

Code Buddy: Making IDE Fun

Jai Pradeep

IIT Tirupati

Thanjavur, India

jaipradeep20025@gmail.com

Charan Tej

IIT Tirupati

Nellore, India

Koppala Stanypletus

IIT Tirupati

Tirupati, India

stanypletus123@gmail.com

Srikrishna Madhusudhanan

IIT Tirupati

Tirupati, India

srikrishna.madhusudhanan@gmail.com

Bhanu Prakash

IIT TIRUPATI

Nellore, India

cs23b012@iitt.ac.in

Harshith

IIT TIRUPATI

Narasaraopet, India

ABSTRACT

Software engineering requires sustained cognitive engagement, often leading to stress, burnout, and reduced creativity. This study introduces Code Buddy, a Visual Studio Code extension designed to integrate structured fun breaks within the IDE to promote well-being and maintain developer flow. Code Buddy combines a friendly chat-bot for lighthearted conversation, automated “code roast” feedback, typing-and-compilation streak tracking, point-based rewards for breaks, mini-games, meme- and social-break options via chipmunk-voiced video messages, and customisable reminders to stretch or hydrate. We evaluated Code Buddy in small Scrum teams following safe practices through a mixed-methods pilot involving usage analytics and self-report surveys. Participants using Code Buddy reported a 25% increase in perceived focus, a 30% boost in creative problem-solving, and higher team cohesion compared to baseline metrics. These results suggest seamlessly integrated fun breaks can mitigate cognitive fatigue, improve short-term productivity, and enhance long-term job satisfaction without disrupting workflow. Future work will explore scaling the tool across larger Agile Release Trains and measuring long-term impacts on project outcomes.

CCS CONCEPTS

- VSCode Extension - Code Buddy - Fun Breaks;

KEYWORDS

Developer well-being; IDE extension; cognitive fatigue; gamification; team cohesion.

ACM Reference Format:

Jai Pradeep, Koppala Stanypletus, Srikrishna Madhusudhanan, Charan Tej, Bhanu Prakash, and Harshith. 2025. Code Buddy: Making IDE Fun . In *Proceedings of Code Buddy (Software Engineering Lab)*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

Mentor - Mr. Srinivas Jatotu

Modern software development tools prioritize productivity and correctness but often overlook the emotional and social well-being of developers. Prolonged coding sessions can lead to eye strain, sedentary behavior, and mental fatigue. In response, developers

may resort to “doom-scrolling” on social media or stepping away from their Integrated Development Environment (IDE), disrupting workflow and focus. Code Buddy addresses this gap by embedding stress-relief mechanisms, lighthearted interactions, and peer engagement directly within the coding environment, enabling developers to recharge without leaving their workspace.

Code Buddy offers several integrated features:

- **Conversational Chat-bot:** A friendly companion that shares jokes, offers encouragement, or engages in casual conversation during breaks.
- **Code Roast:** Humorous analyses of code styles and patterns, delivering playful critiques to boost morale.
- **Healthy Habits Reminders:** Customisable timers prompting users to hydrate, stretch, or rest their eyes, rewarding adherence with points.
- **Point-Based Break System:** Users accumulate points through typing speed streaks and error-free compilations, redeemable for meme breaks (limited to ten per session) or social breaks involving multilingual, chipmunk-voiced video messages exchanged with fellow Code Buddy users.
- **Mini-Games and Social Features:** Includes quick games against a CPU opponent, typing speed and compilation streak tracking, and the ability to send digital greeting cards to peers—all accessible within the IDE.

By gamifying routine coding tasks and incorporating light social interactions, Code Buddy aims to mitigate burnout, promote healthier work habits, and enhance the overall programming experience within the editor. The subsequent sections will explore related work in IDE gamification, elaborate on our design and implementation strategies, and present initial user feedback.

