CHEMSAFE AI: DESIGN AND DEVELOPMENT OF AN INTELLIGENT MOBILE APPLICATION FOR LABORATORY SAFETY EDUCATION AND HAZARD SUPPORT

Wentao Zhou[1], Cesar Magana[2]

[1]Portola High School, 1001 Cadence, Irvine, CA 92618
[2]California State University Long Beach, 1250 Bellflower Blvd, Long Beach, CA 90840

ABSTRACT

With a large portion of educators and students having experienced laboratory accidents, there's been a clear shortage of laboratory safety education, leaving young chemists defenseless against hazardous situations [1]. To address this, I developed ChemSafe AI, a mobile application combining digital safety education with real-time hazard support[2]. The app features 3 key systems: A Lab Planning Assistant for reviewing experiments, an Emergency Assistant for immediate emergency guidance, and a Camera System for visual hazard recognition. Development challenges arose when differentiating visually similar compounds with images proved to be impractical, forcing a redesign of the camera system to generate accurate safety protocols by instead analyzing contextual clues in photos. This change proved to be successful, as experimental testing revealed that the system had high detection and classification accuracy across various images. Ultimately, through guidance and education, ChemSafe AI hopes to become a reliable digital mentor, used to reduce lab accidents while restoring confidence within young aspiring chemists to safely pursue individual experimentation [3].

KEYWORDS

Laboratory Safety, Artificial Intelligence, Hazard Recognition, Mobile Application

1. INTRODUCTION

The presence of hazardous chemicals and equipment has prolonged chemistry laboratories as a place contaminated with risks and dangers. Recent statistics from the Occupational Safety and Health Administration (OSHA) have found that academic laboratories are 11 times more dangerous than industrial settings [4]. Furthermore, research also reveals that 46% of K-12 science teachers have reportedly received no formal safety training, with 40% having experienced or witnessed a laboratory accident. These staggering numbers all stem from the central issue that students and educators from primary and secondary schools receive inadequate access to lab-safety training, leaving them unprepared and vulnerable during emergencies. To add on, not only does this issue leave children susceptible to physical injuries, it also destroys student confidence by portraying chemistry as an exceedingly dangerous field. Studies reveal that over 70% of students view inconsistent safety as a deterrent, showcasing how lab-safety fears can discourage aspiring chemists from participating in laboratories or pursuing individual experiments.

David C. Wyld et al. (Eds): BMLI, CACIT, EDUAIT, InWES – 2026
pp. 69-80, 2026. CS & IT - CSCP 2026 DOI: 10.5121/csit.2026.161806

Solving this issue is critical in order to protect students and educators from unnecessary harm while rebuilding confidence to encourage greater participation and progress in the field of chemistry.

Section 5 evaluated three primary methodologies: AI Educational Robots (AIERs), Virtual Reality (VR), and Educational Android Apps (LabSafety) [5][6]. These approaches all seek to offer modernized lab-safety education, replacing traditional lectures with more engaging digital platforms to improve users' safety skills. However, each of these solutions came with significant limitations. AIERs and VR systems are expensive and difficult to install, making them inaccessible to average users. Existing mobile apps, such as LabSafety, offered more widespread accessibility but only offered passive educational content, incapable of giving real-time assistance to users when they're faced with actual laboratory emergencies. My app, ChemSafe AI, solves many of these existing shortcomings by delivering a free, highly accessible mobile solution that combines passive educational resources with AI-powered emergency assistants. This way, users can receive real-time support and guidance from the app's assistants during experiments while still receiving safety tips to increase general safety knowledge outside the laboratory.

ChemSafe AI is an app that functions as an AI-powered lab supervisor, offering educational content and safety advice through chat interactions [7]. The primary concept was inspired by a personal realization: even with preventive lab-safety preparation, laboratories can create unpredictable situations, often making students frazzled and confused during real experiments. In order to solve this, ChemSafe AI has combined passive learning with real-time assistance to benefit the users' safety inside and outside the lab.

The app addresses the issue of inadequate lab-safety resources by including preventive lab-safety content such as safety tips and GHS hazard symbols to educate users outside the lab [8]. During experiments, emergency AI assistants let users photograph or type prompts explaining their concerns to receive immediate emergency instructions.

