

Quiz Masterz

Von David Levi, Lasse Schulz, Furkan Dinc

Dozent Prof. Dr. Alexander Eck

Datum 03.07.2025

Table of content

01

Problem Definition &
Target Group

02

„Happy Path”

03

Key Design
Decisions

04

Prioritization
Until Submission

1. Problem Definition



Abb. 1

The Problem

Challenges of Traditional Classroom Methods:

- Time-consuming manual grading
- Low student engagement with note-based quizzes
- Lack of gamification to motivate participation
- No instant feedback for learning from mistakes



Our Solution

Quiz Masterz:

New Classroom Learning

For Teachers:

- Manage multiple classrooms
- Create interactive quizzes
- Real-time student analytics
- Automated grading & feedback

Logged in successfully!

Welcome back, Lasse!

Student Dashboard

Join classrooms, take quizzes, and track your learning progress.

My Classrooms

Join Classroom

★ Learning Points
3 LP

1
CLASSROOM(S)

Abb. 3

For Students:

- Gamified learning & rewards
- Track learning progress
- Compete on leaderboards



Abb. 4

Target Users

“Ms. Sarah” - The Modern Teacher

- Age: 32
- Teaches middle/high school subjects
- Values simplicity
- Wants to engage students

“Alex” - The Gamer Student

- Age: 14
- Enjoys interactive learning
- Motivated by competition and achievements
- Prefers immediate feedback over delayed results

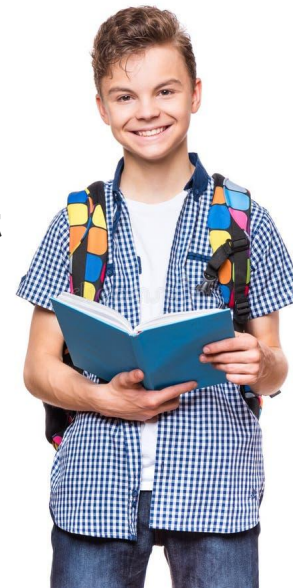
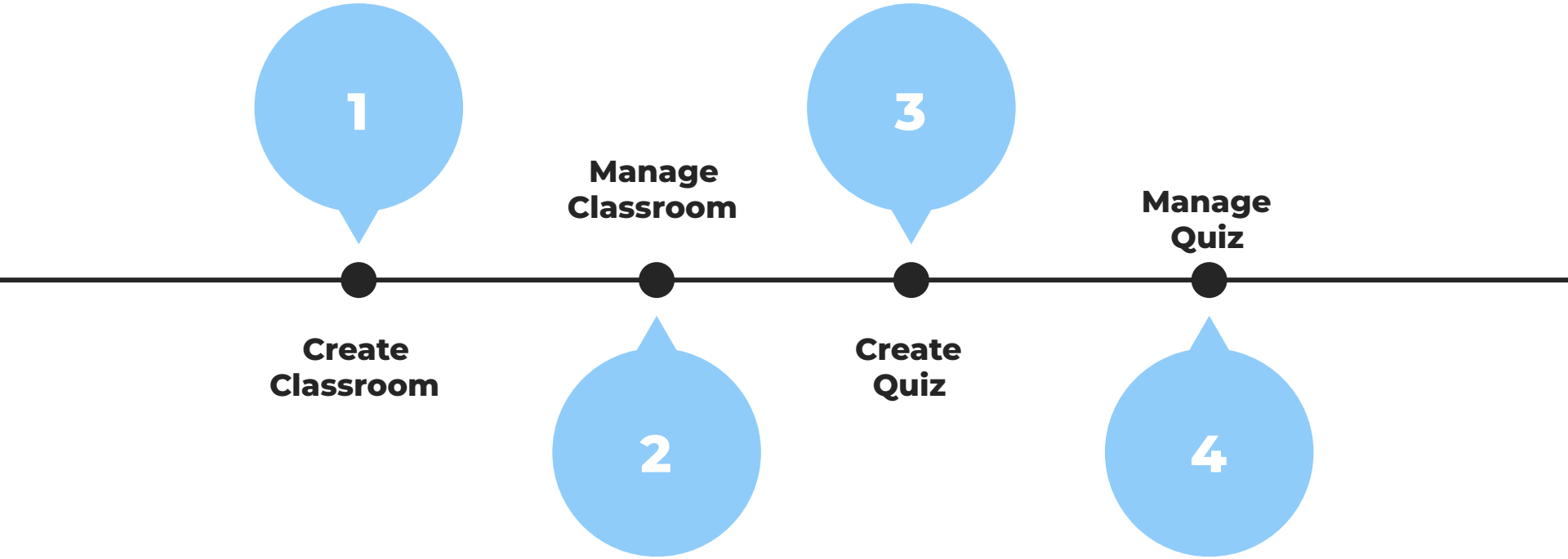


Abb. 5

2. Happy Path: Teacher



✓ Logged in successfully!