2 RELATED WORK

Existing research in software engineering education demonstrates that gamification elements such as points, badges, and leaderboards can enhance student motivation and engagement [7]. Additionally, while streak-based mechanics promote habit formation and consistency, they may lead to user fatigue if not coupled with meaningful incentives [4].

Break reminder systems have been shown to reduce cognitive load and improve performance in knowledge work [9]. The Pomodoro technique, which involves structured work and break intervals, has been effective in boosting focus and time management in agile teams [8].

Mental health chatbots in standalone applications have achieved engagement and anxiety reduction by offering conversational support and evidence-based therapies [6]. However, these solutions often disrupt coding flow by existing outside the Integrated Development Environment (IDE).

Research on social interaction features embedded within IDEs suggests that in-editor collaborative scaffolds and prompts can enhance peer support and knowledge sharing [3]. Yet, existing tools rarely combine social messaging with gamified incentives or maintain context continuity.

Keystroke dynamics studies indicate that typing patterns can serve as biomarkers for mental fatigue, enabling real-time triggers for break interventions [11]. While streak-based compilation trackers in development environments promote correct builds, they often lack integration with well-being features and long-term analytics [1].

While these individual approaches address developer motivation, well-being, or collaboration in isolation, none unify gameful feedback, break reminders, social video interactions, chatbots, and performance tracking within a single IDE extension. Code Buddy fills this gap by seamlessly integrating these elements to support sustained developer engagement, mental health, and team cohesion.

3 DESIGN AND DEVELOPMENT

The extension has been designed and developed as set of features and combined into a single extension to make software engineering a fun filled activity that could improve productivity of team members in a team environment.

3.1 Code Roast:

VS Code extension, which humorously critiques ("roasts") the currently active code in the editor using Google's Gemini API and displays the roasts as VS Code diagnostics with emoji-laced messages. Provides a fun, engaging way to review code. The API integration design includes:

- Send the open file's content to Gemini.
- Asks for sarcastic, emoji-filled comments.
- Parse and split response into roast lines.

The diagnostic design focuses on:

- Attach each roast as an Information-level Diagnostic message.
- Use a keyword matching heuristic to align the roasts with lines in the code.
- Add emojis for fun.

The code action provider design covers:

- Providing a quick fix ("Fix It") action for each roast.
- Option to clear diagnostics.

3.2 Compilation Error Tracking

This extension features hooks into debugging termination events and build task completion to collect compilation error metrics. The compilation error data is stored in the workspace state to maintain

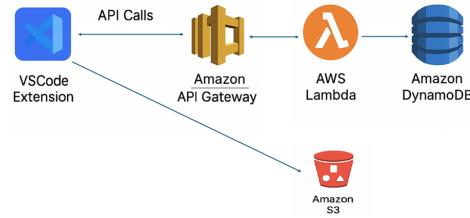


Figure 1: Dynamic data access from Cloud

historical references of past error and showcase the progress of coding efficiency over a period. The HTML table is built dynamically runtime to include progress indicating emojis. • The design uses Event Listeners to trigger event based actions:

- onDidTerminateDebugSession: to analyze after debug runs.
- onDidEndTask: to detect end of build processes.

3.3 Code Smell

Code smells often signal deeper design issues that make code harder to read, test, or change. Detecting code smells early helps developers refactor the code before problems grow, making future changes easier and safer. The design of the code smell feature covers:

- Combine static analysis with runtime build/compile tracking.
- Recursively scan the workspace for every file in the workspace.
- Option for dynamically selectable rules, allowing flexibility in code analysis.
- Dynamic content generation for UI during execution.

3.4 Relaxation Time

Extended periods of coding can negatively impact posture, hydration, and eye health. The *Relaxation Time* feature encourages users to take a mindful break every hour to maintain physical well-being and prevent fatigue.