While traditional lab-safety training relies on passive teaching in the form of lectures and quizzes, ChemSafe AI uses captivating interface designs and personalized guidance to offer a visually pleasing and personalized learning experience for each user. This approach seems more unique and effective compared to conventional solutions, as it supports users in all stages of an experiment, from the planning stage all the way to the experiment stage. By offering assistance both during and outside of experiments, the application reassures many chemists by providing a navigation system when emergencies occur.

An identification test was performed to determine the general accuracy and reliability of my app's camera classification system. The experiment was created by using a dataset of 20 diverse images, designed to evaluate how the system would perform when fed a variety of photos, including images that were blurry, non-hazardous, and hazardous. Findings from the experiment revealed that while the system excelled at detecting visible hazards, even under harsh lighting and blurriness, it occasionally struggled to recognize the absence of subtle elements, such as missing safety equipment. This weakness occurred due to AI models generally being trained to identify what is present in a picture, rather than trying to infer what is missing or should be in a picture. Furthermore, since the camera system can't analyze data in the form of videos, hazards stemming from inaccurate physical technique, such as improper pipetting motion, would sometimes go unaddressed. Despite the shortcomings, the experiment was still a large success, as it confirmed the overall high accuracy of the system, making me confident in the camera's ability to point out dangerous risks and hazards to users.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Preventing Camera Abuse

The app's camera system includes a detection feature that pauses processing whenever an image is perceived to be non-laboratory-related, helping prevent users from repeatedly uploading irrelevant photos to increase API usage and rack up expensive processing costs. While this system cannot prevent all forms of spam or completely eliminate the possibility of users abusing the camera feature, it minimizes the amount of processing that unrelated photos incur, making it more difficult and discouraging for users to use the software in an unethical way.

2.2. Advantages of the Lab Page

Oftentimes, unless specifically requested by the user, search engines and standard AI chatbots like Google and ChatGPT won't include certain safety details, such as required PPE or equipment [9]. The Lab Page system is programmed to deliver outputs following a specific format that always includes critical details, such as safety tips and the necessary PPE and equipment needed for the proposed experiment. Furthermore, the system is designed to provide the best safety advice, even if it means discouraging the user from performing experiments that seem too dangerous, making it less vulnerable to AI sycophancy compared to general-purpose search engines and chatbots.

2.3. Reducing Hallucinations

While the Check Safety system cannot completely eliminate hallucinations, it does reduce the possibility of error through restriction and categorization. Before an output is generated, the system first categorizes incidents based on the four most common lab hazards—Burns, Inhalation, Spills, and Flames—making it more difficult for the AI to produce an off-topic response that is unrelated to lab safety.

3. SOLUTION

Start:After opening the app, the user is loaded into the homepage, where a navigation bar appears at the bottom of the screen. This bar acts as a primary interface, allowing users to navigate between the app's four core sections: Homepage, Safety Page, Camera Page, and Profile Page.

The interactions within each section are as follows:

Homepage: This is the app's home screen, displaying a daily safety tip at the top. Underneath, the user can tap and explore on four different boxes, leading to four different sub-pages (Common Elements, Safety Labels, Safety Equipment, and Safety Page) that detail information about common elements, GHS symbols, and lab tools. The bottom of the homepage displays a box titled Plan a Lab Session. Tapping the box transports the user to the Lab Page where they can input experiment details and receive safety feedback.
Safety Page: This page contains common accident symbols and direct-dial options for emergency health services such as 911 and Poison Control. Users can also interact with the Emergency Assistant, an AI-powered chatbot that allows the user to receive action steps after entering details about an accident or potential concerns.

Camera Page: A visual classification page where users can take photos or select images from their photo library. These images are then sent to classifiers, which evaluate the visual data and then display classification results such as hazardous chemicals, workplace concerns, or missing equipment that pose potential danger to the user.

Profile Page/Login Page: The page begins by verifying whether the user has previously logged into an account. If logged in, the user is sent to the profile page, where they can change their username, toggle dark mode, or request an account deletion. If the login verification returns as false, the user is rerouted to the Login Page where they can sign up for an account by entering a valid email and password.