Abb.6

Quick Actions

+ New Classroom

Abb.7

+ Create New Classroom

Set up a new classroom for your students

Classroom Name

Classroom Name

Choose a descriptive name for your classroom

+ Create Classroom

← Back to Dashboard

Abb.8

✓ Classroom "Full-Stack Web Development SoSe-2025" created successfully! Join Code: B57B2D

Abb.9

Full-Stack Web Development SoSe-2025

Join Code: B57B2D

Manage

Abb.10

Create Classroom

Full-Stack Web Development SoSe-2025
Join Code: B57B2D

Manage

Abb.10

Quiz Masterz

Home Classrooms Profile Logout

Manage Classroom: Full-Stack Web Development SoSe-2025

Abb.11

Student Leaderboard

Abb.12

Draft Quizzes

Quizzes in progress - not yet visible to students

Abb.13

Full Stack Developer Interview Questions Draft

Questions: 1

✓ Ready to upload

Edit Quiz

Upload Quiz

Delete Quiz

Abb.15

Published Quizzes & Submissions

Quizzes available to students with submission reviews

Abb.16

Full Stack Developer Interview Questions 1 questions 0 submissions

No submissions yet for this quiz

Abb.17

Review Quiz Details

Remove from Classroom

Abb.18

Manage Classroom

Create New Quiz for Classroom: Full-Stack Web Development SoSe-2025

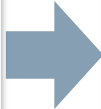
Quiz Name

Full Stack Developer Interview Questions

Create Quiz and Add Questions

Back to Dashboard

Abb.19



Add New Question

Question Text

What is a REST API, and why is it used?

Learning Points (LP)

6

Add Question and Manage Answers

Existing Questions

No questions added to this quiz yet.

Back to Classroom

Abb.20



Question: What is a REST API, and why is it used? (10 LP)

New Answer

Maximum answers reached.

Existing Answers

A REST API (Representational State Transfer Application Programming Interface) is a standardized way for applications to communicate over HTTP by following a set of principles. It allows clients (like web browsers or mobile apps) to interact with servers to perform operations like fetching or modifying data.

a dog

a cat

In a REST API, the HTTP methods (GET, POST, PUT, DELETE) directly map to CRUD operations on a database: GET = SELECT, POST = INSERT, PUT = UPDATE, DELETE = DELETE.

