Informatik Website Projekt 24/25

Von Rushi und Claas

Vorwort:

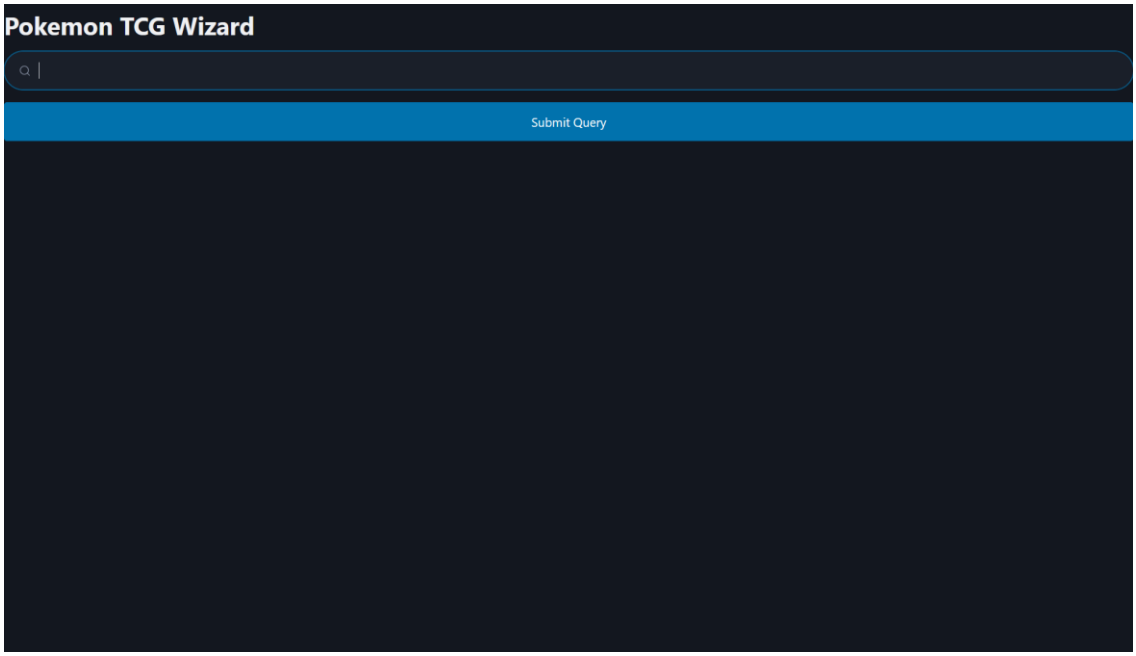
Dies ist ein Teilergebnis einer mehrwöchigen Arbeit von Rushi Ram Panitini und Claas Mönnigmann. Im Folgenden werden Planung, Vorgehen, Vorgehenserläuterung und Vertiefungserläuterung erklärt. Es ist zu beachten, dass diese Ergebnisse aus einem kurzen Zeitraum sind, da wir wegen mehreren Komplikationen erst zwei Wochen vor den Weihnachtsferien mit unserem Projekt anfangen konnten, deswegen haben wir auch nur diesen Zeitraum in unserer Protokollierung einbezogen.

Plan:

Unser Ziel war es eine Website zu kreieren, welche sowohl visuell ansprechend als auch funktional und praktisch ist. Vorerst war uns noch nicht klar in welchem Themengebiet wir uns bewegen, um dieses zu minimieren haben wir uns also auf unsere grundlegenden Vorstellungen einer guten Website konzentriert. Es war uns beiden wichtig, dass die Website auf allen Gerätetypen aufrufbar ist, eine große Menschengruppe anspricht, detailliert, aber nicht zu komplex wird und möglichst schnell ist. Dann kam der Vorschlag eine Pokémon Website zu erstellen. Die finale Idee war eine Website, wo der Nutzer nach einem Pokémon sucht und die Karten des Pokémon angezeigt werden.

