MindQuestLeveling Handbuch

Web- und Multimediatechnologien

Patryk Widulinski, 10016213

Mateusz Goralewski, 10016327

SoSe2024, Projektarbeit (Alternative 3)

Inhaltsverzeichnis

Technische Dokumentation

Entwicklungsumgebung einrichten

Voraussetzung

Als die Voraussetzung schauen wir uns das Betriebssystem und den “Startpunkt” an. Für unseren Fall, haben wir eine Variante für Windows, und eine Variante für Linux eingebaut, da wir für die Entwicklung Windows verwendet haben und für das Deployment einen Linux Debian 12 Server verwenden (In unseren Fall ist es ein Proxmox Container, aber dies sollte keine Rolle spielen)

Die Anleitung für MacOs sollte nicht viel anders sein als die für Windows. für diesen Fall sollte man entsprechende Inhalte online auffinden.

Windows
Installation von Node.js und NPM auf Windows 11
Node.js Installer herunterladen: Besuche die offizielle Node.js Webseite https://nodejs.org/ und lade den Installer für Windows herunter.
Node.js Installer ausführen: Führe die heruntergeladene Datei aus und folge den Anweisungen im Installationsassistenten. Achte darauf, dass die Option npm package manager aktiviert ist.
Installation überprüfen: Öffne die Eingabeaufforderung (Command Prompt) und führe die folgenden Befehle aus:

```
node -v
npm -v
```

Diese Befehle zeigen dir die installierten Versionen von Node.js und NPM an.

Linux (Debian 12)
Installation von Node.js und NPM auf Debian 12
Terminal öffnen: Öffne ein Terminalfenster auf deinem Debian 12 System.
Vorbereitungen - Paketlisten aktualisieren:

```
sudo apt update
sudo apt upgrade
```

Node.js und NPM installieren:

```
sudo apt install -y nodejs
```

Installation überprüfen:

```
node -v
npm -v
```

Diese Befehle zeigen dir die installierten Versionen von Node.js und NPM an.

Wahrscheinlich kommt es dazu, dass die Versionen veraltet sind. Diese können wir wie folgt updaten:

```
npm install -g npm@latest
```

MongoDB

Für die Installation von MongoDB verweisen wir auf die offizielle Docs von MongoDB selbst, da diese sehr gut und einfach den Installationsprozess auf allen Betriebssystemen erklärt.

https://www.mongodb.com/docs/manual/administration/install-community/

Nach der Erfolgreichen Installation, müssen wir noch ein paar Schritte verfolgen, damit die Datenbank für die Weiterverwendung bereit ist.

Unser Backend Server, verbindet sich mit der Datenbank über den Standardlink welcher richtig ist, solange sich die Datenbank auf dem selben Server befindet wie das Backend selbst.

Wichtig zu beachten ist, dass wir eine Instanz der Datenbank MindQuestLeveling erstellen müssen, in die eine Collection users kommt.

Für den weiteren Verlauf werden wir Beispielsweise die Commands für Linux Betriebssysteme zeigen.

Zuerst müssen wir Sicherstellen, dass unsere Datenbank läuft.

```
sudo systemctl start mongod
```

Als nächstes, verbinden wir uns mit der Datenbank Anwendung.

```
mongosh
```

Jetzt erstellen wir vorerst eine temporäre Datenbank mit dem Namen MindQuestLeveling. Damit die App funktioniert, darf der Name der Datenbank nicht verändert werden, oder eventuell entsprechend Angepasst werden.

```
use MindQuestLeveling
```

Jetzt machen wir die Datenbank Persistent und erstellen die entsprechende Collection users in die wir einen User einfügen.

```
db.users.insertOne({name: "testUser"})
```

die Kommandozeile darf geschlossen werden, oder mit dem command `exit` verlassen werden.

Damit wären wir mit dem Setup der Datenbank fertig.

Für Die Installation auf Windows empfehlen wir die Nutzung von MongoDBCompass, welcher sehr einfach ermöglicht sich mit der mongosh console zu verbinden und direkt mit der Datenbank mitinstalliert wird!

