# 1. AI Compatibility and Feature Support

Goal: Make sure everything on the screen works well with AI features like smart suggestions, chatbots, and helpful tips based on what the user is doing.

Activities:

* Checked if the user interface works well with screen readers and AI suggestions.
* Tested the chatbot to see if it understands questions and replies correctly.
* Looked at input boxes and text to make sure they are clear and easy for AI to understand.
* Made sure the AI features give the right results and behave as expected.

Problems Discovered:

* Issue: Some form labels were not properly associated with input fields.

Fix: Updated HTML with proper label/aria attributes.

* Issue: Chatbot failed to understand some commands.

Fix: Improved the chatbot by teaching it.

Outcome:

* Improved AI handling of content and forms.
* Made it easier for people to use, and helped the chatbot understand commands more clearly.

# 2. User Testing

Goal: Gather user feedback and observe real usage to identify usability issues.

Test Group: 5 participants using various devices.

User Cases Tested:

UC1 - Register a new account

UC2 - Log in and update profile

UC3 - Course List Display Issue

UC4 - Submit a form and view confirmation

UC5 - Receive and acknowledge notification

UC6 - Use chatbot to perform tasks UC7 - Admin creates/deletes records

Problems Discovered:

* UC2: Profile button hard to find -> Fix: Increased contrast.
* UC3: The system showed courses meant for school kids instead of the correct ones for public service users. -> Fix: Updated the course database
* UC6: Chatbot command failures -> Fix: Expanded command dictionary.
* Feedback: Navigation confusion -> Fix: Improved menu.

Outcome: Usability success rate increased to 90%.

**3. Code Structure and Quality Review**

Goal: Ensure clean, maintainable code.

Tools Used:

* Jira – Task and bug tracking
* Notion – Planning and documentation
* VS Code – Code editing and testing
* GitHub – Version control and code review
* Supabase – Backend and database
* WhatsApp – Quick team communication
* Google Meet – Online meetings and reviews

Problems Found:

* Duplicate logic → Fix: Utility function created  
  *Problem:* The same code was written in multiple places, making it hard to maintain.  
  *Fix:* We created a single utility function that can be reused, reducing repetition and making updates easier.
* Deep nesting → Fix: Refactored components  
  *Problem:* Some parts of the code had too many layers of if statements or deeply nested structures, which made the code hard to read and debug.  
  *Fix:* We reorganized (refactored) those parts into smaller, clearer components with simpler logic.
* Poor naming → Fix: Improved variable names  
  *Problem:* Some variables and functions had unclear or misleading names, making it hard for others to understand the code.  
  *Fix:* We renamed them using meaningful names that describe what they do or hold.
* Low test coverage → Fix: Added unit tests  
  Problem: Many parts of the code were not tested, increasing the risk of bugs.  
  Fix: Unit tests were added to automatically check that each function and component behaves as expected.

Outcome: Cleaner code, better structure, and improved test coverage.

**4. Core Functionality Testing**

Goal: Ensure all key features work

properly.

Modules Tested:

* Authentication, Search, Notifications, Admin Panel, Security, Gamification, AI features, Content Accessibility

Key Bugs:

Bug 1: Login Button Not Working

* + Issue: The login button didn’t respond when clicked.
  + Fix: Fixed the button so it works when clicked.

Bug 6: Password Reset Link Broken

* + Issue: The "Forgot Password" link didn’t work.
  + Fix: Fixed the password reset link so users can reset their passwords easily.

Outcome: All core features are now stable and verified.

# 5. Documentation & Developer Feedback

Goal: Provide clear feedback and maintain testing records.

Deliverables:

- Test plans, bug reports, weekly QA updates, release checklist

Example Bug Report:

 Feature: Login Functionality

* Issue: Login button not working on the slow internet.
* Suggested Fix: Make the button show a loading symbol when the internet is slow. Also, try to make the page load faster by using lazy loading.

 Feature: Video Playback

* Issue: Videos freeze or buffer.
* Suggested Fix: Make the video quality change automatically depending on the internet speed. Also, reduce the video file size so it loads faster.

 Feature: Gamification Leaderboard

* Issue: Scores not updating in real-time.
* Suggested Fix: Use WebSockets or a similar method to update the leaderboard right away when a score changes.