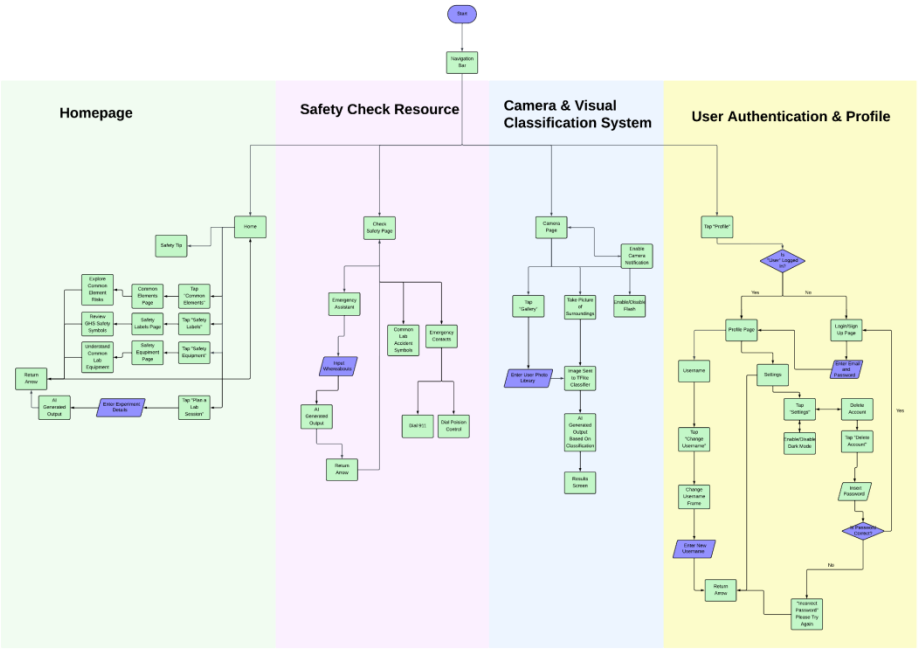


Figure 1. Overview of the solution

Plan a Lab:The first system is a Lab Planning Page where users input ideas for experiments and receive detailed advice such as necessary PPE, common mistakes, and risk scores based on toxicity, flammability, and reactivity. This system relies on the Gemini API to generate and modify outputs to match a specific design format, while utilizing Large Language Models (LLMs) to comprehend user inputs and generate safety evaluations [10].

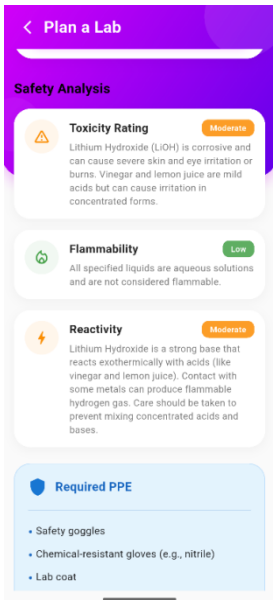


Figure 2. Lab Planning Page interface displaying risk assessment cards for toxicity, flammability, and reactivity

```
Widget _buildResultsSection(bool isDark, Color cardColor, Color textColor) {
  return Column(
    children: [
      _buildRatingCard(
        title: "Toxicity Rating",
        level: _analysisResult!['toxicity']['level'],
        description: _analysisResult!['toxicity']['description'],
        color: _getColor(_analysisResult!['toxicity']['level']),
        icon: Icons.warning_amber_rounded,
        isDark: isDark,
      ),
      _buildRatingCard(
        title: "Flammability",
        level: _analysisResult!['flammability']['level'],
        description: _analysisResult!['flammability']['description'],
        color: _getColor(_analysisResult!['flammability']['level']),
        icon: Icons.local_fire_department_outlined,
        isDark: isDark,
      ),
      _buildRatingCard(
        title: "Reactivity",
        level: _analysisResult!['reactivity']['level'],
        description: _analysisResult!['reactivity']['description'],
        color: _getColor(_analysisResult!['reactivity']['level']),
        icon: Icons.bolt,
        isDark: isDark,
      ),
```