Build und Deployment Information

Um Best-Practice-Methoden zu folgen, die Zugänglichkeit zu enkapsulieren und den Datenverkehr zu optimieren, sollte die MongoDB-Datenbank auf demselben Server wie die App (das Backend) laufen. Die Verbindung zur Datenbank erfolgt über localhost und den Standardport.

In unserem Fall, wird der localhost als eine numerische IP angegeben, um Probleme mit IPv6 zu beheben. `mongodb://127.0.0.1:27017`

- Der Connectionstring ist in `server/.env` aufzufinden. Es ist bereits der default Port von MongoDB und localhost eingetragen

Sollte der Backend auf einem separaten Server als das Frontend laufen, muss auch der apiUrl link angepasst werden. In unserem Fall, ist es `http://localhost:3000/api`

- Die API Url ist in `src/app/environment.ts` aufzufinden. Es ist bereits der default Port von Nodejs Server und localhost eingetragen

Frontend-Setup

Angular CLI installieren

```
npm install -g @angular/cli
```

Abhängigkeiten installieren: (Innerhalb des Ordners des Projekts, im `mind-quest-leveling/` verzeichnis)

```
npm install
```

Angular Projekt bauen: (Innerhalb des Ordners des Projekts, im `mind-quest-leveling/` verzeichnis)

```
ng build
```

Backend-Setup

Abhängigkeiten installieren: (Innerhalb des Ordners der App, im Ordner: `/mind-quest-leveling/server/`

```
cd ./server/
npm install
```

Server starten (Innerhalb des Ordners der App, im Ordner: `/mind-quest-leveling/server/`)

```
node server.js
```

Der Server läuft auf dem standardmäßigen Port In unserem Fall: `http://localhost:3000/`

Ergänzung weiterer Elemente

Items & Rarity Types

Die Itemliste ist in der Datei `item.service.ts` aufzufinden. Sie wird clientseitig zur Verfügung gestellt. Jedes Item besteht aus einer einzigartigen ID, die fortlaufend ergänzt werden sollte. Danach folgen ein Name und eine Beschreibung, die frei gewählt werden können.

Nach der Einfügung neuer Items in die Liste muss noch ein Eintrag in die styles.css erfolgen, welcher das Bild des Items definiert. Dieser Ansatz ermöglicht es auch, potenziell einfach ein weiteres Styling für jedes Item einzufügen.

Für jedes Item muss auch ein Bild gewählt werden. Um ein neues Bild hinzuzufügen, kann der Pfad `/assets/img/items/[bildname].[format]` gewählt werden. Die Verwendung von GIFs sollte hierbei ebenfalls keine Probleme darstellen.

Quests

Die Quests sind in `/assets/jsonDatabase/` aufzufinden. Jede der dort befindlichen Dateien ist als .json-Datei eingebunden und hat einen unterschiedlichen Aufbau. Die Datei classQuests.json enthält zuerst Klasseneinträge, welche dann alle Quests für die jeweilige Klasse beinhalten.

Jede der Quests in allen Dateien hat den gleichen Aufbau. Sie beginnt mit einer einzigartigen ID, die als String angegeben wird. Die ID sollte immer einzigartig bleiben und folgt einem spezifischen Muster, das beim ersten Anblick selbsterklärend ist. Um sicherzustellen, dass die ID immer einzigartig bleibt, startet die ID mit einem Präfix der jeweiligen Kategorie (zum Beispiel: Klasse Magier startet mit "mag"), danach folgt ein Unterstrich und eine Zahl für die Nummer der Quest. Hier kann die ID beliebig angepasst werden; wichtig ist allerdings, dass sie immer einzigartig bleibt. Wir empfehlen auch, das Muster beizubehalten, um die Sache einfacher zu machen.

Nach der ID folgen Name, Beschreibung, XP und Zeit. Die Zeit sollte immer numerisch bleiben und wird in Stunden angegeben, da wir in unserer App davon ausgehen, dass jede der Quests eigentlich länger sein sollte. Es ist jedoch möglich, eine Dezimalzahl anzugeben und somit unter einer Stunde zu bleiben. Für diesen Fall haben wir das Frontend so angepasst, dass der Nutzer den Fortschritt seiner Quest als Minutenzahl sieht, damit es intuitiver ist und auch kürzere Quests kein Problem darstellen.

