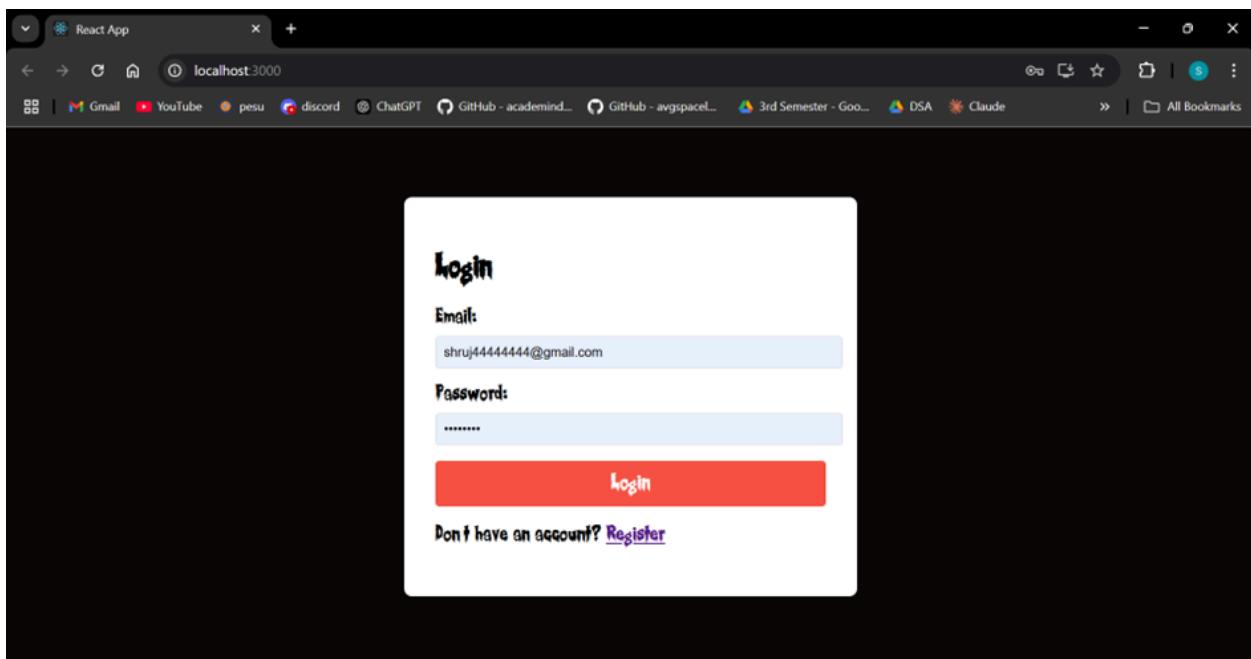
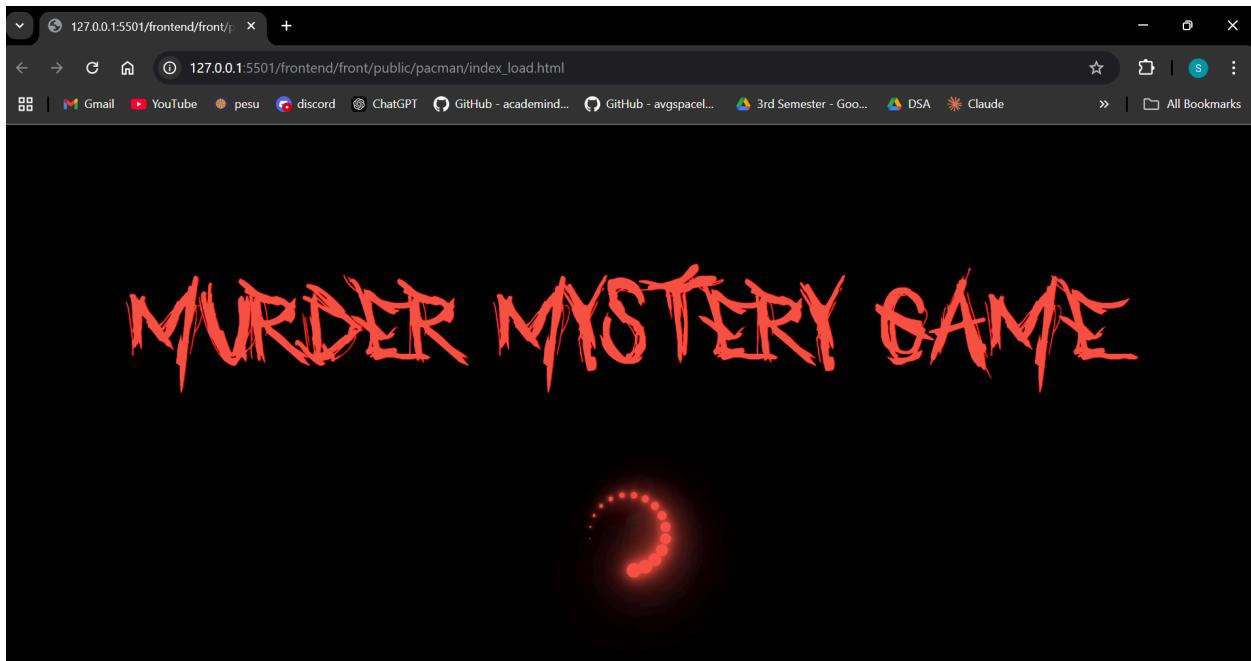
The "Murder Mystery Game" is a web application featuring three levels: Tic Tac Toe, Memory Matching, and Pac-Man.

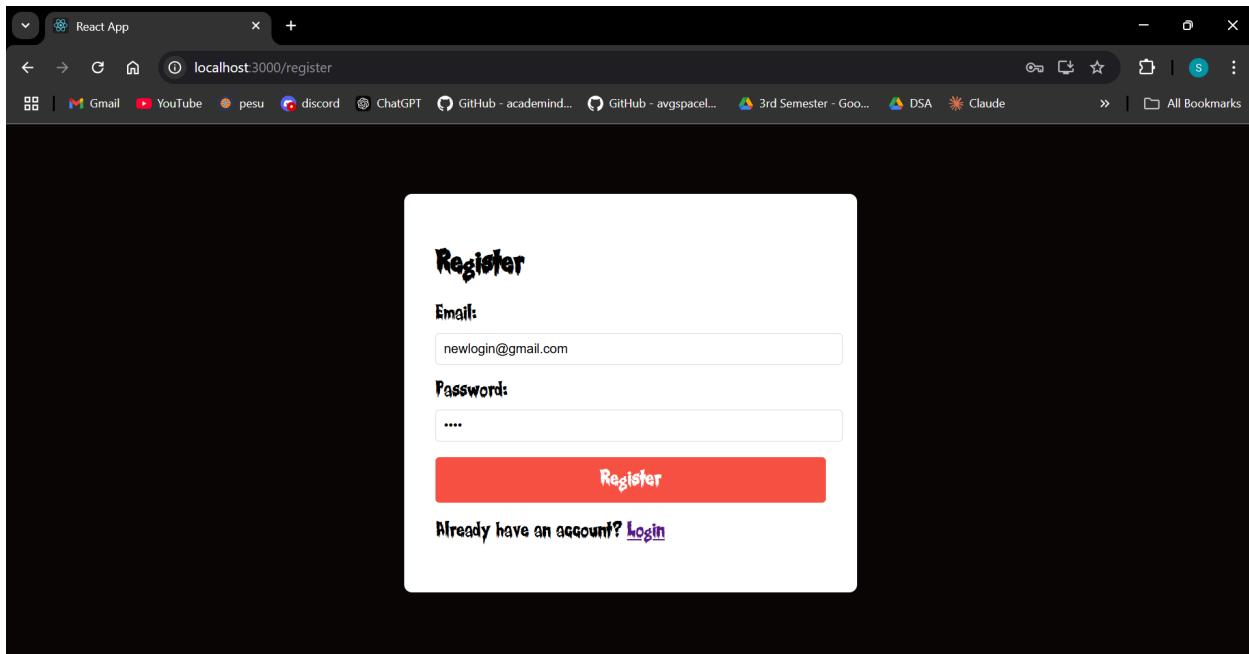
The login and signup functionality is implemented using the MERN stack, with React for the frontend, Node.js and Express for the backend, and MongoDB for user authentication and data management.