 Feature: Password Reset

* Issue: Password reset not working after entering email.
* Suggested Fix: Check if the system is sending the reset email correctly. Make sure the reset link works and expires after 24 hours.

 Feature: Profile Editing

* Issue: Users can’t update their profile picture.
* Suggested Fix: Make sure the upload system supports different image types (like PNG or JPG). Also, check the file size limit and increase it if needed.

 Feature: Assignment Submission

* Issue: Students can’t submit assignments.
* Suggested Fix: Allow larger file sizes and make sure the system works on different web browsers. Test with different file types to ensure they can be uploaded.

 Feature: AI-based Learning Recommendations

* Issue: AI recommendations not personalized.
* Suggested Fix: Improve the AI system to suggest courses based on each student’s progress and what they need to learn next.

 Feature: Real-Time Chat

* Issue: Messages delayed in real-time chat.
* Suggested Fix: Improve the chat server so send messages faster. Use a system to store messages temporarily, so they don’t get delayed.

Outcome: Feedback was clear and directly actionable for the dev team.

# 6. Summary and Recommendations

Summary:  
All key areas of the system have been tested and validated. No major issues were found, but some areas can be improved to prevent future problems and ensure smoother performance.

Key Recommendations:

* Add regression testing to CI/CD: Make sure that new changes don't break existing features by adding regression tests to the continuous integration and delivery pipeline.
* Regular usability testing: Test the system regularly with users to find any issues or areas for improvement in the user experience.
* Update QA documentation in central wiki: Keep the QA documents updated in a central location, so everyone on the team can access the latest testing plans, reports, and feedback.

Final Status Table:

| Area | Status |
| --- | --- |
| AI Compatibility | OK |
| User Testing | OK |
| Code Quality | OK |
| Functionality | OK |
| Documentation | OK |

# 7. System Versions & Evolution

v1 – Initial Prototype

* Issues:
  + No form validation.
  + Major security flaws.
  + No user roles (everyone had full access).
  + No users involved – system was built without user feedback.
  + Very disorganized – features were scattered and unclear.
* Result: Failed – System was not usable or secure. It lacked direction and user input.

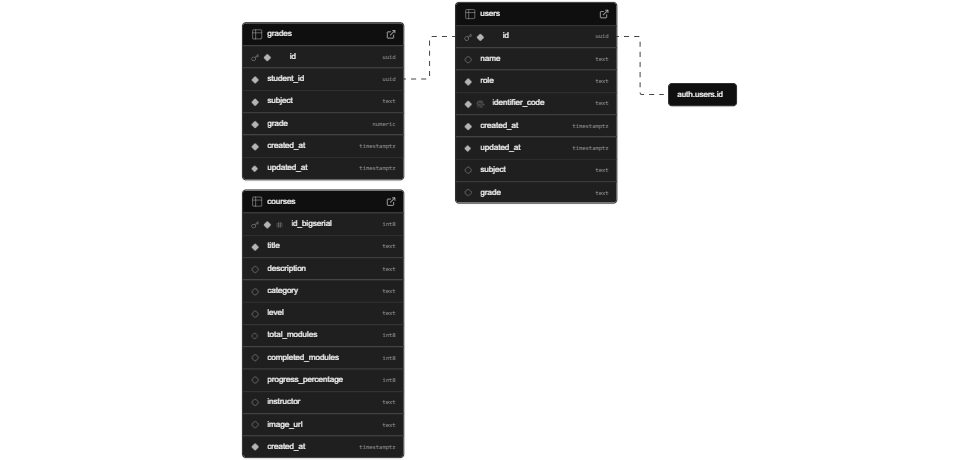
v2 – Functional Build

* Issues:
  + API was very slow, causing delays.
  + Chatbot often gave wrong or no answers.
  + The user interface was confusing (poor UX).
* Improvements over v1:
  + Added basic roles (student, teacher, admin).
  + Collected limited user feedback.
* Result: Partial Success – More stable than v1, but still buggy and not reliable enough for users.