Figure 3. Implementation of the _buildResultsSection method for rendering risk assessment cards

Beginning with the _buildResultsSection, this method acts as a bridge that takes raw data from Gemini AI analysis, then renders it to match the app's user interface. The method then checks _analysisResult to determine the assigned risk rating in order to create three vertically stacked cards for toxicity, flammability, and reactivity. As the code progresses through each of the three cards, it updates each card's visual appearance by using the risk ratings to determine background color (e.g., green for low-risk rating, red for high-risk rating). Finally, the method then adds AI-generated safety text onto each card, creating a final output that's visually attractive and informative. Using this process, the system is able to convert raw JSON data from the Gemini API into legible, organized cards that use color and text to reflect the appropriate risk ratings for toxicity, flammability, and reactivity.

Another primary system is the Emergency Assistant, which provides interactive chat assistance for questions concerning lab concerns or accidents. The system uses Gemini API backend to process inputs, then relies on Natural Language Processing (NLP) and Large Language Models (LLMs) to accurately classify a user's text into matching categories based on the four most common lab hazards (Burns, Inhalation, Spills, and Fire) [11].

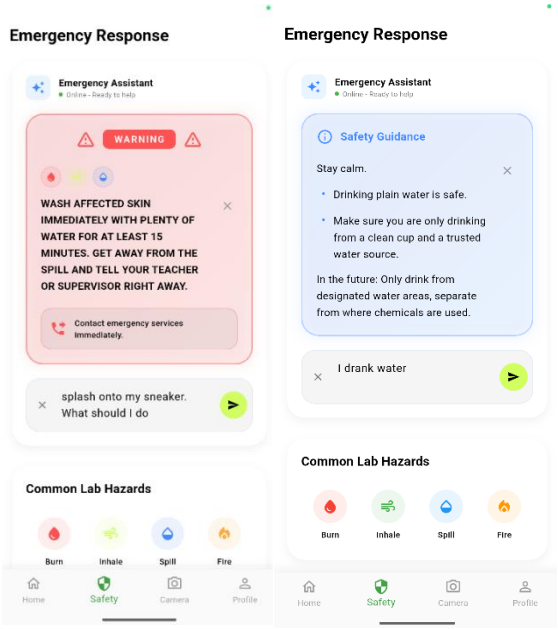
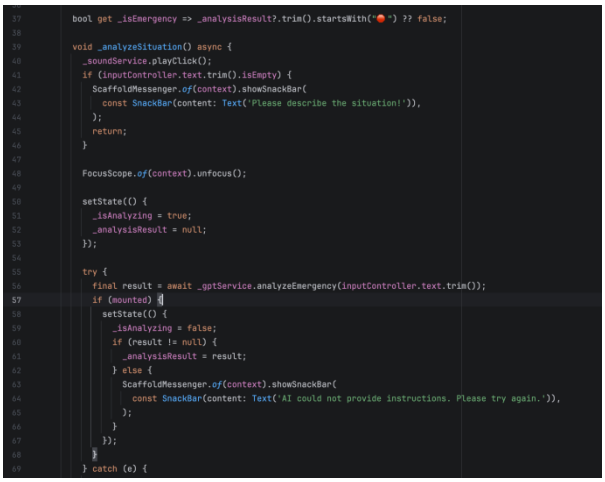


Figure 4. Emergency Assistant chat interface showing user input and AI-generated response screens



```
bool get _isEmergency => _analysisResult?.trim().startsWith("🔴") ?? false;

void _analyzeSituation() async {
  _soundService.playClick();
  if (inputController.text.trim().isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('Please describe the situation!')),
    );
    return;
  }

  FocusScope.of(context).unfocus();

  setState(() {
    _isAnalyzing = true;
    _analysisResult = null;
  });

  try {
    final result = await _gptService.analyzeEmergency(inputController.text.trim());
    if (mounted) {
      setState(() {
        _isAnalyzing = false;
        if (result != null) {
          _analysisResult = result;
        } else {
          ScaffoldMessenger.of(context).showSnackBar(
            const SnackBar(content: Text('AI could not provide instructions. Please try again.')),
          );
        }
      });
    }
  } catch (e) {
```