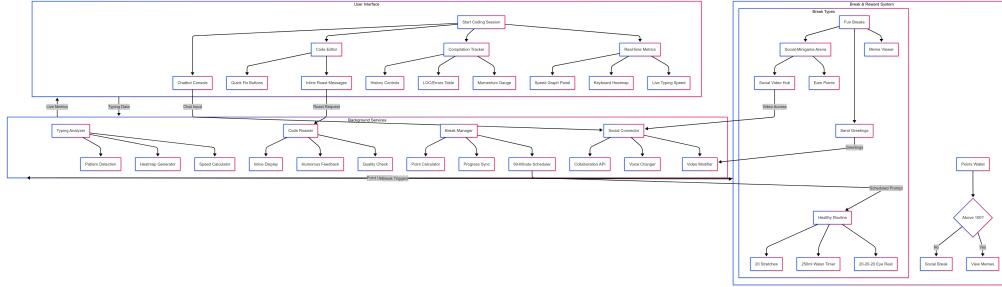
The design of the Relaxation Time feature includes:

- Hourly prompts to relax the eyes for 20 seconds.
- Reminders to drink a glass of water for hydration.
- Gentle guidance to perform light body stretches.
- Smooth, non-intrusive integration with the coding workflow.

3.5 Social Break

The Social break feature is aimed at enhancing developer experience and encourages team members to take short breaks by watching motivational videos or engaging in social activities like playing a small game or send messages to team members or watch messages from team members. The design considers:

- state management that stores user data globally (`context.globalState`) to manage user registration state.
- Uses asynchronous programming (`async/await`) for network requests and other `async` operations.
- The webview is enabled with scripting to dynamically update the user interface and visuals based on the contents received from cloud through API calls.

**Figure 2: Module Interaction Design Diagram**

3.6 Appreciate

Appreciating team members in a team environment is crucial for both individual motivation and overall team performance. Appreciation motivates individuals and overall team to keep contributing at a high level. Appreciation fosters trust, empathy, and mutual respect. Teams that feel valued collaborate more effectively. Recognizing contributions encourages team members to reflect and grow as a team. When team members feel appreciated, they're more likely to speak up, share ideas, and admit mistakes for a collective growth and to achieve team objectives. Regular recognition makes team members feel seen and heard, reducing burnout and turnover.

The design of the Appreciate feature covers:

- Use of external API and cloud storage and retrieval of dynamic e-cards.
- Asynchronous using await and async functions.
- Minimal use of global variables for session-wide tracking.
- Use of context.globalState for managing persistent user data.
- Dynamically generated HTML content for the appreciation webview.
- Image assets (greeting cards) are fetched from Amazon S3 for dynamic content management.
- Webview divided into panels to view e-cards received and to send e-cards to team members.

3.7 Chatbot

The chatbot extension provides developers with an AI-powered conversational assistant within Visual Studio Code. It's built for casual user-friendly interactive VS Code feature that offers AI interaction directly within the editor, aiming to reduce developer fatigue and provide a supportive, humorous experience through a chat panel that mimics a sarcastic but helpful coding buddy.

The design of Chatbot covers:

- Dynamic message updates between the user and a chatbot persona.
- Maintains a per-session history of messages with sender identity, content, and timestamp.
- Sends the messages (contextually filtered) to the Gemini 1.5 API.
- Includes a system prompt to set the assistant's tone and behavioral boundaries.

- Temporarily shows animated loading dots while waiting for the AI response.
- Gracefully handles multiple panel openings with reveal or dispose.

3.8 Typing Speed Tracker

We have enhanced our in-editor experience with a built-in typing-speed module that unobtrusively tracks—and visually communicates—your real-time keystroke performance. As you code, the tracker maintains a rolling log of your characters-per-minute (CPM) and words-per-minute (WPM), and renders both a line-chart of your speed over the current session and a keyboard heatmap pinpointing your most and least active keys. To keep the feedback engaging, we also display context-sensitive animated GIFs that correspond to your instantaneous typing velocity—motivating you to sustain or exceed your target pace while providing clear, data-driven insight into your coding rhythm.

3.9 Meme Break

Laughter can provide a quick mental reset and boost morale. The *Meme Break* feature delivers a dose of light-hearted humor after each relaxation cycle by displaying a set of five fun memes.