Correct

Incorrect

Incorrect

Incorrect

Back to Questions for: Full Stack Developer Interview Questions

Save Quiz and Go Back to Classrooms

Abb.21

Create Quiz

Happy Path: Student

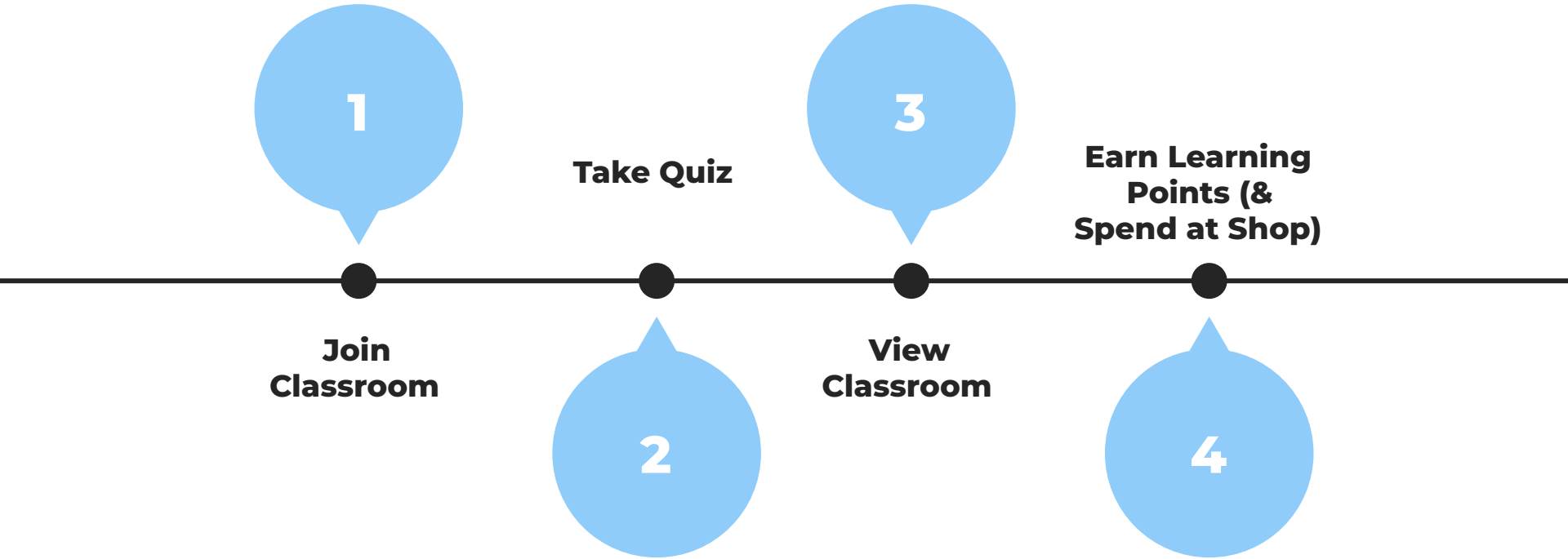




Abb.22



Abb.23

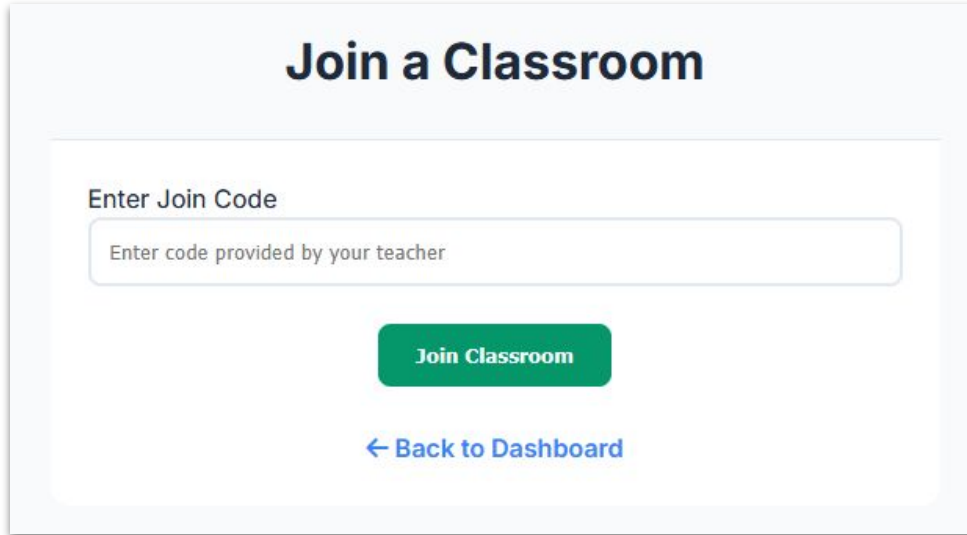


Abb.24

Join Classroom



Abb.25

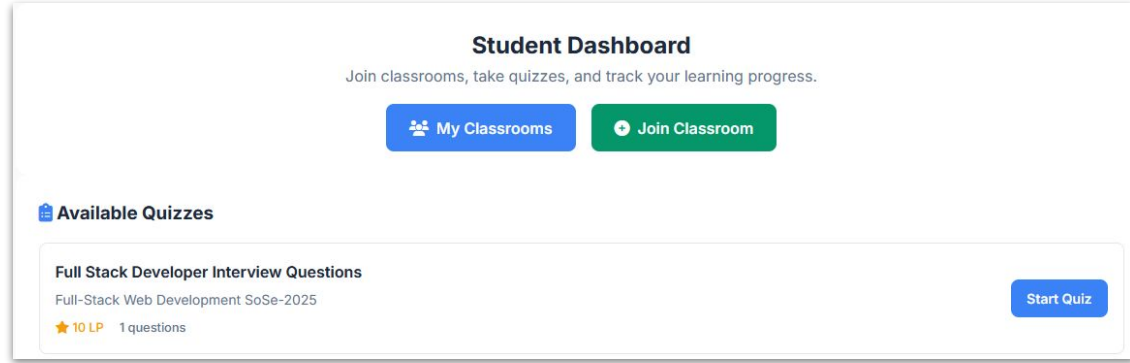


Abb.26



Abb.27

Take Quiz



Abb.28

My Classrooms

View and manage your joined classrooms

History 101

Hans

→ View

Full-Stack Web Development SoSe-2025

Hans