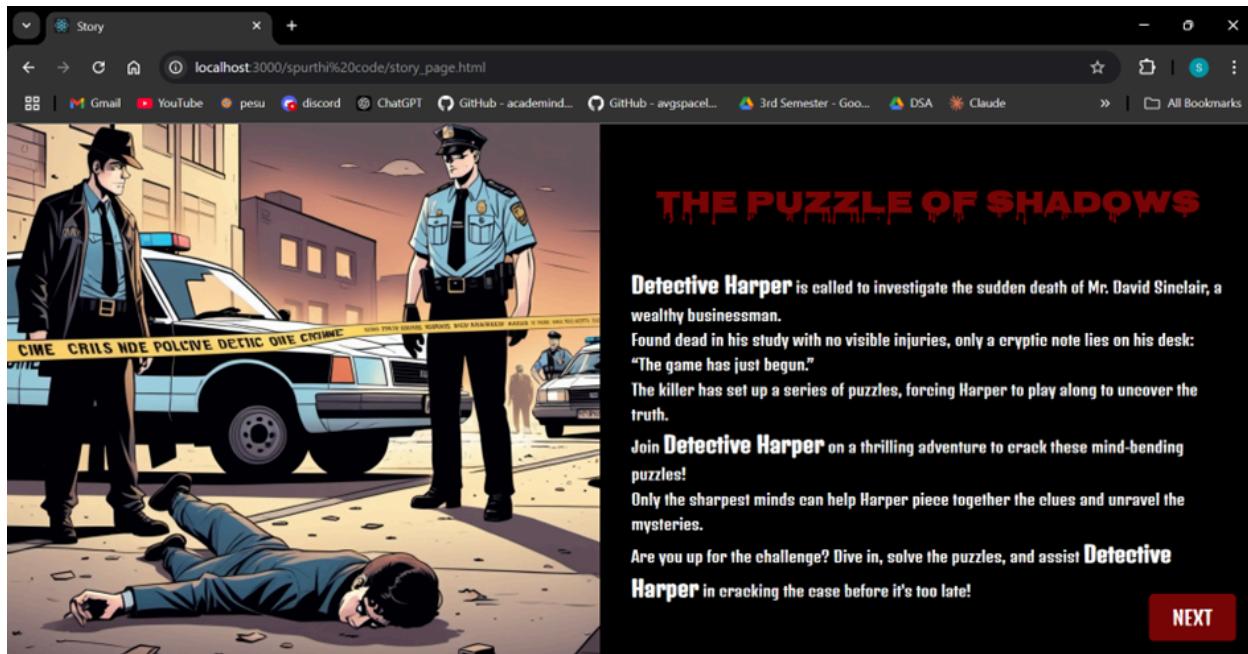
The game levels are developed using HTML, CSS, and JavaScript to provide interactive and engaging gameplay. The remaining pages of the application, such as the victory and failure pages, are designed using HTML and CSS.

The feedback page, built with HTML and CSS for the frontend and Node.js with Express for the backend, allows players to submit feedback that is stored in MongoDB. Players must complete all three levels to win; failing at any stage sends them back to Level 1.

After submitting feedback, users have the option to replay the game or can choose to exit, creating a dynamic and enjoyable experience.



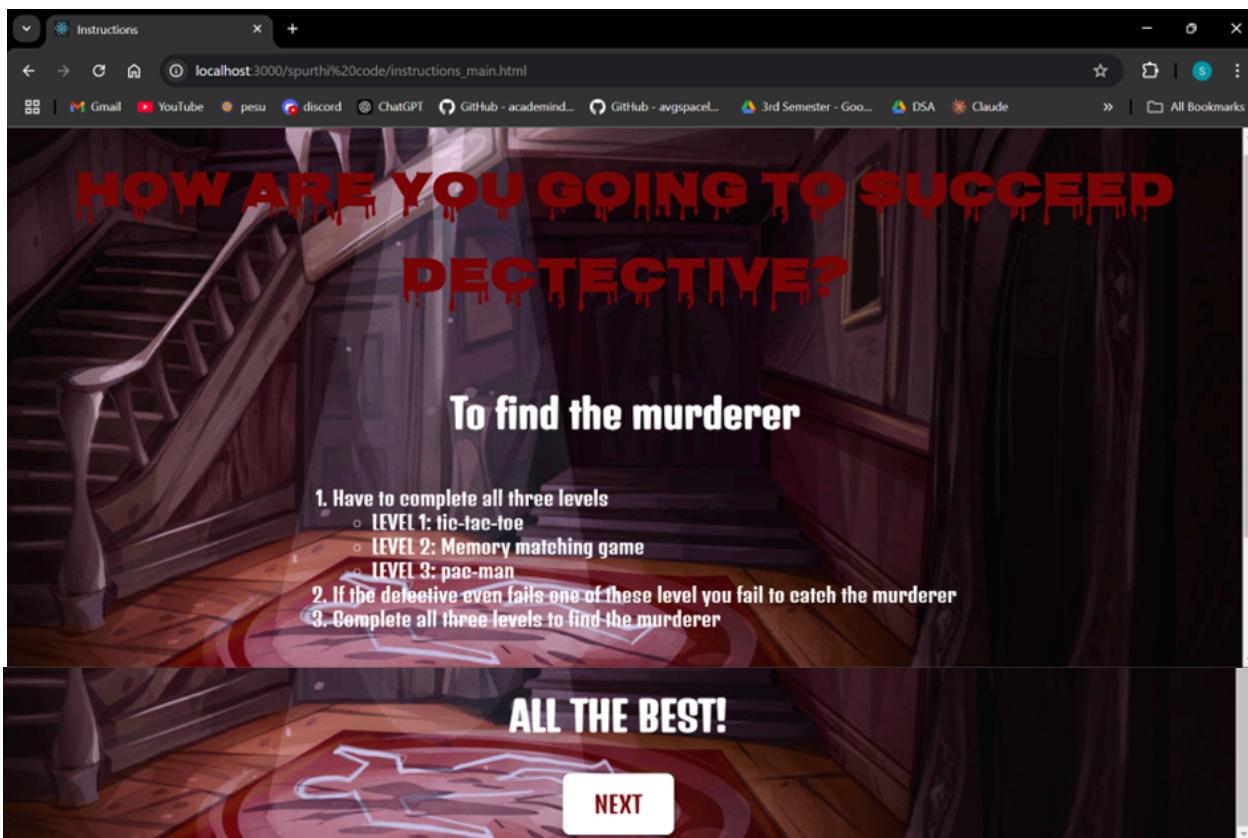




THE PUZZLE OF SHADOWS

Detective Harper is called to investigate the sudden death of Mr. David Sinclair, a wealthy businessman. Found dead in his study with no visible injuries, only a cryptic note lies on his desk: "The game has just begun." The killer has set up a series of puzzles, forcing Harper to play along to uncover the truth. Join **Detective Harper** on a thrilling adventure to crack these mind-bending puzzles! Only the sharpest minds can help Harper piece together the clues and unravel the mysteries. Are you up for the challenge? Dive in, solve the puzzles, and assist **Detective Harper** in cracking the case before it's too late!