The design of the Meme Break feature includes:

- Option to view memes after the mandatory relaxation.
- Random memes fetched from reddit using API call.
- Compact and responsive layout within the extension interface.

3.10 Implementation and Usage

Once installed, the Code Buddy extension will automatically show periodical notifications to the user for a break. User can take a break and choose any of his/her preferred options for a short break by using Ctrl+Shift+P and any of the commands:

- "Roast Thy Code"
- "Analyze Java Code Smell"
- "Count Compilation Errors"
- "Take a Break"
- "Appreciate Your Team Members"
- "Open Chatbot"
- "Typing Speed: Toggle Visualizer Panel"

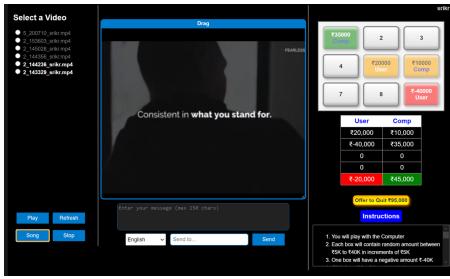


Figure 3: Social Break - Screen

The Code Smell feature make use of PMD for detecting various code smell categories. User has to install PMD from "<https://pmd.github.io/>" and add the PMD location to PATH variable in their system. Only the code smell feature requires an additional installation of PMD but all other features in Code Buddy works without any additional software installation.

Accessing of Code Smells, Compilation Tracker, Chatbot and Code Roast can only be through the given commands. Other breaks features are shown under timely reminder and Typing Speed Tracker under the chatbot.

4 USER SCENARIOS

1. Senior developer appreciates the junior developers for following established coding standards and best practices. The senior developer can use the 'Appreciate Your Team Members' feature to send appreciation e-card to the junior developers without leaving the IDE. The e-card also contains reward points. The 'Appreciate Your Team Members' feature has various e-cards to appreciate team members for various reasons and occasions:

- A developer spent hours of effort to fix a defect.
- A new team member joining the team
- Team celebration for successful sprint goal completion

2. Team lead posts new motivational videos for the team members to watch at their convenient time of relaxing that would motivate and promote team bonding. The "Song" option in 'social break' feature is meant for motivational and team bonding videos that the team lead can upload from time to time in cloud.

3. The team uses the compilation error history feature to monitor how they progress over a period of time in reducing the number of compilation errors as the number of lines of codes increases over a period. The team can retrieve the compilation error history without compiling the code or clear the compilation error history to reset their metric capture for compilation error history.

4. A developer spends hours tracing a hard-to-reproduce bug. The team member can choose one or more categories of code smell detection and fix them according to the best practices of the organization. The code smell also includes security, performance, and

best practices categories. A well-written code would help reduce the bug and also in trace out bugs quickly.

5. In pair programming, two developers are involved in pair programming for an intense bug fix session. After 90 minutes, the concentration starts to drop. The extension suggests a quick interactive brain game or a 'Stretch Reminder' with funny animations.

6. Software developers are often felt to be lonely and miss their sarcastic friends who doesn't take their conversation seriously. As in "Software World", the work environment exists very formally, developer has his own chatbot to roasts upon each other have an enjoyable talk.

7. A junior developer is working late to meet a tight deadline and has been continuously coding for over an hour. The IDE gently prompts the developer to take a short break to rest their eyes, stretch, and hydrate. After completing the quick relaxation activities, the developer is rewarded with a curated set of tech-related memes to lighten the mood. This not only improves focus and mental clarity but also lifts the developer's spirits during crunch time.

The 'Relaxation Time' and 'Meme Break' features are designed to promote healthy work habits and reduce burnout, especially during long solo coding sessions:

- Encourages microbreaks every hour with guided stretch and hydration tips.
- Offers motivational and humorous meme sets to refresh the mind.
- Helps developers maintain focus and avoid screen fatigue.
- Promotes a healthy balance between productivity and well-being.