Protokoll:

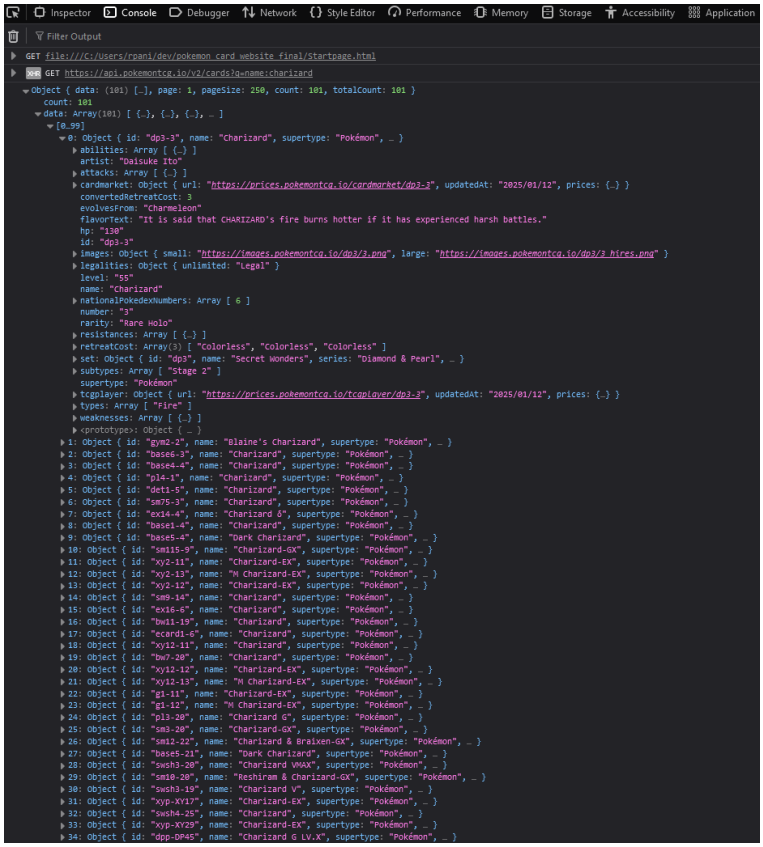
Freitag 06.12.24:



Als Erstes haben wir mit der CSS-Framework pico.css eine eher einfache HTML-Webseite erstellt. Diese HTML-Webseite hat eine Überschrift, eine Suchleiste und ein Suchknopf der den Input in der Suchleiste an eine Javascript-Funktion mit dem Namen returnname() weitergibt.

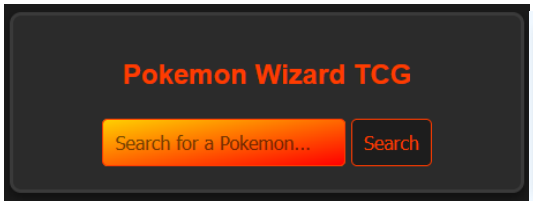
Montag 09.12.24:

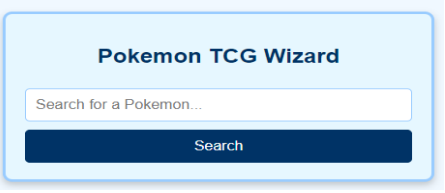
Am 09.12. haben wir mit der Programmierung des Javascript der Webseite angefangen. Die Funktion returnname() stellt mithilfe des Inputs von der Suchleiste einen GET-Request an eine API. Die API antwortet auf den GET-Request mit einer JSON-Datei die eine Liste aller Karten beinhaltet, die den gleichen Namen wie den Input in der Suchleiste haben. Diese JSON-Datei hat fast alle Informationen über die Karten wie z.B. Name der Karte, Attacken der Karte, das Preis der Karte usw. Außerdem haben wir auch damit begonnen unsere Website zu verschönern. Vorerst haben wir uns mit verschiedenen Farbkombinationen und Layouts auseinandergesetzt s.u.