→ View

← Back to Dashboard

Abb.29

Full-Stack Web Development SoSe-2025

Taught by: Hans

Available Quizzes

Full Stack Developer Interview Questions

🕒 Available now

✅ Completed

🔄 Retake Quiz

← Back to Dashboard

🏆 Leaderboard

1



Student 9
Student

10
POINTS

2



Student 2
Student

10
POINTS

View Classroom

Abb.30

Full-Stack Web Development SoSe-2025

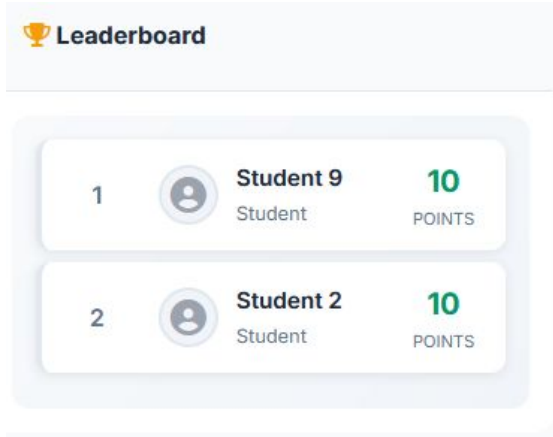


Abb.31

Welcome back, Student 9!



Abb.33

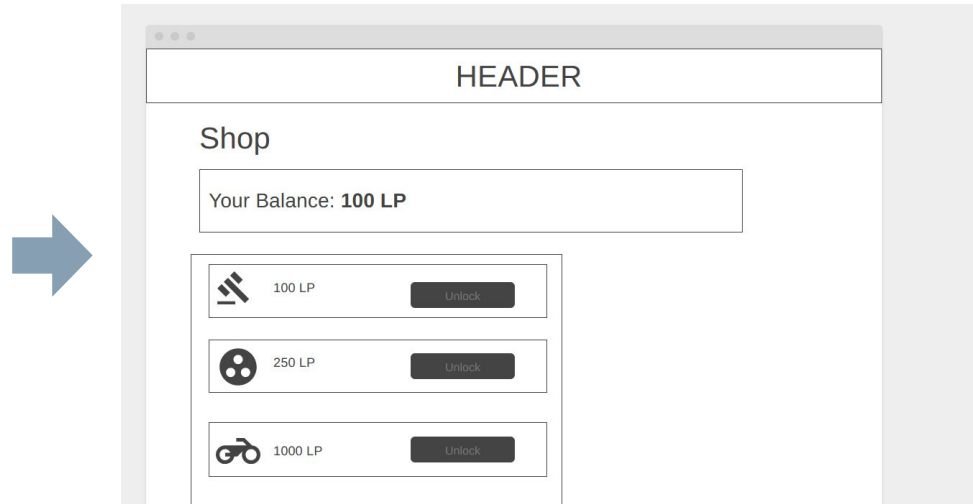
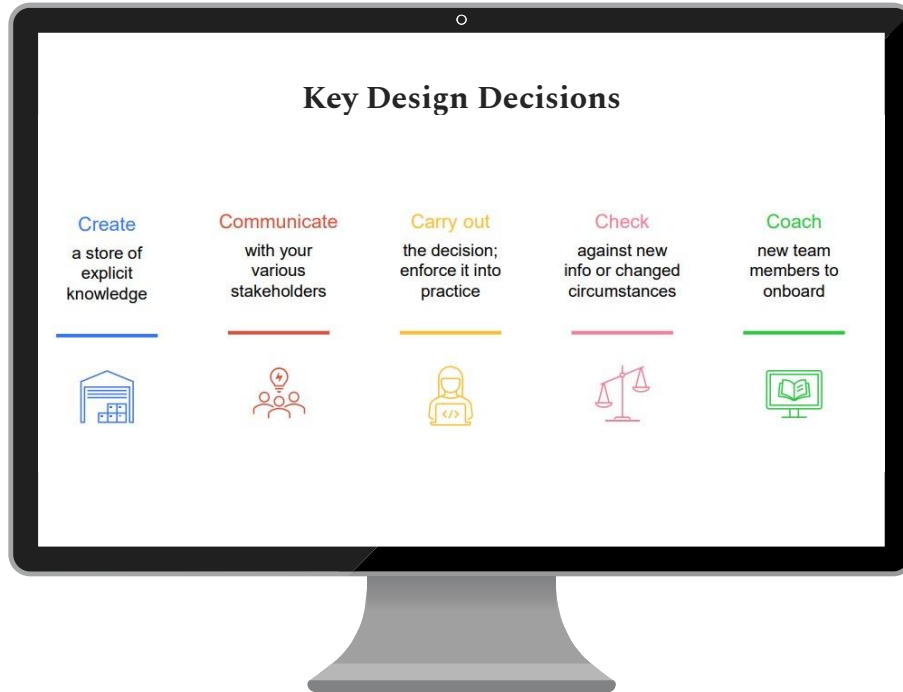