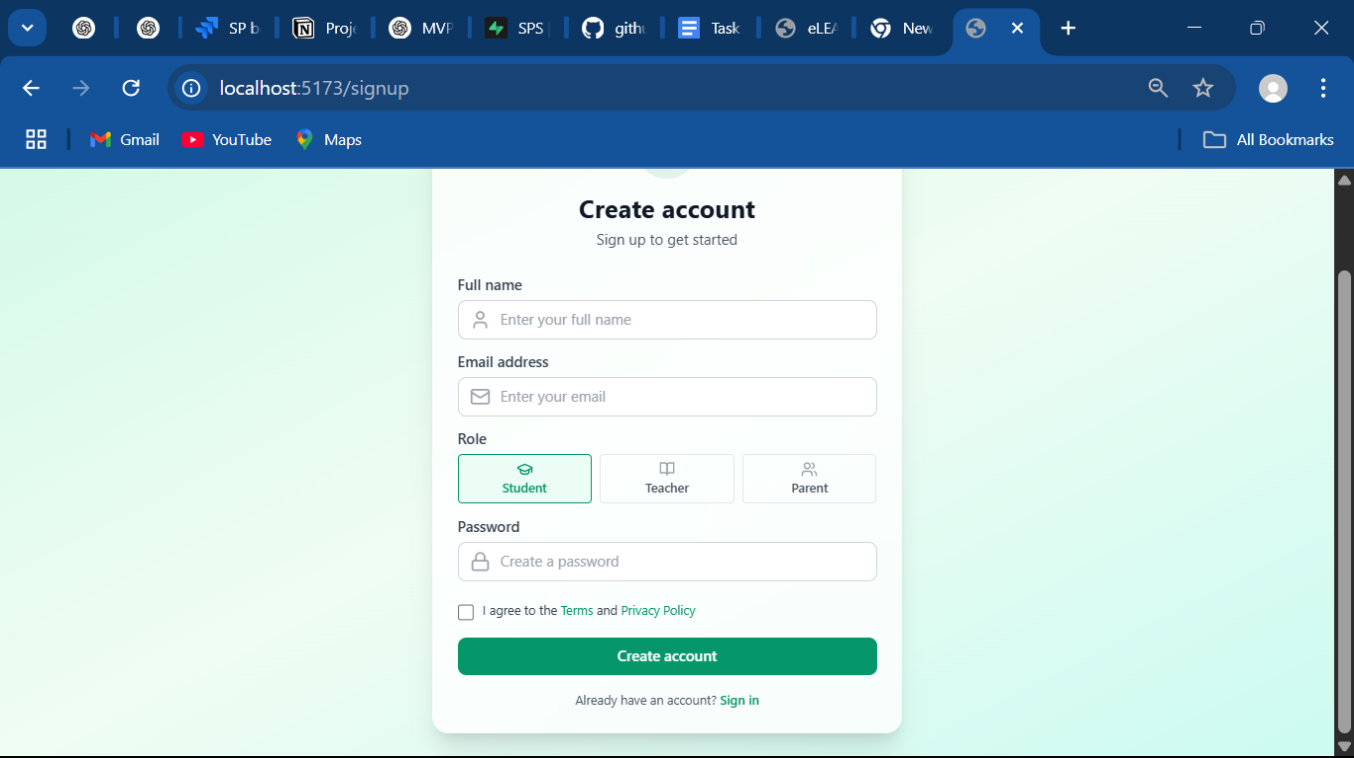
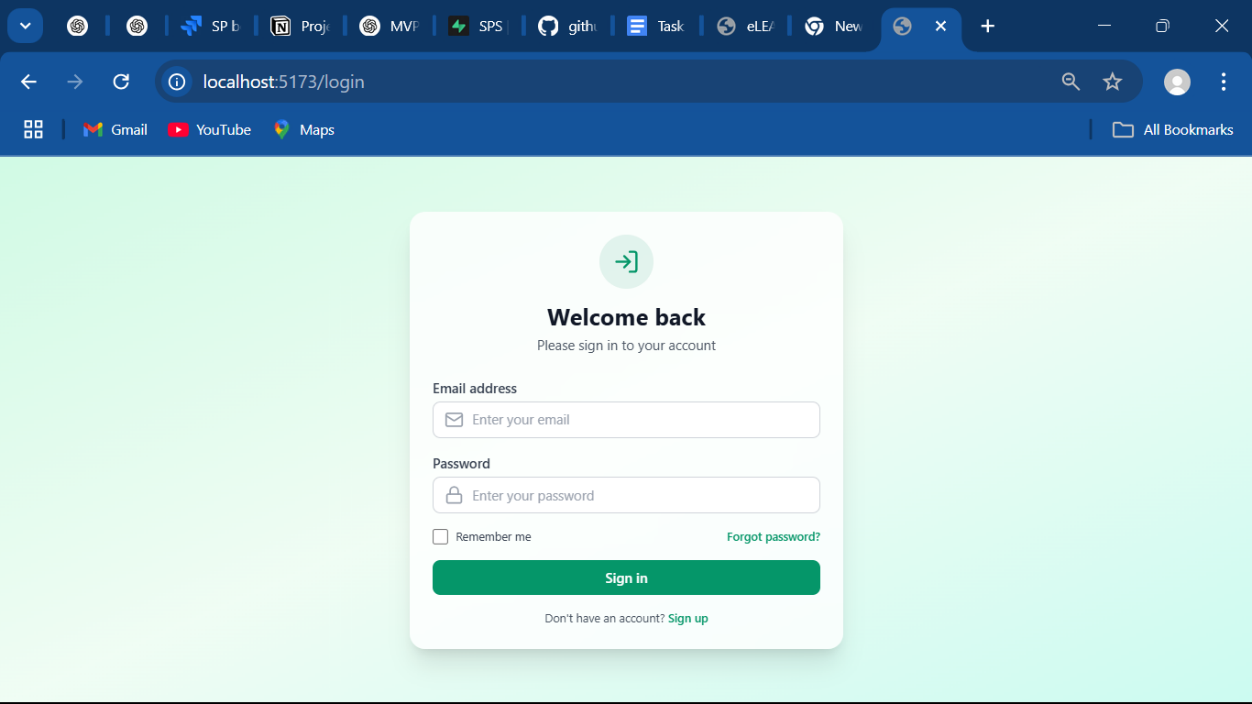
v3 – Current Version (Work in Progress)