Die XP sollte ebenfalls numerisch bleiben. Hierbei ist wichtig zu beachten, dass jeder Level 100 XP entspricht. Es sollten keine Dezimalzahlen verwendet werden, da das System dafür

nicht vorbereitet wurde. Ein Problem sollte es trotzdem nicht darstellen, wir empfehlen es allerdings definitiv nicht.

Name und Beschreibung können hierbei frei und beliebig lang als String angegeben werden.

Benutzerhandbuch

Benutzeroberfläche und Funktionen

Einführung

Zuerst wird man auf die Landingpage geleitet, die alle Grundinformationen über die App darstellt.

Navigation Landingpage

In der oberen Navigationsleiste findet man zwei Buttons: einen zum Login und einen zur Registrierung.

Registrierung

Wenn man sich ein Konto erstellen möchte, sollte man sich registrieren.

Bei der Registrierung muss ein Nutzername angegeben werden (dieser wird später nicht mehr benötigt, sondern dient nur zur Identifikation).

Danach gibt man die E-Mail-Adresse an, die als Login dient.

Dann folgt das Passwort, das zweimal eingegeben werden muss und folgende Kriterien erfüllen muss:

- Eine Mindestlänge von 8 Zeichen
- Mindestens eine Zahl

Nachdem man dies erfüllt hat, wird man automatisch zum Login weitergeleitet, wo man seine Daten eingeben muss.

Interview

Danach folgt ein „Interview“, bei dem man zu verschiedenen Aspekten befragt wird. Die Antworten beeinflussen die zugewiesene Klasse, die man später jederzeit ändern kann, falls sie nicht passend ist.

Hauptnavigation

Der wichtigste Aspekt der Applikation ist nun die untere Navigationsleiste (Navbar), die die Navigation zu den wichtigsten Elementen ermöglicht.

Dort sind drei Buttons zu finden:

1. Einer, der zum Profil weiterleitet, wo man seinen Charakter anpassen kann.
2. Einer in der Mitte für das Dashboard, wo man weitere Elemente der Applikation findet, wie die Questliste, aktuelle Quests, Informationen oder die Aufgabenliste.
3. Ein dritter Button für das Inventar, in dem alle Items aufgelistet sind.

Erste Schritte

Der Benutzer bekommt am Anfang 5 kostenlose Kisten, um zu testen, wie dies funktioniert und um motiviert zu werden, eine Quest zu beginnen.

Questliste

In der Questliste gibt es mehrere Abschnitte:

- Einen für klassenspezifische Quests
- Einen für allgemeine Quests für alle
- Einen Abschnitt für Event-Quests, die zum Beispiel zum Testen dienen

Belohnungen

Nach erfolgreichem Abschluss einer Quest (die im realen Leben absolviert werden sollte) bekommt man eine Kiste, die geöffnet werden kann.

Die Kiste wurde mit ThreeJs integriert, was eine 3D-Ansicht ermöglicht, sowie die Drehung in allen Achsen. Nachdem man den Button auf der Kistenseite betätigt hat, wird die Kiste eingefroren und eine Animation abgespielt. Nach Abschluss der Animation wird man auf eine neue Seite weitergeleitet, wo man das Item sieht, das man erhalten hat.

Dieses Item kann ebenfalls gedreht werden, allerdings nur in zwei Richtungen, da es sich um ein 2D-Objekt handelt.

Von dort aus kann man entweder zum Inventar zurückkehren oder zum Dashboard, um weitere Aktivitäten zu unternehmen.

Aufgabenliste

Ein weiterer wichtiger Aspekt ist die Aufgabenliste, die durch iframes verschiedene Minigames integriert, um unsere Zeit produktiv zu gestalten.

Die Aufgabenliste wurde hauptsächlich für die Verwendung auf Desktop-Computern entwickelt, sollte jedoch auch auf anderen Geräten problemlos funktionieren.