Error Handling Helpline

Group Members

- Reece Willoughby
- Alexi Colley
- Nawaraj Budhathoki
- Roman Reshetniak

Task board:

https://github.com/users/WILR96/projects/1

GitHub:

https://github.com/WILR96/ErrorHandlingHelpline

Project Description

This project is a Debugging & Programming Help platform that lets students post technical programming problems and receive help from the community.

Users can share error messages, code snippets and environment issues. Others can respond with explanations and debugging guidance.

The platforms' goal is to promote collective learning, support and help in developing critical thinking skills.

Code of Conduct

Our Pledge

We are committed to making participation in this project a respectful, inclusive, and collaborative experience for everyone involved.

Our Standards

Examples of behaviour that contributes to a positive environment include:

- Respectful and constructive communication.
- Being open to feedback and different viewpoints.
- Taking responsibility for assigned tasks.

Examples of unacceptable behaviour include:

- Harassment, disrespect, or dismissive language
- Repeated failure to contribute without communication.
- Removing or altering others' work without discussion.
- Academic misconduct or reuse of prior assessed work.

Responsibilities

All contributors are expected to:

- Follow the agreed development workflow.
- Meet deadlines or communicate early if issues arise.
- Act in line with university academic integrity rules.

Enforcement

Breaches of this Code of Conduct should be raised with the group.

By contributing to this project, you agree to follow this Code of Conduct.

Persona 1

Alex Johnson – Junior Developer

Demographics

Age	21
Role	Second-year Computer Science Student
Experience Level	Beginner to Intermediate (18 months of programming)
Primary Languages	Python, Java, JavaScript (React)
Development Environment	VS Code, Git

Background

Alex is a motivated Computer Science student balancing academic coursework with personal coding projects. While enthusiastic about programming, Alex frequently encounters debugging challenges that slow progress and cause frustration. Limited industry exposure means Alex often struggles to interpret error messages or identify root causes efficiently.

Goals	**Needs**	Pain Points & Frustrations
Resolve coding errors quickly and effectively	Clear explanations of what errors mean in plain language	Error messages feel cryptic and overwhelming (e.g., 'NullPointerException' without context)
Develop independent debugging skills	Step-by-step debugging guidance tailored to the specific issue	Stack Overflow solutions often assume advanced knowledge
Understand error messages beyond surface-level fixes	Code examples demonstrating correct implementations	Public forums can be intimidating or dismissive of 'basic' questions
Build confidence in problem-solving abilities	Educational context explaining why errors occur	Uncertainty about whether solutions are secure or best practice
Meet assignment deadlines without excessive stress	Non-judgmental support that encourages learning	Time wasted cycling through trial-and-error approaches

Technical Profile

Strengths	Weaknesses:
Understands fundamental programming concepts	Limited exposure to advanced debugging tools
Capable of writing functional code for simple applications	Struggles with asynchronous programming and state management
Familiar with basic version control (Git)	Difficulty reading stack traces or console logs effectively

Scenario

Alex encounters a "TypeError: Cannot read property 'map' of undefined" while building a React component for coursework.

Trigger:

Error appears at 10:30 PM, two days before submission deadline.

Action:

Submits error message, relevant code snippet, and brief description of what they were.

attempting

Interaction:

Reviews the helpline's automated analysis and suggested solutions.

Resolution:

Receives guidance from Sarah explaining async data loading, with a code example showing conditional rendering.

Outcome:

Fixes the issue, understands the underlying problem, and saves the solution for future reference.

Persona 2

Sarah Patel – Senior Software Engineer

Demographics

Age	35
Role	Senior Software Engineer & Technical Mentor
Experience Level	Advanced (12+ years in industry)
Specializations	Full-stack development, debugging, system architecture
Industries:	Fintech, cloud services, e-commerce

Background

Sarah is an experienced software engineer who values knowledge-sharing and community

support. Having mentored numerous junior developers throughout her career, she understands common learning gaps and enjoys helping others develop problem-solving skills. She dedicates 5-10 hours weekly to mentoring activities, viewing it as both professional development and a way to give back to the community.

Goals	**Needs**	Pain Points & Frustrations
Guide junior developers toward independent problem-solving	Structured format for reviewing submissions efficiently	Repeatedly encountering the same basic errors without proper documentation
Share industry best practices and debugging methodologies	Sufficient context (code snippets, error logs, attempted solutions)	Submissions lacking context (e.g., no error message, incomplete code)
Reduce repetitive errors through educational explanations	Annotation tools to highlight specific issues and suggest improvements	Students expecting immediate answers rather than guidance
Foster a supportive, inclusive developer community	Filtering options to prioritise questions matching her expertise	Limited time to provide detailed explanations for every request
Maintain a reputation as an approachable, effective mentor	Time-efficient workflow given limited availability	Difficulty tracking which students she's previously helped

Technical Profile

Expertise	Teaching Philosophy
Proficient in Python, JavaScript, Java, TypeScript, Go	Believes in Socratic questioning to guide discovery
Deep knowledge of debugging tools (Chrome DevTools, debuggers, profilers)	Emphasizes understanding 'why' over quick fixes
Experience with error monitoring systems (Sentry, New Relic)	Encourages best practices from the start
Strong understanding of common anti-patterns and code smells	

User Journey with the Helpline

Scenario

Sarah reviews Alex's React error submission during her morning coffee break.