Abb.32

Earn Learning Points



3. Key Design Decision s

Key Design Decisions

Problem:

How do we implement RBAC so teachers manage quizzes and students only access assigned ones?

Decision: Simple role-based system with database-stored roles

Communicate: Role field in User model + route-level guards

Carry out: Abb.33

Key Design Decisions

```

# forms.py
class SignUpForm(FlaskForm):
    role = SelectField('Role', choices=[('student', 'Student'), ('teacher', 'Teacher')],
    validators=[DataRequired()])
# views.py
def require_role(role):
    if current_user.role != role:
        flash('Access denied.', category='error')

@views.route('/student/quizzes')
def student_quizzes():
    if current_user.role != 'student': # ← Route-level guard
        flash('Access denied. Students only.', category='error')
        return redirect(url_for('views.home'))
```

Key Design Decisions

Problem:

How can teachers navigate in the quiz creation process ?

Decision: Implement contextual navigation

Communicate: Teachers need to move quickly between creation levels

Carry out:

```
<button onclick="location.href='{{ url_for('views.teacher_manage_quiz_questions', quiz_id=quiz_id,
classroom_id=classroom_id) }}'" class="btn btn-primary">
  Back to Questions for: {{ question.quiz.name }}
</button>

{% if classroom_id %}
<button onclick="location.href='{{ url_for('views.edit_classroom', classroom_id=classroom_id) }}'"
class="btn btn-success">
  Save Quiz and Go Back to Classroom
</button>
{% else %}
<button onclick="location.href='{{ url_for('views.teacher_classrooms') }}'" class="btn btn-success">
  Go Back to Classrooms
</button>
{% endif %}
```

Key Design Decisions

Problem:

How should form validation be handled ?

Decision: Implement server-side validation using Flask-WTF

Communicate: Shows users form validation errors

Carry out:

```
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, SelectField, SubmitField
from wtforms.validators import DataRequired, Email, Length, EqualTo

class SignUpForm(FlaskForm):
    email = StringField('Email', validators=[DataRequired(), Email()])
    first_name = StringField('First Name', validators=[DataRequired(), Length(min=2, max=150)])

    password1 = PasswordField('Password', validators=[DataRequired(), Length(min=7)])
    password2 = PasswordField('Confirm Password', validators=[DataRequired(), EqualTo('password1')])
    role = SelectField('Role', choices=[('student', 'Student'), ('teacher', 'Teacher')], validators=[DataRequired()])
```