Figure 5. Implementation of the emergency detection logic (_isEmergency) and UI state handling
Emergency Check: The code starts with an emergency check using the _isEmergencyboolean getter to check whether the AI analysis includes an emergency indicator (a red emoji). If the indicator is detected, the system recognizes that the user has described an emergency situation, and the output's UI is changed to a red WARNING screen, containing urgent instructions and direct emergency service links. If the presence of an emergency indicator is false, the system recognizes the situation as a typical accident, producing an alternative output with a blue screen containing calmer step-by-step instructions.
Input Validation: Underneath, the system checks inputController.text (textbox) to determine whether the user has typed something. If the text box is empty, a SnackBar will display "Please describe the situation" and halt the system to prevent a blank input from being sent for analysis.
UI Update: Beginning on line 48, a FocusScope causes the keyboard to be dismissed once the user's input is validated. The user's input is then transmitted to the external Gemini API backend, where the AI analyzes the input and returns emergency response steps. Once the system receives the AI response, the output's user interface is updated according to the AI's response. If the

analysis fails for any reason, the API will return a null value, causing a SnackBar to alert the user by displaying “AI could not provide instructions. Please try again.”

The Camera Page system allows users to take photos or select images from their photo library and receive a lab environment analysis according to the image. This system uses Gemini API backend alongside TensorFlow Lite to classify elements in visual images [12]. The component also relies on Machine Learning to recognize visual patterns from uploaded image data, allowing the system to accurately identify common labels such as GHS safety symbols.

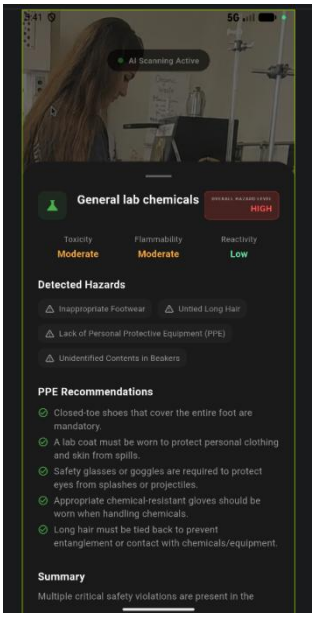


Figure 6. Camera Page interface for image-based hazard classification

```
try {
  final tfliteResult = await _tfliteService.classifyImage(File(image.path));

  final bytes = await File(image.path).readAsBytes();
  final base64Image = base64Encode(bytes);

  var gptResult = await _gptService.analyzeImage(base64Image, tfliteResult: tfliteResult);

  if (gptResult is String) {
    try {
      gptResult = jsonDecode(gptResult);
    } catch (e) {
      gptResult = null;
    }
  }

  if (mounted) {
    if (gptResult == null || gptResult is! Map) {
      _showError("Analysis failed or returned invalid format.");
    } else if (gptResult['status'] == 'emergency') {
      _showEmergencyAlert(gptResult);
    } else if (gptResult['status'] == 'error' || gptResult['status'] == 'irrelevant') {
      _showError(gptResult['message'] ?? "The image is not lab-related or too blurry.");
    } else {
      _showResult(gptResult, tfliteResult);
    }
  }
} catch (e) {
  if (mounted) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Analysis Error: $e'), backgroundColor: Colors.red),
    );
```

Figure 7. Implementation of the image classification and result-handling logic for the Camera Page

Classification: Starting from the top, the system uses _tfliteService to initiate the image classification process. After the initial classification, the image is converted into a Base64 string, which is then sent to _gptService for a secondary visual data analysis. After this secondary

analysis is finished, the data is returned in the form of a JSON response, which the system converts into a Map object. In the case that the conversion is unsuccessful, gptResult defaults to becoming null.

Result: Before generating any UI for the output, the state of the system is first validated using if (mounted). There are three possible result statuses in this step:

If gptResult is null, an error is displayed for the output.

If gptResult reveals an “emergency” status, an urgent alert is triggered.