NEXT



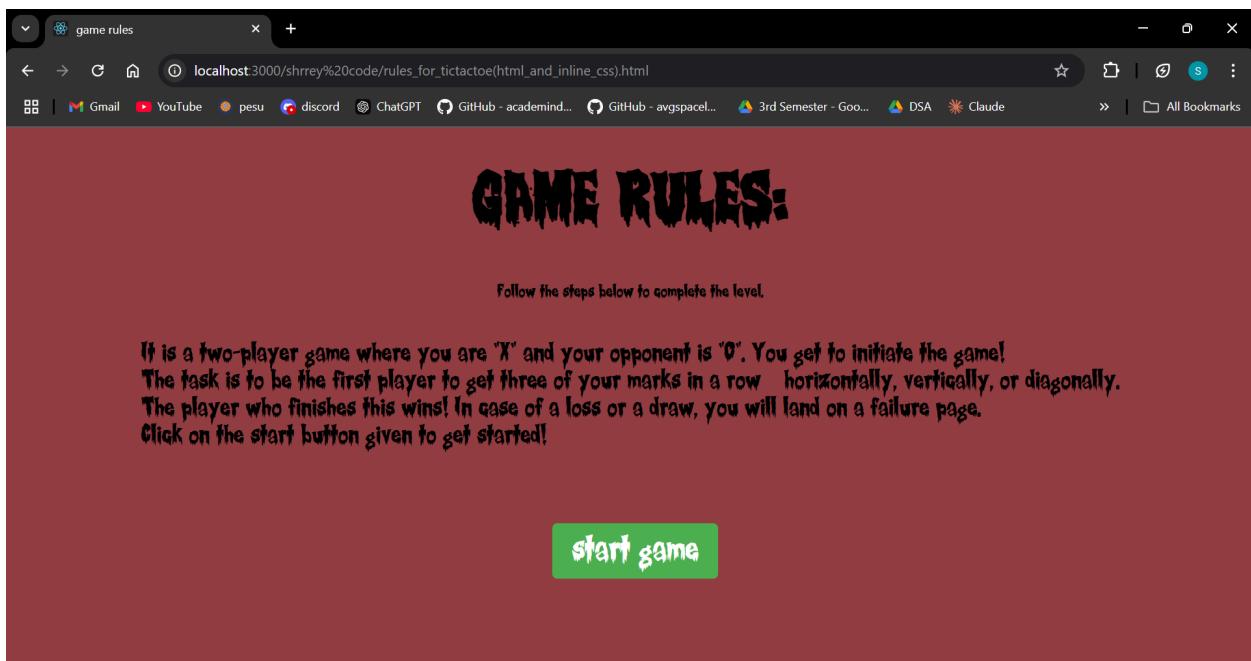
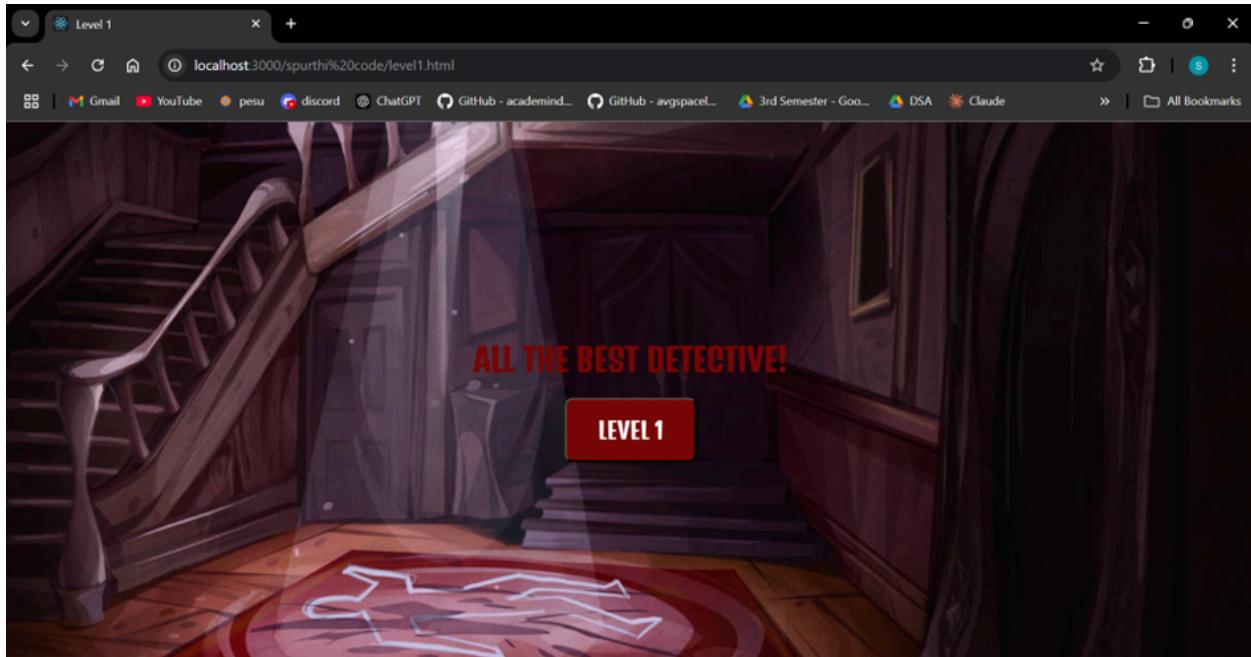
**HOW ARE YOU GOING TO SUCCEED
DETECTIVE?**

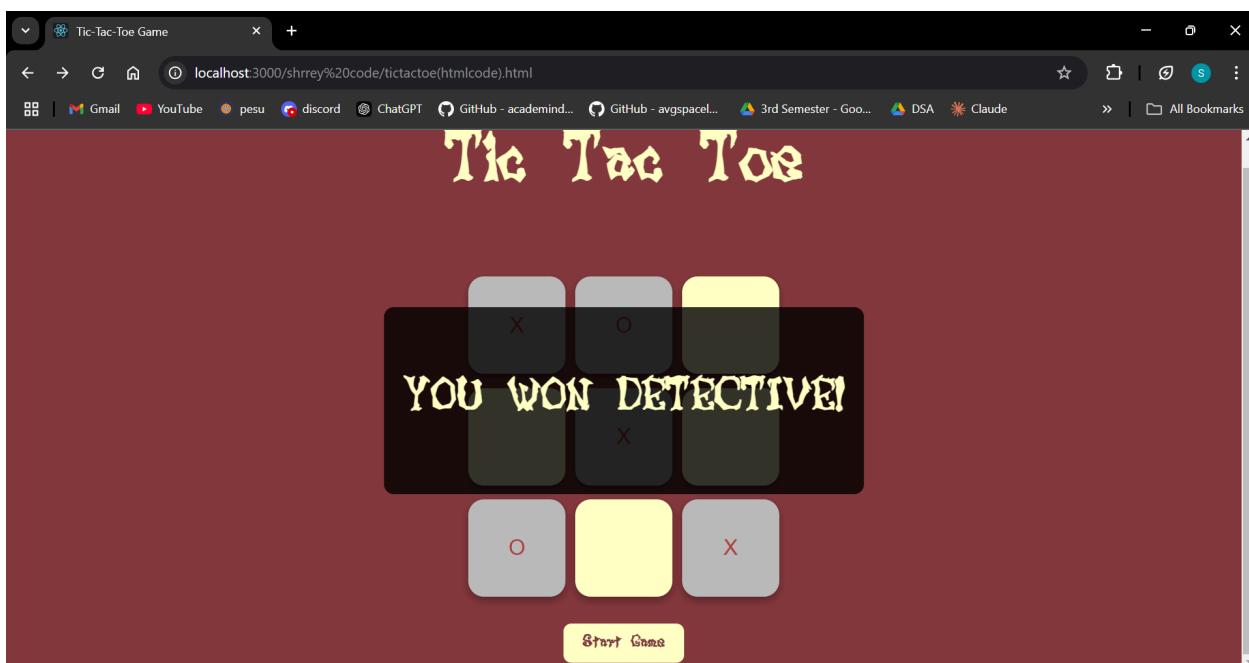
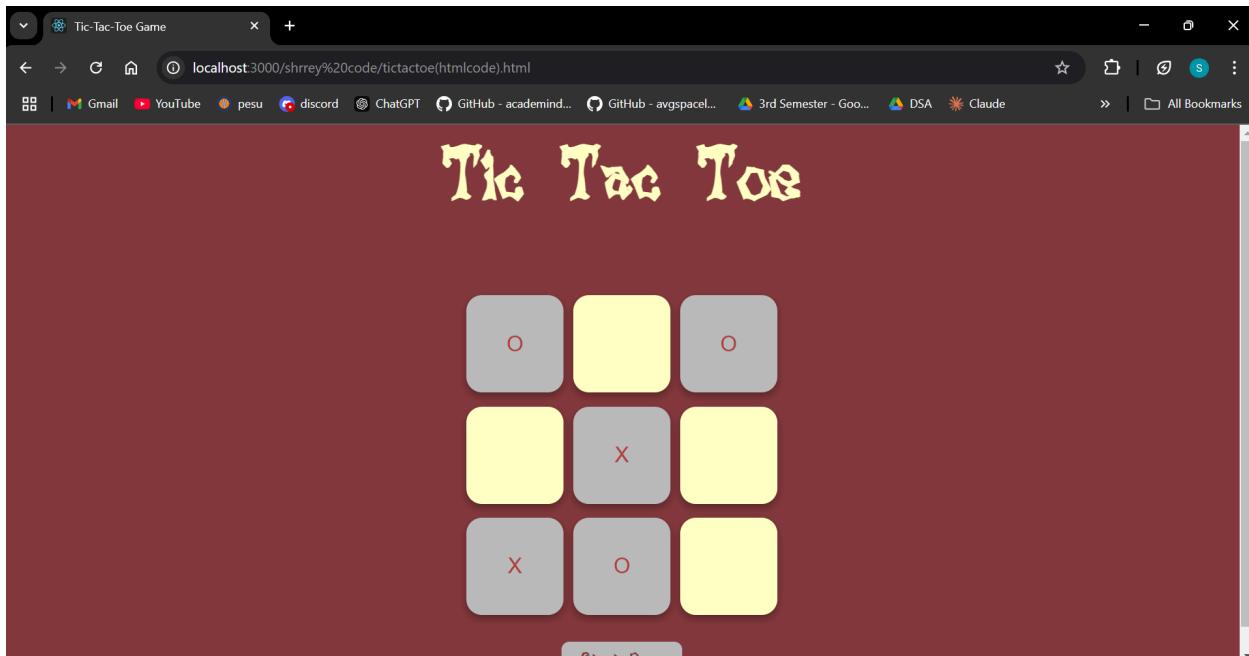
To find the murderer

