1. Introduction

[TODO write a preamble
maybe include increased digitalization, overworked teachers, ubiquity of smartphones, short attention spans, briefly pitch learning app as solution]

2. Positioning

2.1 Problem Statement

Problem	Disorganized learning materials and exercises in learning environments Existing classroom learning/exercise structure sometimes demotivating
Affects	Lecturers and students in learning environments
Impact	For lecturers: increased workload (difficulty revising material, manually grading exercises, organizing documents) For students: low learning enthusiasm and ensuing suboptimal performance, difficulty estimating progress
Solution	Customizable, user-friendly gamified learning platform

[TODO elaborate on table info]

2.2 Product Position Statement

Target Demographic(s)	Lecturers and students in learning environments
Target Demand	Learning materials easy to revise and update Engaging learning exercises Track learning progress
Product Category	Gamified learning platform
Benefit	Standardized learning materials (e.g. flashcards, presentations, digital exercises) Gamified learning exercises drive enthusiasm and engagement Progress tracking helps organize study priority, e.g. finding points of struggle
Primary Market Alternative	Kahoot
Primary Differentiation	Gamification scope

[TODO explain theoretical benefit of gamification, compare similarities and differences with Kahoot

Maybe also mention other similar sites/apps e.g. khan academy, duolingo, moodle
Also elaborate on "gamification scope"]

[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]

[Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:]

3. Stakeholder Descriptions

3.1 Stakeholder Summary

3.1.1 Internal Stakeholders

Name	**Description**	**Responsibilities**
Project Lead	Overall project coordinator	Project planning and scheduling Production documentation Production timeline development Rresource and risk management Task allocation Team coordination Stakeholder communication
Software Architect	Technical structure designer	Technical architecture design Technical documentation Technology selection Technical guidance Code standardization and review
Quality Manager	Quality assurance officer	QA strategy coordinator Test planning Bug fixing priority coordinator Performance testing Compliance audit Continuous improvement User feedback
Developers	Development team members	Feature development Coding and implementation Version control Bug hunting and fixing Team cooperation General documentation

[TODO describe goals and roles of dev team]

3.1.2 External Stakeholders

Name	**Description**	**Use-Case**

Lecturers	Instructors in classroom settings and learning environments	Classroom supplement Create/manage repositories Organize lecturing materials Create exercises for students Track student progress
Students	Learners in classroom settings and learning environments	Interaction (use repositories) Utilize lecturing materials Supplement personal notes Complete gamified exercises Build enthusiasm Track/compare own progress
Independent instructors/ Hobbyists	Hobbyists or other independent instructors	Create/manage repositories Documentation Drive interest in a subject Free exchange of information Build independent learning community
Independent learners	Amateurs or other individuals seeking independent learning	Interaction (use repositories) Pursue independent learning Utilize information repositories Build enthusiasm for a topic Compete with other amateurs Track personal progress
Investors	Outside investors financing development	ROI Influence development Support userbase growth

[TODO explain use cases further, explain difference between learning in a learning environment and independent learning/teaching, and why we split them up into two distinct groups]

3.2 User Environment

[TODO elaborate further on previous chapter re:different use cases. Give concrete examples]

[Detail the working environment of the target user. Here are some suggestions: Number of people involved in completing the task? Is this changing? How long is a task cycle? Amount of time spent in each activity? Is this changing? Any unique environmental constraints: mobile, outdoors, in-flight, and so on? Which system platforms are in use today? Future platforms? What other applications are in use? Does your application need to integrate with them? This is where extracts from the Business Model could be included to outline the task and roles involved, and so on.]

4. Product Overview

4.1 Needs and Features

Need	Priority	Features	Planned Release
Top Level Domain	High	TODO	TODO
React.js	High	Frontend	TODO
Node.js	High	Backend	TODO
SQL	High	Database	TODO
GitHub repository	High	Internal Version Control	TODO
TODO (Lua/PHP)?	Low	Additional Scripting	TODO
Mobile Application	Low	TODO	Roadmap
[TODO: maybe think of some more]			

[TODO justify choices, figure out releases]

[Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented. Capture the stakeholder priority and planned release for each feature.]

4.2 Other Product Requirements

Requirement	Priority	Planned Release
Information Security (ISO 27001)	High	TODO
Data Privacy (GDPR, ISO 27701)	High	TODO

Comply with relevant laws and standards (???)	High	TODO
TODO: Find additional requirements	TODO	TODO
.		
.	.	.
.	.	.
.	.	.

[TODO elaborate (see below paragraph)]

[At a high level, list applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements. Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.]

©Nabu UG (2025)