4. Prioritization

Until

Submission

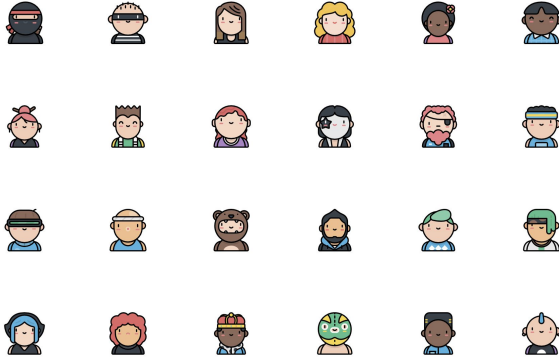
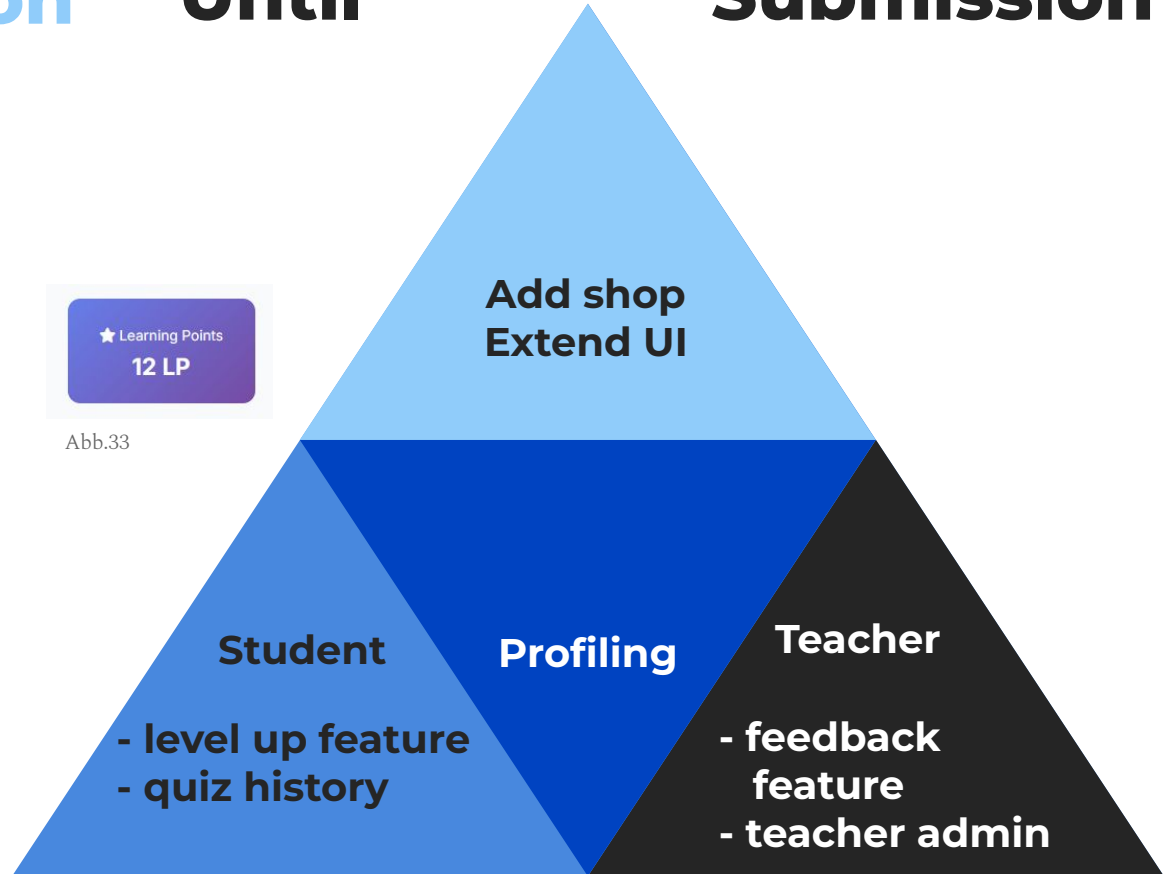


Abb.36



Abb.33



Quellen

Abb.1:

<https://specials-images.forbesimg.com/imageserve/61bd343aec04efd3b8fba2a/problem-solving--solution--systems-thinking--light-bulb/960x0.jpg?cropX1=0&cropX2=2347&cropY1=115&cropY2=1435>

Abb.2:

<https://i.guim.co.uk/img/static/sys-images/Guardian/Pix/pictures/2014/10/30/1414700714198/Praising-pupils-can-be-ha-014.jpg?width=1900&dpr=1&s=none&crop=none>

Abb.3:

Screenshot from Quiz Masterz Student Homescreen

Abb.4:

<https://stock.adobe.com/de/images/pretty-teacher-smiling-at-camera-at-back-of-classroom/202523652>

Abb.5:

<https://www.dreamstime.com/portrait-young-student-school-bag-books-teenager-smiling-looking-camera-happy-teen-boy-isolated-white-image112010632>

Abb.6-35:

Screenshots from the Quiz Masterz Web App

Abb 36:

<https://www.flaticon.com/packs/kawaii-avatars-3>



Source Code