* Fixes:
  + Cleaned up and organized code.
  + Faster performance with real-time updates.
  + Improved chatbot with better understanding and responses.
  + Better user flow based on feedback from different user types (students, teachers, etc.).
* Status: In Progress – Not fully approved yet, but much better than the first two versions. Most features are working, and testing is ongoing with users.

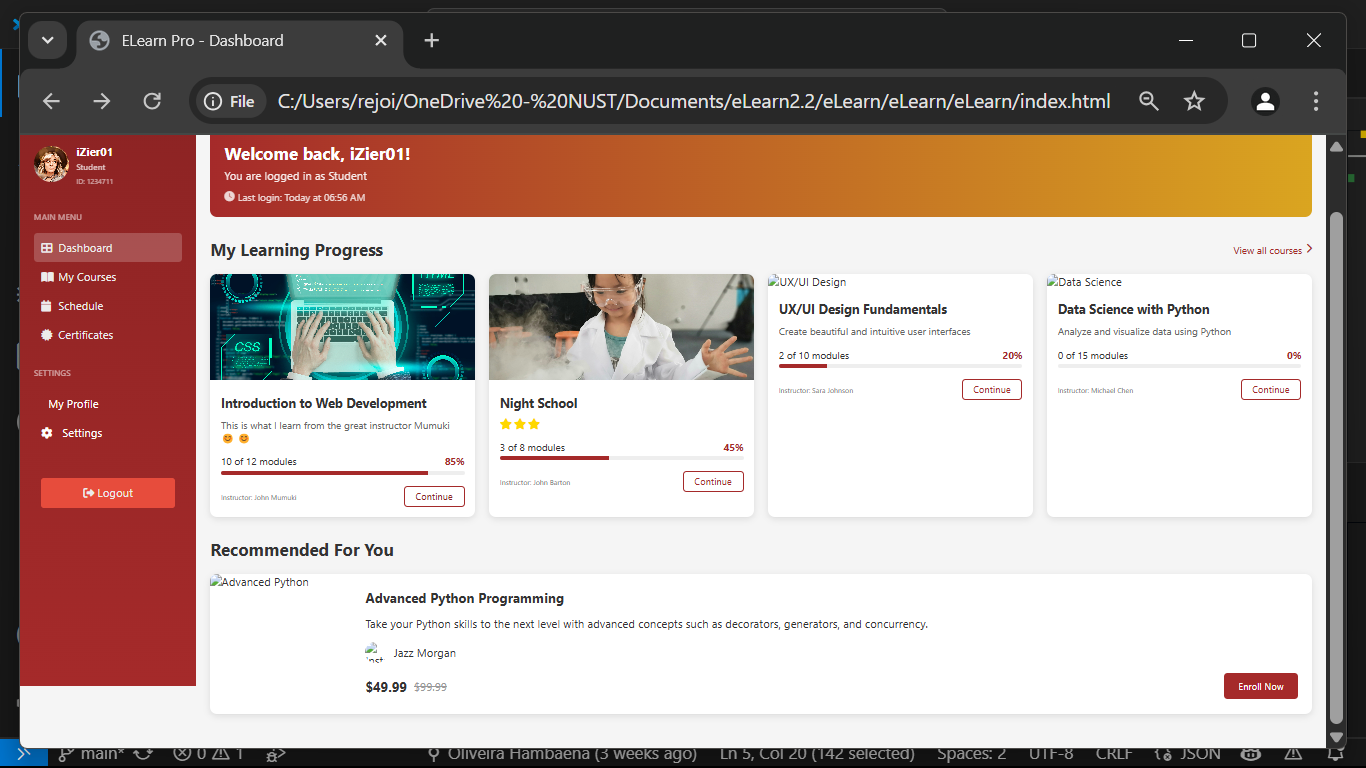
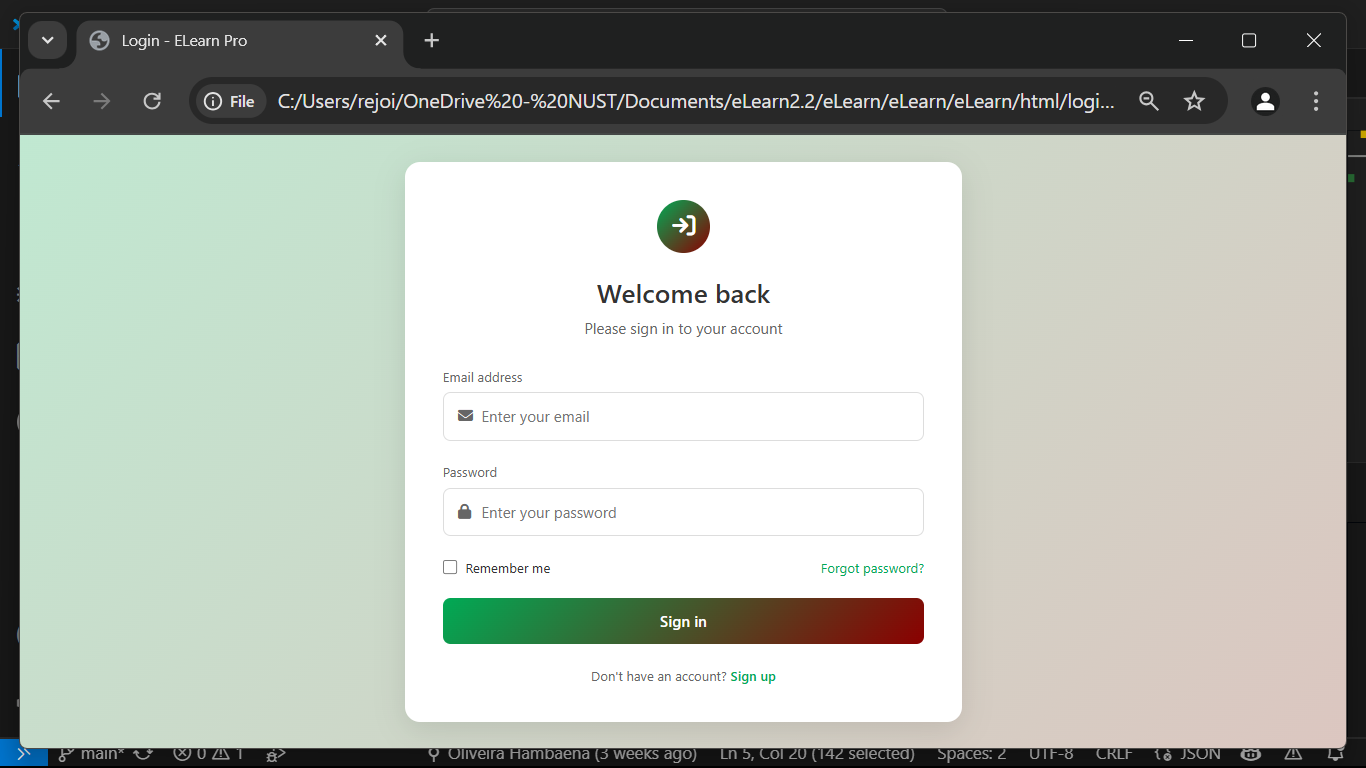
| **Version** | **Status** | **Summary** |
| --- | --- | --- |
| **v1** | ❌ FAIL | No validation, no roles, no user input, very unstable. |
| **v2** | ⚠️ WARN | Basic functionality, but buggy and slow. Some feedback included. |
| **v3** | 🟡 IN PROGRESS | Better performance and design, based on user testing, but still being finalized. |