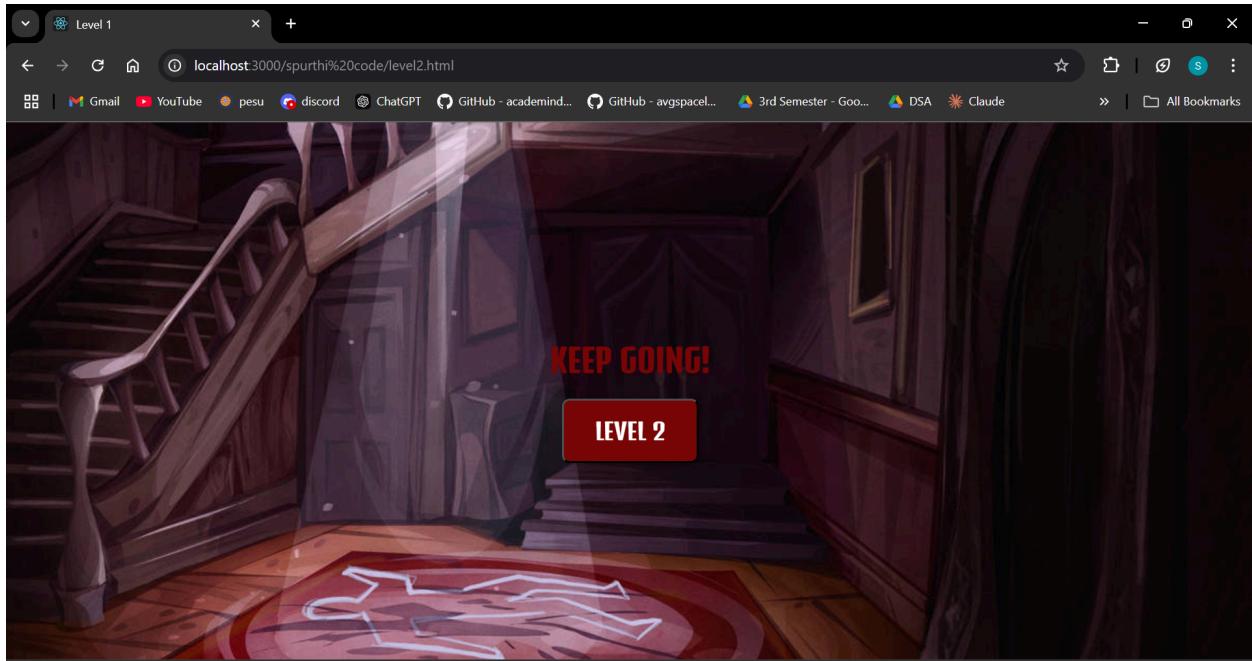
1. Have to complete all three levels
 - LEVEL 1: tic-tac-toe
 - LEVEL 2: Memory matching game
 - LEVEL 3: pac-man
2. If the detective even fails one of these level you fail to catch the murderer
3. Complete all three levels to find the murderer

ALL THE BEST!