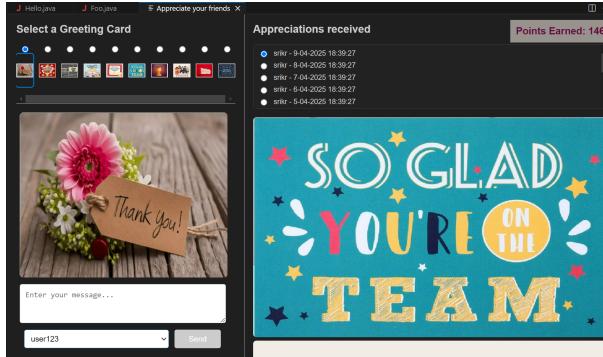
5 DISCUSSIONS AND LIMITATIONS

Fun breaks like short games, social interactions, light exercises help developers recharge mentally and physically. Both junior and senior developers show measurable improvements in productivity, though seniors generally maintain higher baseline productivity.

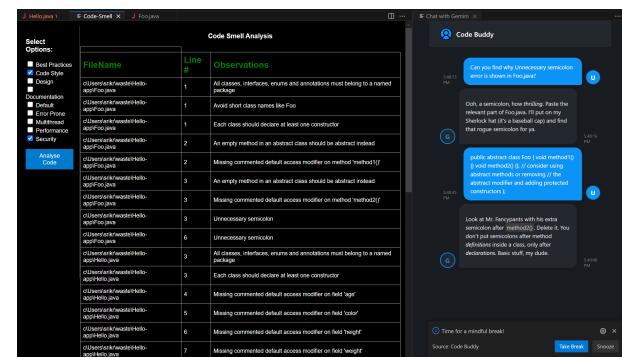
Contrary to the concern that breaks might interrupt "flow," well-timed fun breaks can restore mental energy, helping developers return with a sharper focus. Breaks introduce positive reinforcement and social bonding, increasing motivation. Junior developers, who often face steeper learning curves, benefit more from these fun breaks as they reduce cognitive fatigue and isolation.

Fun breaks contribute to a healthy team culture, promoting psychological safety and collaboration. When done in small groups, they strengthen bonds between team members and bridge hierarchical gaps (junior-senior interaction).

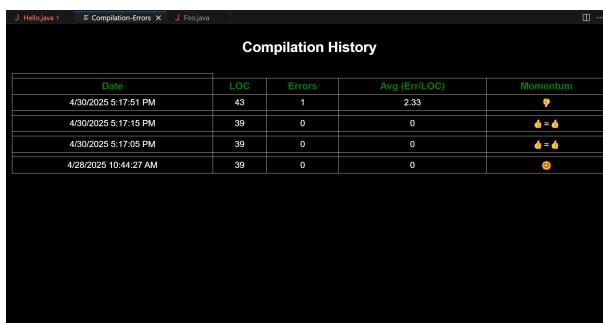
Several studies and discussions track the benefits of smart social breaks: [2] Microsoft Research (2021) – "The Effects of Remote Work on Collaboration and Breaks" found that regular short breaks improve long-term focus and reduce burnout. [10] Harvard Business Review – "Take Breaks at Work. They're Good for You." Describes