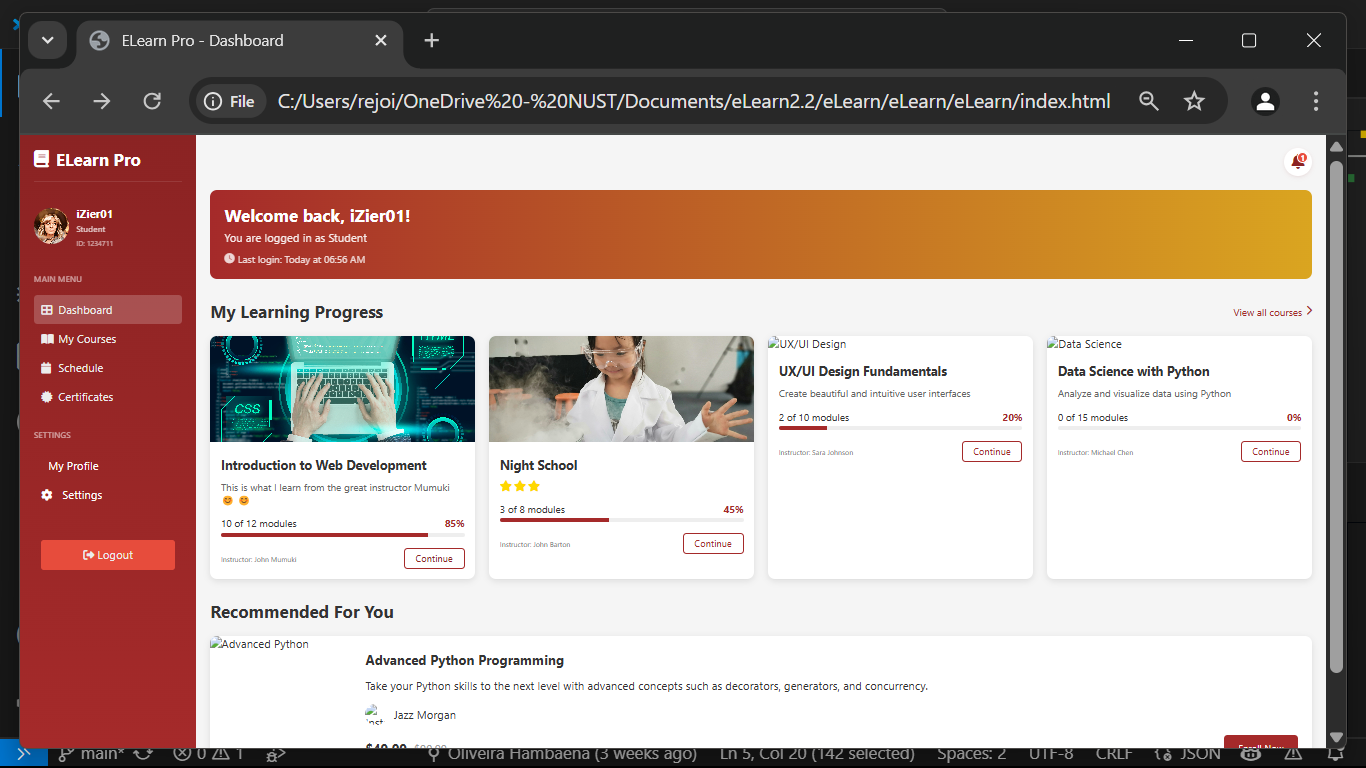


Database

****

**version 1**

****

****

**Version 2**