If gptResult indicates “error” or “irrelevant”, a message appears explaining the photo is not lab related.

If any error occurs during the AI analysis, a red SnackBar notification is triggered, displaying a message containing the specific error to the user.

4. EXPERIMENT

This experiment is aimed at testing the accuracy of the visual classification system in the app’s Camera Page. Testing this feature holds significant value as false classifications could provide users with inaccurate feedback that could risk the user’s safety. Performing this experiment will provide important feedback on the system’s classification accuracy while revealing any shortcomings in the system.

The experiment’s dataset will consist of twenty diverse images that represent common laboratory environments, both safe and hazardous. Half of the dataset will include images that test the camera’s ability to detect visual elements under non-ideal conditions such as poor lighting and blurriness. The remaining images will test the system’s ability to accurately recognize GHS chemical symbols, lab equipment, and hazards. By utilizing a variety of lab images, this experiment captures the unpredictability of real laboratory environments that don’t always offer perfect visual conditions, providing practical feedback that closely reflects the system's performance in real, erratic laboratories.

	Input	Expected Output	Actual Output
1	Picture of Pollutant Symbol (Clear)	Environmental Hazard	Environmental Hazard
2	Picture of Chemical Symbol (Blurry)	Environmental Hazard	Environmental Hazard
3	Picture of Chemical Symbol (Very Blurry)	Environmental Hazard	Environmental Hazard
4	Picture of Carcinogen Symbol (Clear)	Toxicity Hazard	Toxicity Hazard
5	Picture of Chemical Symbol (Blurry)	Toxicity Hazard	Toxicity Hazard
6	Picture of Chemical Symbol (Very Blurry)	Toxicity Hazard	Toxicity Hazard
7	Picture Including Multiple Chemical Symbols (Clear)	Flammable Hazard Toxicity Hazard Environmental Hazard Oxidizer	Flammable Materials Serious Health Hazard Biohazard Environmental Hazard
8	Picture Including Multiple Chemical Symbols (Blurry)	Flammable Hazard Toxicity Hazard Environmental Hazard Oxidizer	Flammable Materials Serious Health Hazard Biohazard Environmental Hazard
9	Picture Including Multiple Chemical Symbols (Very Blurry)	Flammable Hazard Toxicity Hazard Environmental Hazard Oxidizer	Flammable Materials Serious Health Hazard Biohazard Environmental Hazard
10	Picture of Chemical Burn	(Warning!) Burn Detected	(Warning!) Chemical Burn
11	Picture of Large Flame	(Warning!) Fire Detected	(Warning!) Fire
12	Picture of Dangerously Excessive Fuming (Without Fumehood)	Fuming Hazard Inhalation Hazard Lack of Fume Hood	Inhalation Hazard Inadequate Fume Hood Containment Chemical Spill/Splash Risk
13	Picture of Improper PPE	Lack of PPE	Inappropriate Footwear Untied Long Hair Lack of PPE
14	Picture of Disorganized Lab Environment	Messy Lab Space	Cluttered Workplace Unlabeled or Poorly Stored Chemicals
15	Picture of Student Handling Chemicals Incorrectly	Improper Handling of Chemicals	Violation of Lab Safety Protocols
16	Picture of Inadequate Lab Safety Equipment	Missing Safety Equipment	N/A
17	Picture of Student Using Tools Incorrectly	Improper Use of Tools	Violation of Lab Safety Protocols
18	Picture of Student Using the Wrong Tools	Incorrect Safety Equipment	Chemical Spill/Splash Risk Sharp Hazard (Pipette Tip)
19	Picture of Clean High School Lab	Minimal Risks and Hazards Detected	Potential Biological Hazard Unidentified Chemical Presence
20	Satirical Picture of a Dog Chemist	Image Unrelated or Blurry	Image is not lab-related or too blurry

Figure 8. Classification results summary for the 20-image test dataset

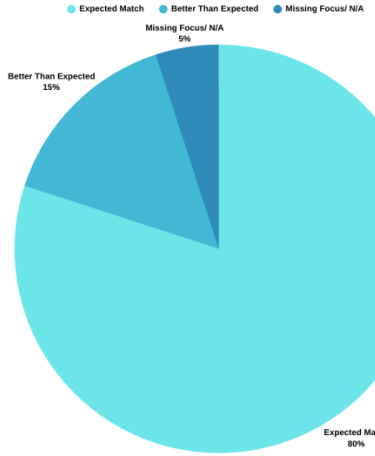


Figure 9. Sample outputs from the camera classification experiment

Out of the total 20 test inputs, the system correctly classified 19 images, resulting in an overall classification accuracy of 95%. Notably, 3 of the images received outputs that were superior to the expected outputs by noticing more safety details than expected.