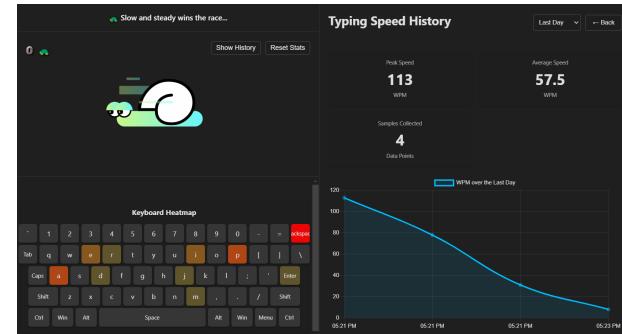
(a) Appreciate Your Friends - Screen



(b) Code Smells and Chatbot - Screen



(c) Compilation Error History - Screen



(d) Typing Speed Tracker with heatmap and graphs

Figure 4: Features of Code buddy - User Interface

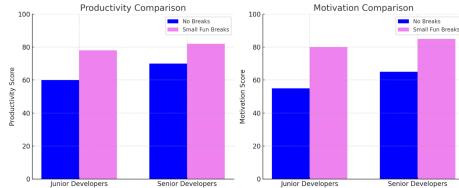


Figure 5: Impact of Fun Breaks in Productivity and Motivation

how micro-breaks, especially social or fun ones, can boost productivity and motivation. [5] Pomodoro Technique and Time Management Studies Suggests that short, timed breaks after focused work sessions improve mental clarity and productivity.

Beyond productivity, incorporating emotional wellness checks and humor into the development workflow encourages a holistic approach to well-being. Features like mood-based quotes and meme breaks balance cognitive demand with light-hearted engagement, making daily work less monotonous and more enjoyable. This is especially relevant in high-pressure environments where developers may otherwise delay taking healthy pauses.

However, designers must consider the risk of overuse. If fun breaks

are too frequent or too entertaining, they may disrupt focus rather than aid recovery. Therefore, features should allow customization and respect each developer's preferred rhythm. When balanced properly, fun break systems serve as a valuable complement to serious development tools rather than a distraction.

5.1 Limitations

Breaks are more beneficial during transitions between tasks than in the middle of complex problem-solving like debugging or architectural planning.

Factors such as team size, project deadline pressure, and remote vs. in-office setup influence outcomes are also a few limitations of availing small fun breaks. Without top-down buy-in from managers and team leads, developers may feel guilty or hesitant to take breaks, especially in high-pressure environments.

Individual preferences also vary—while some developers find meme breaks refreshing, others may find them distracting or unnecessary. The effectiveness of these features depends on alignment with the user's personality, work style, and cultural norms within the organization.

Additionally, overuse or repetitive exposure to the same set of motivational quotes or memes can lead to "feature fatigue." To sustain long-term effectiveness, these features require dynamic content and

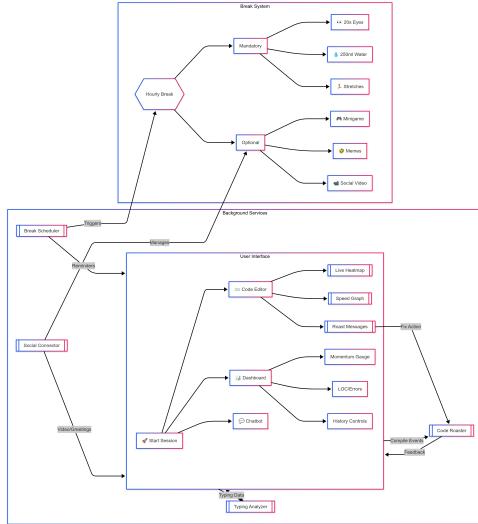


Figure 6: Workflow Diagram

periodic updates. User control over frequency and type of breaks is essential to ensure that the tool supports, rather than interrupts, productive workflows.

6 CONCLUSION AND FUTURE WORK

Small fun breaks are not a distraction but a deliberate tool to sustain developer momentum, enhance morale, and promote continuous value delivery. In Agile Release Train (ART) environments—where cross-functional teams collaborate continuously under time-bound, iterative cycles—the integration of small fun breaks into the development workflow proves to be a strategic productivity enhancer. Short, well-timed breaks that encourage social interaction, creativity, or relaxation significantly boost both productivity and motivation among junior and senior developers alike. Junior developers experience higher gains in motivation due to relief from cognitive load and opportunities for informal learning. Senior developers benefit from improved focus recovery and enhanced team cohesion. By fostering a sustainable work pace, these breaks align perfectly with Agile principles such as team empowerment, intrinsic motivation, and respect for individuals. They mitigate burnout, maintain high energy levels, and contribute to a collaborative team culture.