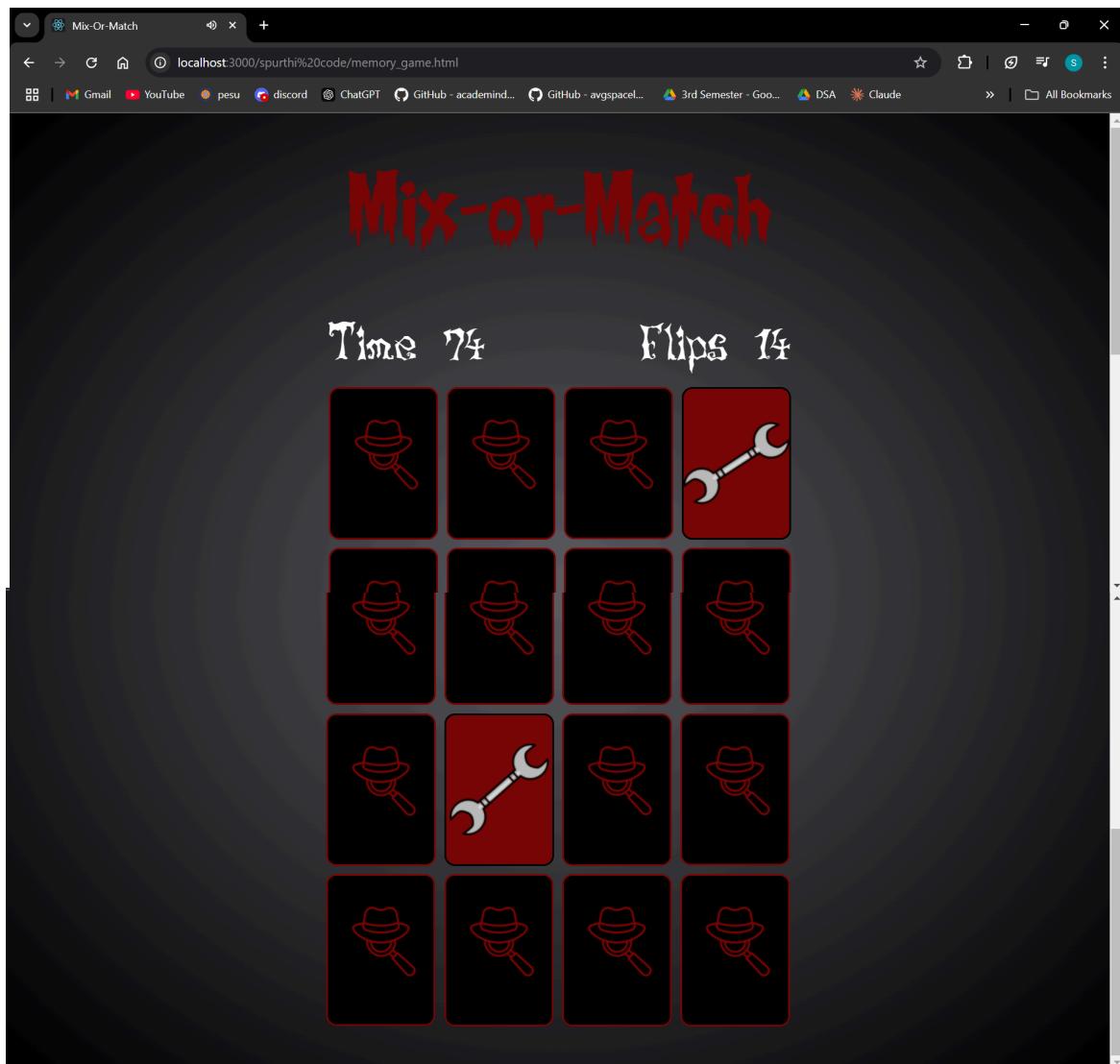
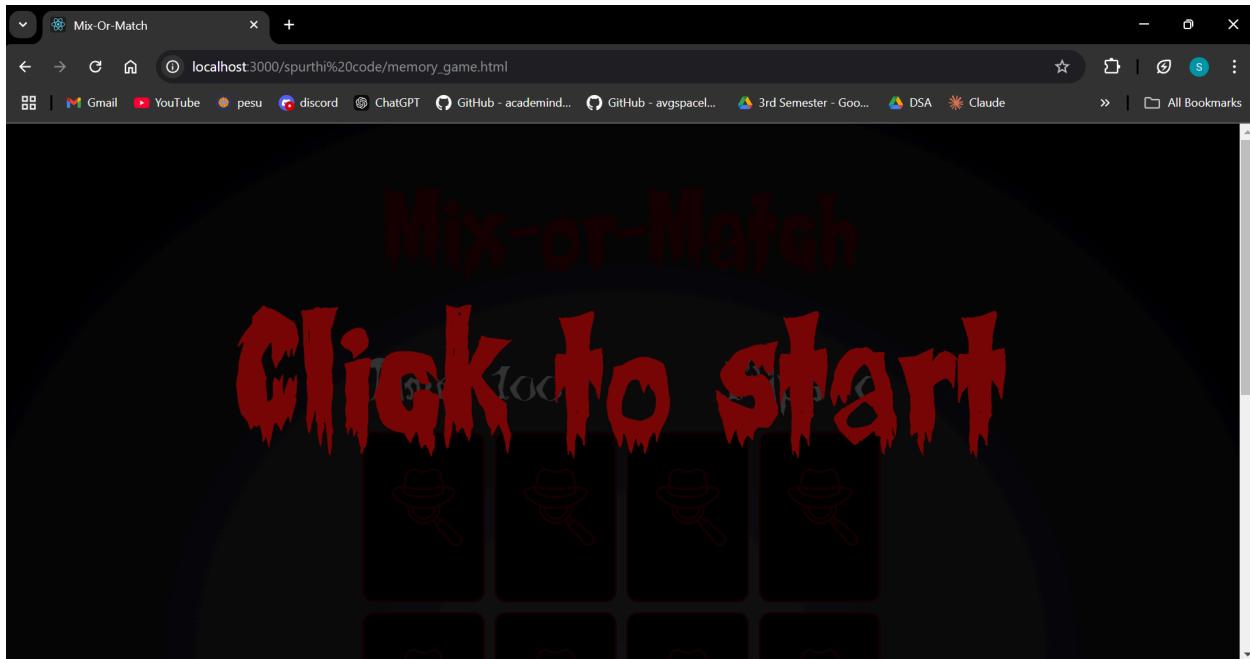
NEXT

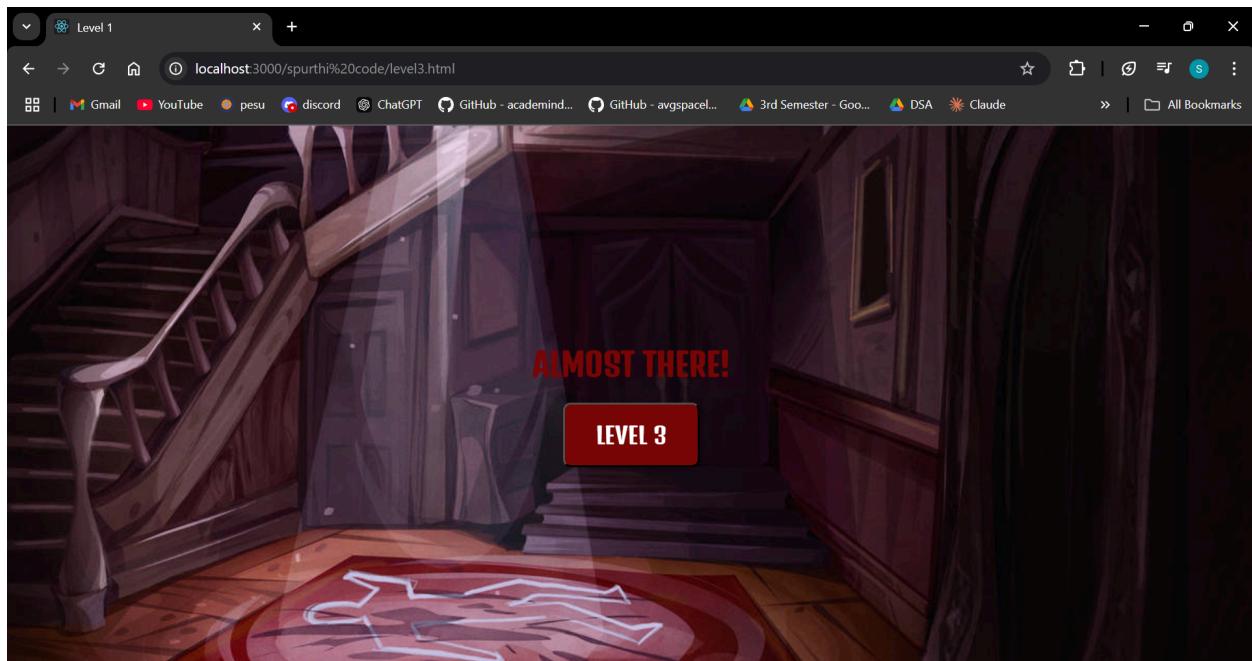
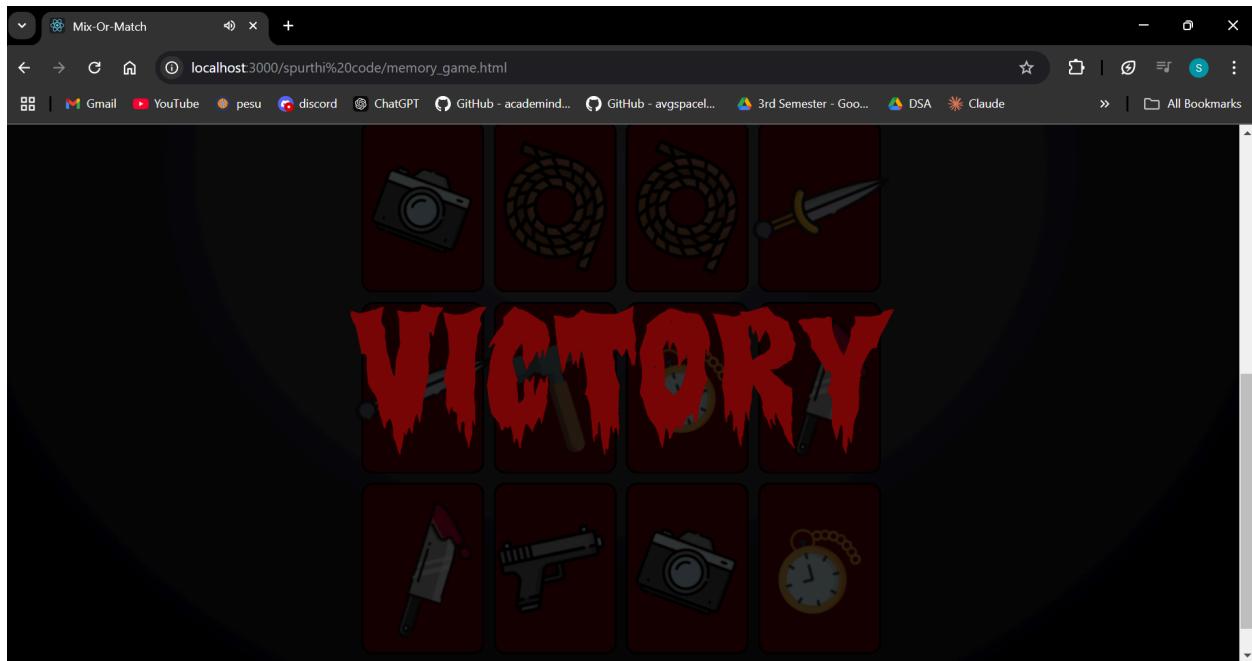