Discovery

Logs into the helpline and filters for JavaScript/React questions.

Evaluation

Reviews Alex's error message, code snippet, and description.

Diagnosis

Identifies the issue as an async data handling problem (common pattern).

Response

• Asks diagnostic questions to confirm understanding.

• Explains the root cause in accessible language.

• Shares a corrected code example with inline comments.

• Links to React documentation on conditional rendering.

• Adds a best-practice tip about loading states.

Follow-up

Checks back later to see if Alex found the solution helpful.

How They Interact

The Error Handling Helpline facilitates meaningful knowledge exchange between help seekers and experienced developers.

Typical Exchange

11:47 PM – Alex submits:

"Getting 'map is not a function' error in my React app. Tried adding .map() but still broken. Code attached. Help?"

8:15 AM (next day) – Sarah responds:

"Hi Alex! This is a common async issue. Can you check what console.log(data) shows when the component first renders? I suspect it's undefined initially. Here's why..."

9:30 AM – Alex replies:

"Oh! It was undefined at first. Your conditional rendering example fixed it. Thank you!"

9:45 AM – Sarah:

"Excellent! This pattern will help with many similar situations. Bookmark this for future reference."

Ethical issues

User Data & Privacy

Users can submit help requests via the ticketing system by specifying a subject and describing their problem in detail. These submissions may include personal information or critical technical details. An ethical challenge emerges when ensuring that user data is managed appropriately and not captured or retained beyond its intended purpose. The system should capture only the data necessary for delivering help and ensure that submitted information is only accessible to authorised support team members.

Confidentiality & Responsible Data Handling

When requesting assistance using the ticketing system, users may provide personal projects or confidential information. There is an ethical need to protect the confidentiality of these submissions and prevent unauthorised access or disclosure. Support team must exercise discretion when handling all issues and refrain from disclosing user content outside of the intended help context. This also includes

ensuring that submitted data is handled securely within the system.

Transparency & User Trust

Users should be aware of how the ticketing system operates, including what information is required, how tickets are processed, and who may read their submissions. A lack of transparency might lead to misunderstandings or mistrust. To enable informed user engagement, the system should properly express its purpose and restrictions.

Accessibility & Inclusivity

There is an ethical responsibility to ensure that the ticketing system is accessible to people of various abilities and technical backgrounds. Clear form labels, understandable text, and simple navigation help ensure that all users can submit tickets without encountering needless obstacles. Failure to address accessibility may preclude specific user groups from receiving assistance.

Impact & Misuse

Poor management of user input may expose the system to abuse, such as spam submissions or malicious content. There is an ethical concern in ensuring that the website handles user input properly to avoid potential harm to both users and the system. Measures like input validation and responsible access control help to decrease the risks of misuse or data leakage.

Date and Time	3rd February 2026 – 5:00
Project Name	Error Handling Helpline
Meeting Goal	Agree group name Choose project Assign Tasks
Facilitator	Reece Willoughby
Note taker	Reece Willoughby
Attendees	• Reece Willoughby • Alexi Colley • Nawaraj Budhathoki

Roundtable Updates (each group member to contribute)	Reece proposed several possible project ideas and outlined potential scopes. All group members discussed feasibility and relevance to the module requirements. General agreement that the project should be educationally focused and technically realistic within the timeframe given.
Discussion points	We discussed the ideas we had as a group: **Community Repair Platform:** Users post broken items and others volunteer repair help. **Peer Programming Help:** Students help each other with programming problems. **Programming Challenge Platform:** Users create and review small coding challenges. **Reverse-Engineering & Security Learning:** Educational focus on binary analysis and reverse engineering concepts. **Debugging & Programming Help:** Users post specific technical issues and receive debugging assistance. After discussion, the group agreed that “Debugging & Programming Help” best fit our collective technical skill level whilst also hitting the assessment criteria.
Actions (list tasks and assign a group member)	At least one commit from every group member - All Each member of the team able to run the development environment - All At least two ‘Personas’ - Nawaraj Ethical issues identified - Alexi Kanban Board and Github project - Reece Scaffolding files added to Github and README file customised for your project - Reece Project description refined - Reece Code of Conduct - Reece Meeting records completed - Reece