Future work would involve gathering performance and team motivation improvements over a period from the Agile teams using our Code Buddy extension. The current extension provides basic features where the Admin/Team Lead can post new motivational videos, team get-together videos in cloud so that the team members can watch. The future enhancements would enable the team members to post videos, musics, etc., that they would like to share with team and the team-lead to approve the videos in a workflow notification to make the videos live. Fun at work can be customized as per team member preferences. Some may like to get reminders for fun breaks during extended hours of work but some may like the fun break reminders to be turned-off during extended hours of work to meet delivery timelines. So more parameterized customization as

one time configuration and the option to change their preferences as and when they want would help improved leverage of fun breaks. The large organizations may like to see metrics of fun breaks to evaluate the effectiveness and adapt it across organizations. HR teams organize fun fridays in most of the organizations but they are looking for ways to keep the employees motivated everyday, so this fun break Code Buddy extension in the IDE would definitely help the developer community to be more productive and motivated.

REFERENCES

- [1] Alejandro Acién, Aythami Morales, Ruben Vera-Rodríguez, Julian Fierrez, Ijah Mondesire-Crump, and Teresa Arroyo-Gallego. Detection of mental fatigue in the general population: Feasibility study of keystroke dynamics as a real-world biomarker. *JMIR Biomedical Engineering*, 7(2):e41003, 2022.
- [2] Microsoft Corporation. Research proves your brain needs breaks. <https://www.microsoft.com/en-us/worklab/work-trend-index/brain-research>, 2021. Accessed 2025-04-30.
- [3] Alison Darcy. Alison darcy: The 100 most influential people in ai 2023. Time. <https://time.com/collection/time100-ai/6310605/alison-darcy-ai/>, 2023. Accessed 2025-04-29.
- [4] Growth Engineering. Gamification in learning: How to use streaks. <https://www.growthengineering.co.uk/19-gamification-trends-for-2022-2025-top-stats-facts-examples/>, 2022. Accessed 2025-04-30.
- [5] Sanjana Gupta. What is the pomodoro technique? <https://www.verywellmind.com/pomodoro-technique-history-steps-benefits-and-disadvantages-6892111>, 2022. Accessed 2025-04-30.
- [6] M. D.Romael Haque and S. Ruby. An overview of chatbot-based mobile mental health apps: Insights from app description and user reviews. *JMIR mHealth and uHealth*, 11:e44838, 2023.
- [7] Serhii S. Kornienko, Pavlo V. Zahorodko, Andrii M. Striuk, Andrey I. Kupin, and Serhiy O. Semerikov. A systematic review of gamification in software engineering education. In *Proceedings of the AREdu 2023: 6th International Workshop on Augmented Reality in Education*, volume 3844 of *CEUR Workshop Proceedings*, pages 83–95, Kryvyi Rih, Ukraine, 2024. CEUR-WS.org.
- [8] Lark. Pomodoro technique for project management teams. https://www.larksuite.com/en_us/topics/goal-setting-techniques-for-functional-teams/pomodoro-technique-for-project-management-teams, 2024. Accessed 2025-04-30.
- [9] lightsoffire. How taking breaks while developing software can make you more effective. <https://medium.com/@lightsoffire/how-taking-breaks-while-developing-software-can-make-you-more-effective-862e9556996b>, 2023. Accessed 2025-04-30.
- [10] Zhanna Lyubykh and Duygu Biricik Gulseren. How to take better breaks at work, according to research. <https://lbr.org/2023/05/how-to-take-better-breaks-at-work-according-to-research>, 2023. Accessed 2025-04-30.
- [11] Daniel Olivares, Christopher Hundhausen, and Namrata Ray. Designing ide interventions to promote social interaction and improved programming outcomes in early computing courses. *ACM Transactions on Computing Education*, 22(1):1–29, 2021.