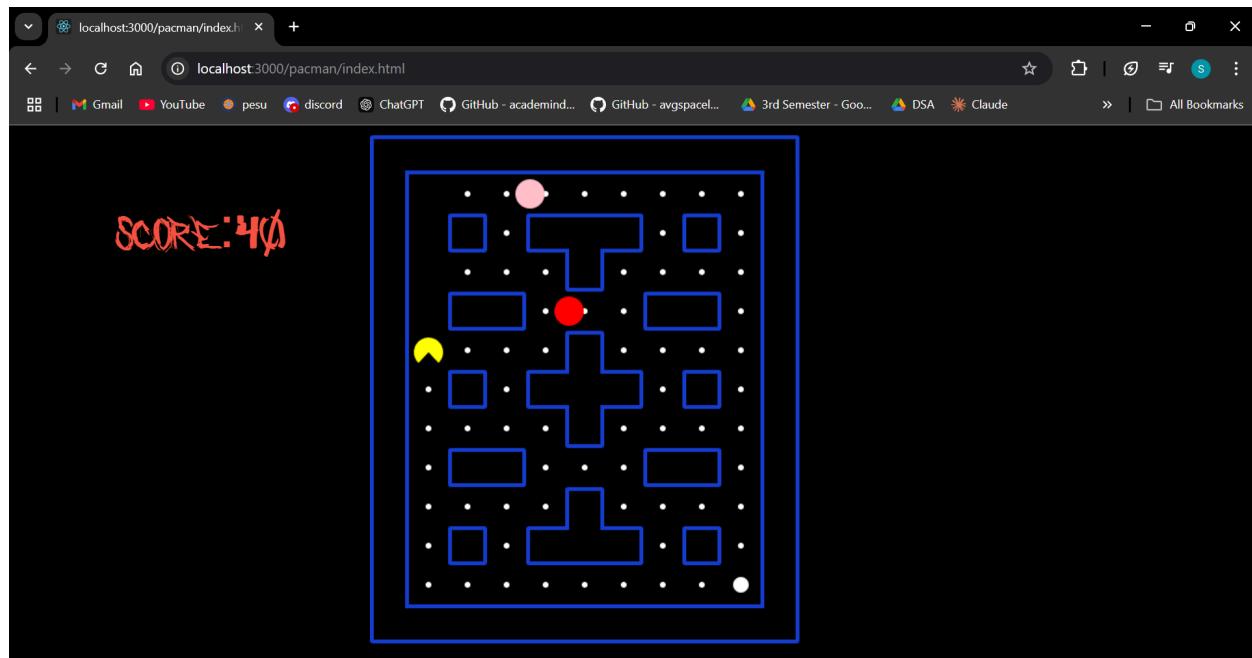
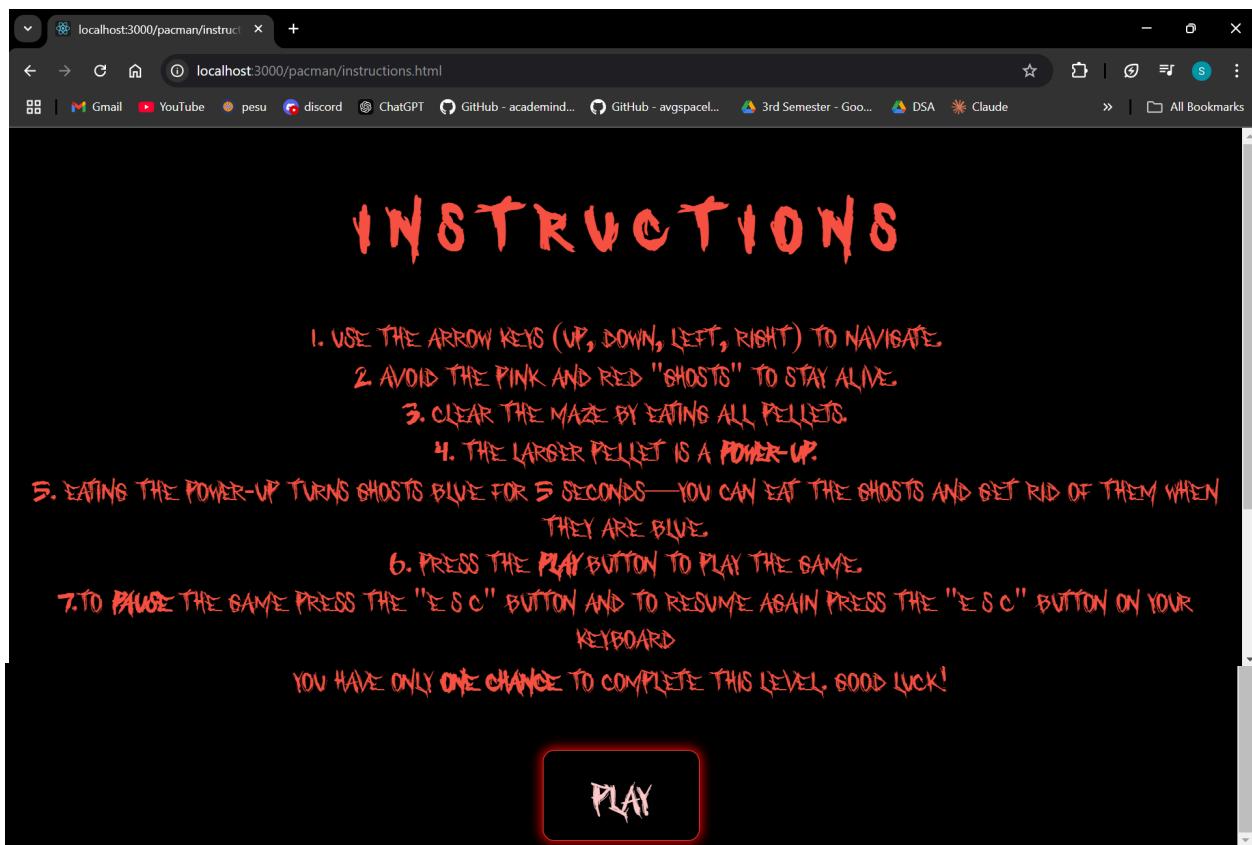