While the system didn't make any false or inaccurate classifications, a single error was classified as "Missed Focus/NA" since the error resulted from the AI overlooking a subtle hazard rather than making an outright inaccurate statement. This result revealed that while the system excels at correctly identifying visible elements, it struggles with identifying implicit hazards, specifically

the absence of non-critical equipment like chemical racks. Standard detection models, as used in my system, naturally prioritize identifying visible elements over noticing missing items; however, this shortcoming primarily applies to missing tools that don't pose major risk, as the detection system in my app is programmed to specifically determine whether the user is using vital safety equipment such as PPE or gloves around a hot plate. Additionally, because the system can only evaluate visual data in the form of photos instead of videos, the system cannot evaluate one's physical technique, such as a student pipetting improperly. Overall, the system is more effective at identifying primary safety hazards rather than noticing niche items.

5. RELATED WORK

Yang et al. introduced the use of AI educational robots (AIERs) as a superior method of teaching lab safety to students [13]. These robots feature verbal interactions and Q&A systems, relying on verbal exchanges with students to boost their motivation and retention of safety content. Like my application (ChemSafe AI), robots use AI-powered conversations to keep students engaged during learning sessions. Unlike the robot, however, my application is purely digital, which allows it to simultaneously operate on multiple devices, benefitting a greater audience. Additionally, while robots can seem exciting at first, setting them up can be a tedious process. Finally, the presence of physical robots alone can become a safety hazard in chemistry labs; spilling chemicals on these robots can result in electrical fires, making mobile applications a safer and cheaper solution.

In response to rising lab accidents, this study introduces Virtual Reality (VR) as a potential solution for the ongoing crisis, addressing the lack of safety training by immersing users in simulations of chemical laboratories [14]. In this virtual world, users can train by completing safety tasks that prepare them for hazardous situations in real laboratories. Similar toVR training, my application also seeks to modernize lab safety education by shifting away from classroom lectures and instead using digital resources. However, while VR systems educate trainees through immersive visual training, my app prepares users through text-based guidance. Although VR systems can be very effective at helping users practice movement and perception, these systems are extremely pricey and technically demanding, making them inaccessible to average students and chemists. My project solves this issue, as it's a free application, making it accessible to most chemists. Also, my application can be used during experiments or standard lab activities, offering users real-time guidance instead of solely preparation outside the laboratory.

The study covers LabSafety, an Android application that provides pharmacy education by offering users safety content and self-assessment quizzes [15]. Compared with my app (ChemSafe AI), both systems share nearly identical purposes, acting as mobile applications that provide users with accessible safety education. Despite this, my app shares a key distinction; while LabSafety focuses on passive education aimed at preventing accidents through preparation, my app includes AI models that can deliver real-time guidance during experiments. This allows ChemSafe AI to offer more than just prevention tips, but also support users in case they become flustered or concerned during active lab sessions. In the case of a sudden accident, users can also use my app to receive instructions for an unpredictable situation, making ChemSafe AI a better provider for assistance in most cases.

6. CONCLUSIONS

A major limitation in my application is the heavy reliance on Gemini AI's API, which is used to power all the AI chatbot systems in my project. Since Gemini AI has usage limits, AI assistants in ChemSafe AI can only be used a limited number of times before the limit is met. Furthermore,

a high number of simultaneous users can lead to API throttling, causing request timeouts or service unavailability that can damage the user experience. This dependence on Gemini AI's API largely restricts the app's scalability and reliability during peak usage periods.