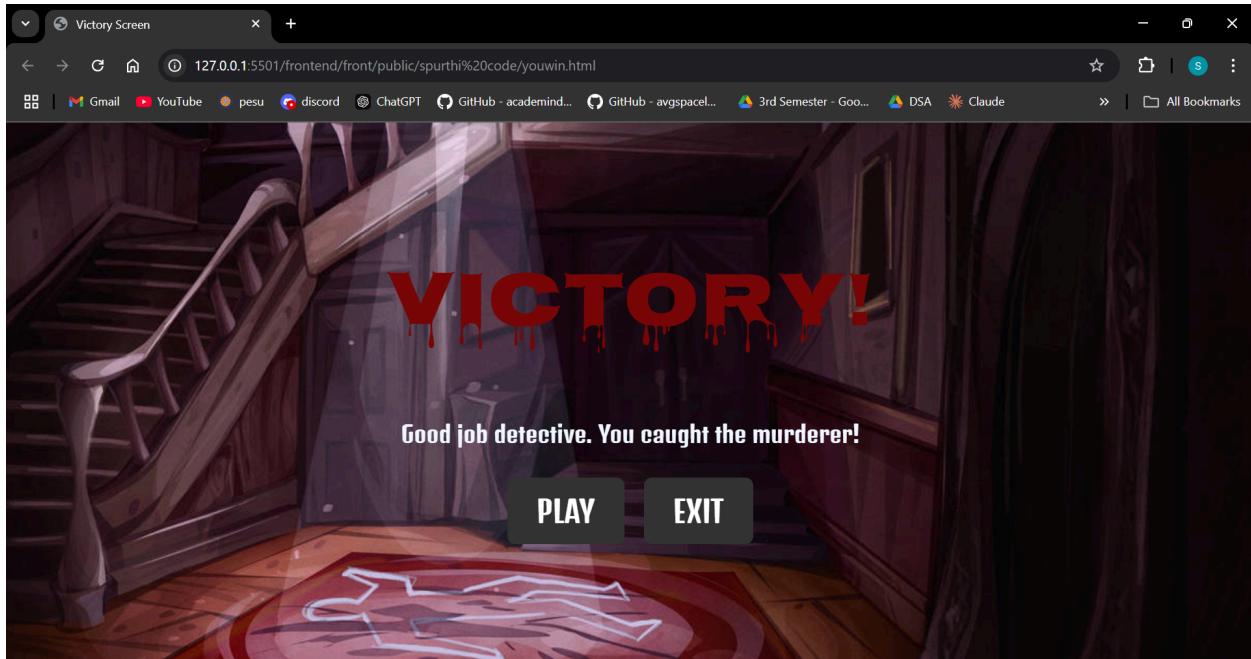


A screenshot of a web browser window showing game instructions. The title "Instructions" is displayed in a large, stylized font. Below the title, there are four numbered steps: 1. Your goal is to find and match all pairs of identical cards within the game grid. 2. The game starts with a set of cards laid face down. Each card has an identical pair hidden on the grid. 3. Click any two cards to flip them. If the cards match, they stay face-up. If not, they flip back over. 4. Match all the pairs to win. At the bottom, there is a red button with the text "CLICK TO PROCEED". The browser's address bar shows "localhost:3000/spurthi%20code/instructions.html". The top of the browser has standard navigation and search controls.

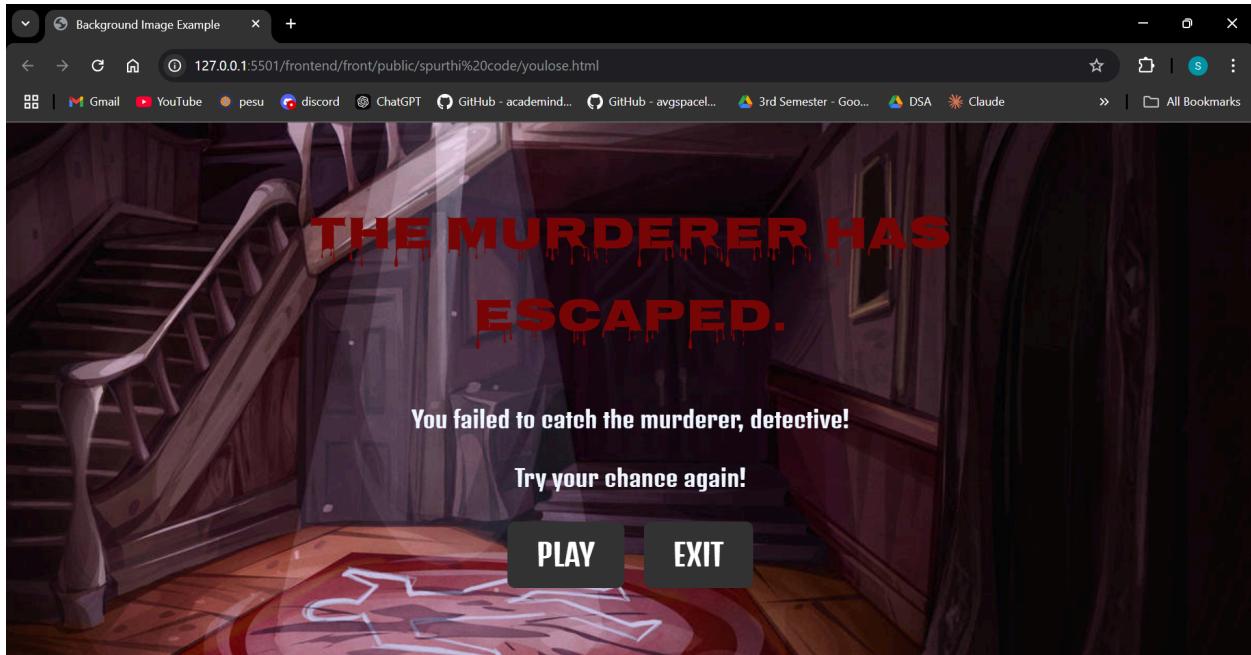


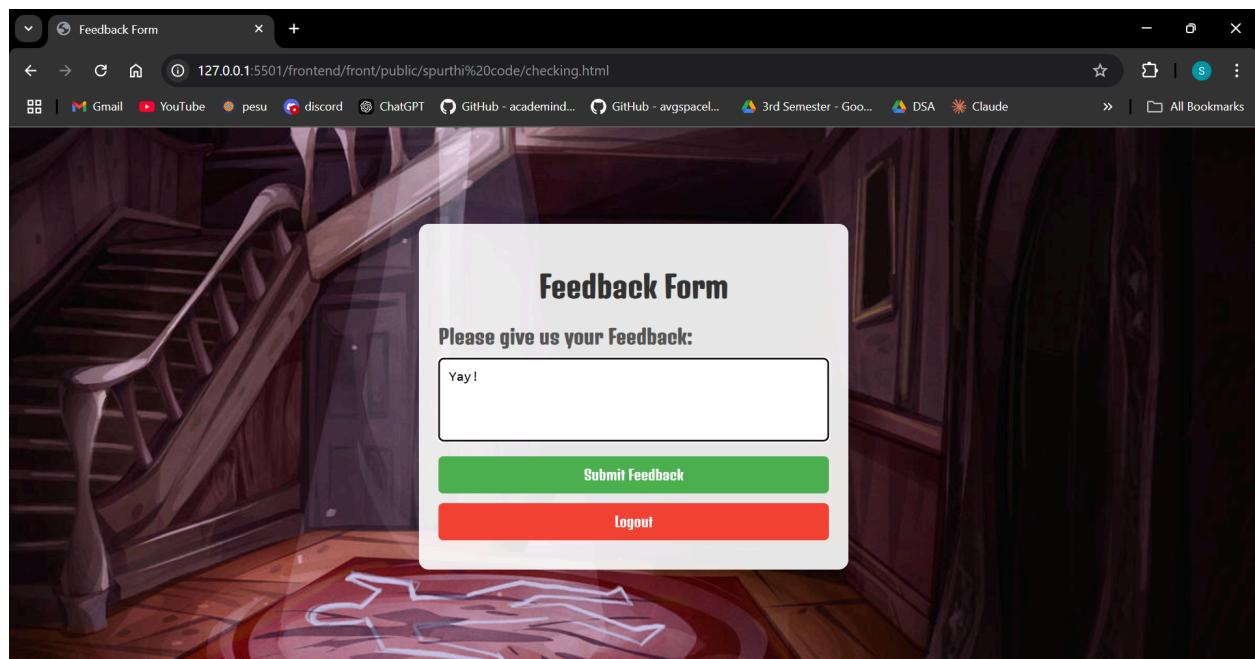
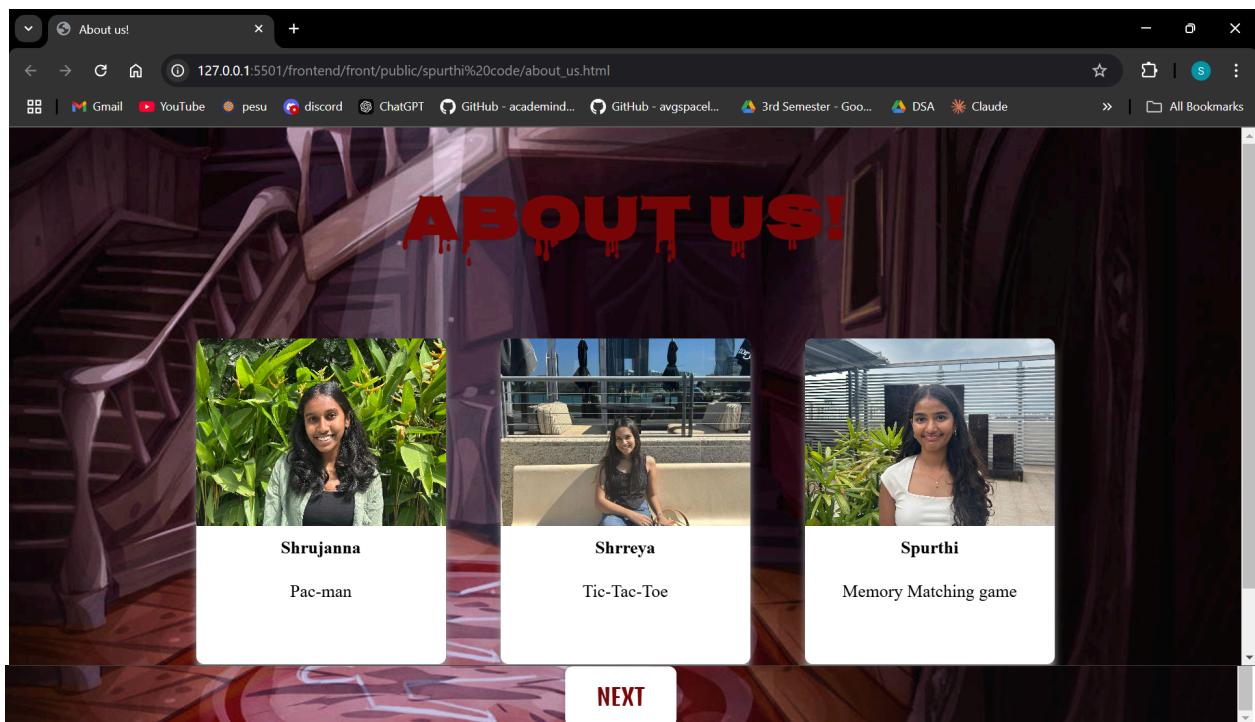


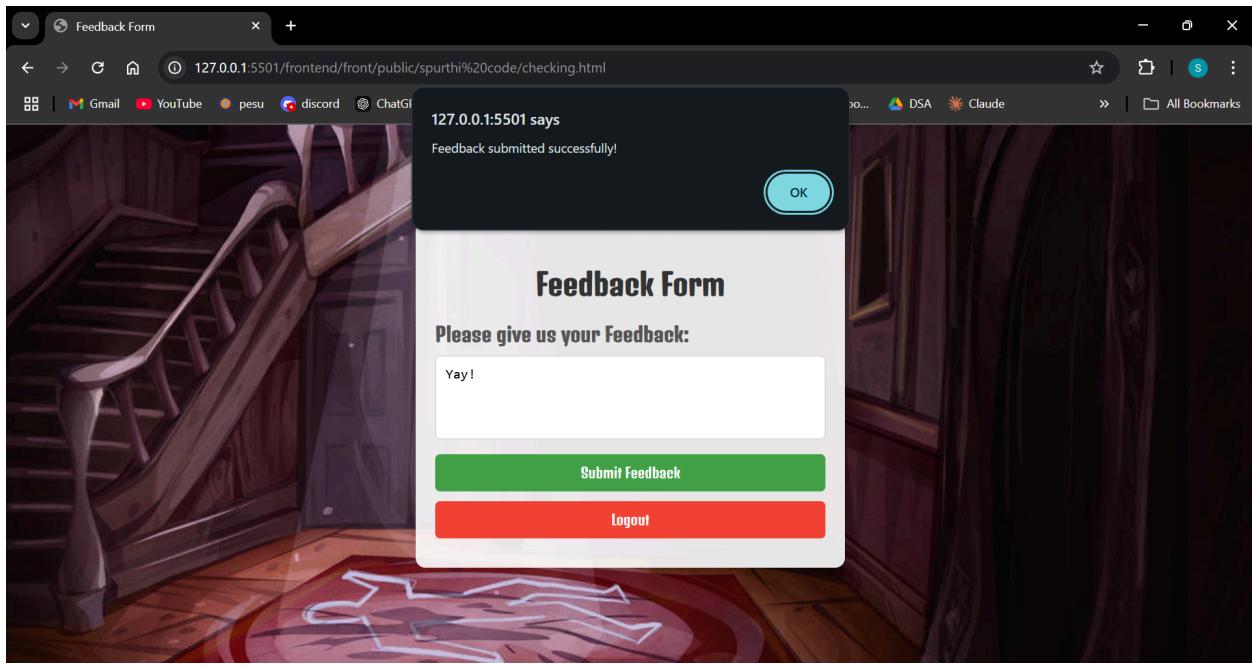




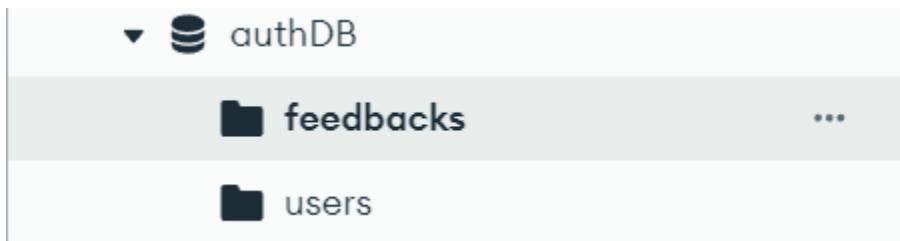
A dedicated failure page to be displayed whenever the user fails to progress past a level.







The database:



The screenshot shows the 'users' collection in the 'authDB' database. The interface displays 49 documents. Each document includes fields: '_id', 'email', 'password', and '__v'. The 'email' field shows three distinct email addresses: 'spurthi@gmail.com', 'meghana@gmail.com', and 'newlogin@gmail.com'. The 'password' field shows three distinct hashed passwords. The 'Find' button is highlighted in green.

Document ID	Email	Password	__v
_id: ObjectId('673c0bfff9f167c36979b60fb')	spurthi@gmail.com	\$2a\$12\$1Kucz00QRJjEAJBivM.H.tDavhZUR/3UJRTZRUZ6Gn3l3A0UHiq	0
_id: ObjectId('673c122b9f167c36979b6101')	meghana@gmail.com	\$2a\$12\$YQt/1yCbbkw5xFuWpj87o.80h.oKLEuh0iA6gb0Q6m2/YWDvbzX3i	0
_id: ObjectId('674c4eae851d3f36a517a687')	newlogin@gmail.com	\$2a\$12\$UmKwaewbWJRmm1m.Xbug0/Vxft0Ebhf9Rlr7IlPhzUP1ND7ValBy	0

The screenshot shows the 'feedbacks' collection in the 'authDB' database. The interface displays 24 documents. Each document includes fields: '_id', 'feedback', 'createdAt', and '__v'. The 'feedback' field shows three distinct feedback entries: 'djasdjkf', 'Good!', and 'Yay!'. The 'createdAt' field shows three distinct dates: '2024-11-19T04:00:50.667+00:00', '2024-11-19T04:25:20.168+00:00', and '2024-12-01T11:53:58.646+00:00'. The 'Find' button is highlighted in green.

Document ID	Feedback	Created At	__v
_id: ObjectId('673c0d729f167c36979b60fd')	djasdjkf	2024-11-19T04:00:50.667+00:00	0
_id: ObjectId('673c13309f167c36979b6103')	Good!	2024-11-19T04:25:20.168+00:00	0
_id: ObjectId('674c4e55851d3f36a517a684')	Yay!	2024-12-01T11:53:58.646+00:00	0