If given more time, I would love to resolve the usage and availability constraint by incorporating a larger number of AI services, such as ChatGPT and Claude AI, into the system. This would create a fallback system that automatically reroutes analysis requests to other AI APIs when Gemini AI becomes unavailable due to high traffic or if usage limits are met. Implementing this system would significantly improve user experience by reducing interruptions and increasing usage limits. Furthermore, I've also been working on implementing local Large Language Models, such as Qwen 2.5, to offload usage from external AI models like Gemini AI. Using these local models will allow processing to happen directly on the user's device rather than sending prompts over the internet, heavily reducing cloud computation tasks, which will mitigate API costs.

Aside from implementing more AI services, I want to take better advantage of the profile page, specifically the account creation feature. First, I want to implement a "Check History" feature that saves past responses and lab submissions to a user's account, allowing them to review past advice from AI assistants at a later time. Additionally, I have considered introducing subscription models that use paywalls in case API operational costs become a larger issue. Reflecting on the current app design also made me notice the creativity of certain features, which I would've liked to emphasize more by implementing better UI if I were to start over from day one.

REFERENCES

[1] Lv, Pengfei, Shucheng Zhu, and Lei Pang. "Statistical analysis of laboratory accidents in Chinese universities from 2011 to 2021." Process Safety Progress 42.4 (2023): 712-728.

[2] Estellés, Marta, and Andrew Doyle. "From safeguarding to critical digital citizenship? A systematic review of approaches to online safety education." Review of Education 13.1 (2025): e70056.

[3] Seyed, Alitabar Seyed Hadi, and Zohreh Zadhasn. "Digital mentorship: mentor-mentee relationships in virtual spaces." (2023): 12-18.

[4] Michaels, David, and Jordan Barab. "The occupational safety and health administration at 50: protecting workers in a changing economy." American journal of public health 110.5 (2020): 631-635.

[5] Chassignol, Maud, et al. "Artificial Intelligence trends in education: a narrative overview." Procedia computer science 136 (2018): 16-24.

[6] Ameri, Arefeh, et al. "Acceptance of a mobile-based educational application (LabSafety) by pharmacy students: An application of the UTAUT2 model." Education and Information Technologies 25.1 (2020): 419-435.

[7] Meyer, Phumrapee, et al. "AI-Powered Telepresence Laboratory: Generative AI Co-Pilot for Automated Experiment Guidance and Instruction." International Journal of Engineering Education 41.6 (2025): 1412-1432.

[8] Hesse, Robert G., et al. "Evaluating hazard symbols for the globally harmonized system (GHS) for hazard communication." Proceedings of the Human Factors and Ergonomics Society Annual Meeting. Vol. 54. No. 21. Sage CA: Los Angeles, CA: SAGE Publications, 2010.

[9] Ram, Bal, and Pratima Verma. "Artificial intelligence AI-based chatbot study of ChatGPT, Google AI Bard and Baidu AI." World Journal of Advanced Engineering Technology and Sciences 8.01 (2023): 258-261.

[10] Chang, Yupeng, et al. "A survey on evaluation of large language models." ACM transactions on intelligent systems and technology 15.3 (2024): 1-45.

[11] Wong, Gordon C., and Kevin C. Chung. "Natural language processing and large language models." Hand clinics 42.1 (2026): 27-36.

[12] Team, Gemini, et al. "Gemini: a family of highly capable multimodal models." arXiv preprint arXiv:2312.11805 (2023).

[13] Yang, Qi-Fan, Li-Wen Lian, and Jia-Hua Zhao. "Developing a gamified artificial intelligence educational robot to promote learning effectiveness and behavior in laboratory safety courses for undergraduate students." International journal of educational technology in higher education 20.1 (2023): 18.

[14] Poyade, Matthieu, et al. "A transferable psychological evaluation of virtual reality applied to safety training in chemical manufacturing." ACS Chemical Health & Safety 28.1 (2021): 55-65.

[15] Ameri, Arefeh, et al. "LabSafety, the pharmaceutical laboratory android application, for improving the knowledge of pharmacy students." Biochemistry and Molecular Biology Education 48.1 (2020): 44-53.

©2026 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.