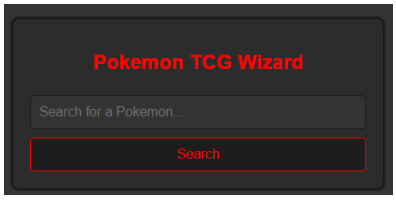


Freitag 13.12.24:

An diesem Freitag haben wir uns hauptsächlich auf das Aussehen der Pokémon-Website fokussiert. Wir haben einen Namen festgelegt und uns dafür entschieden, dass wir Pico.CSS nicht mehr verwenden wollen und uns auf normales CSS konzentrieren wollen. Darüber hinaus haben wir mit der Planung für die nächste Stunde begonnen, da wir die Effizienz unserer Arbeit steigern wollten.





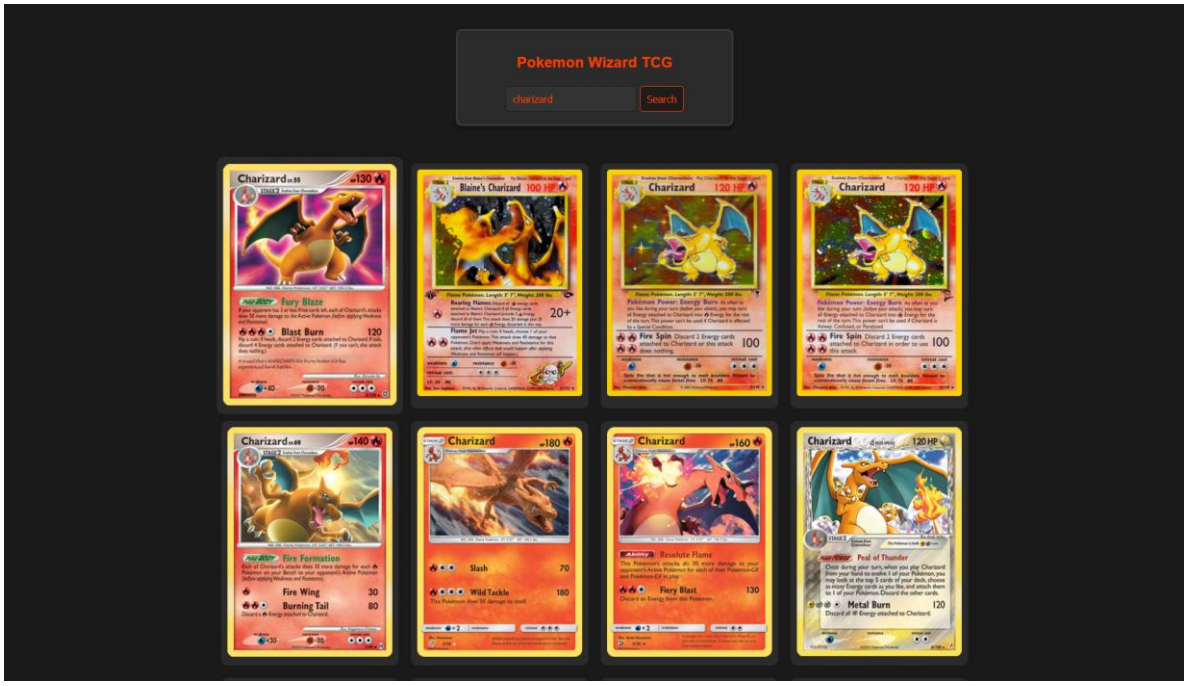


Montag 16.12.24:

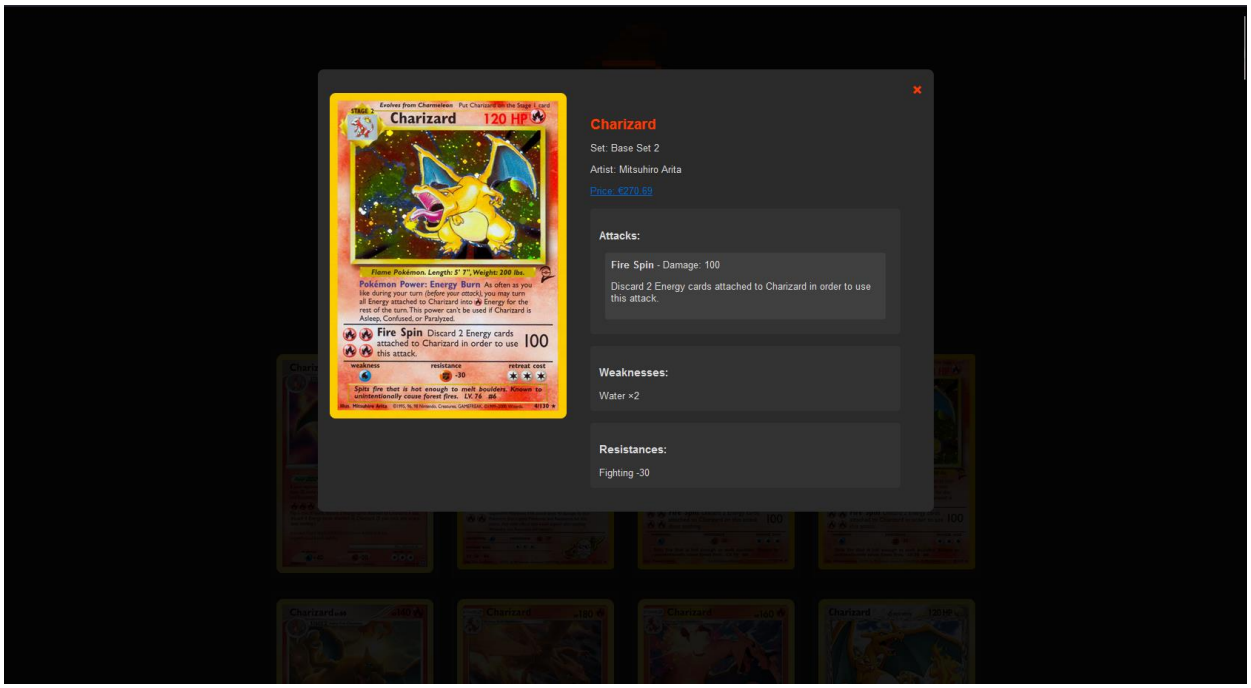
In der letzten Doppelstunde vor den Ferien haben wir uns wie in der letzten Stunde mit dem Aussehen und dem ersten Kontakt des Nutzers mit der Website auseinandergesetzt. Wir dachten an ein Maskottchen, welches das erste ist, was dem Nutzer ins Auge fällt. Dabei dachten wir an ein Pokémon, was passen zu unserem Namen einen Zauberhut trägt. Das Pokémon war letztendlich auch ausschlaggebend für die Farbauswahl unserer Website und mit einer Starterentwicklung wie Glurak die jeder kennt, konnten wir nichts falsch machen. Somit mussten wir uns entscheiden, wie wir den Glurak auf der Website darstellen, da wir so viel wie möglich von Uns stammen sollte haben wir uns für ein Pixelart etschieden.

In den Ferien (Vertiefung):

In den Ferien haben wir mit dem JavaScript und Design weitergemacht. Zuerst haben wir einen div-Element für die Bilder der Karten erstellt. Wir haben dann die returnname() Funktion erweitert. Da wir wollten, dass die Karten tabellenartig angezeigt werden, haben wir dafür ein neues div-Element mit dem Namen “card-grids” erstellt und in CSS mithilfe des “display” Attributes die Karten tabellenartig dargestellt. Dann habe ich einen map() loop erstellt, der jede Karte nacheinander herunterlädt, anzeigt und dem div Element “card-item” beifügt. So habe ich viele Möglichkeiten, die angezeigten Karten besser dazustellen. So habe ich mithilfe der card-item class eine Animation mithilfe der transform() Funktion in CSS erstellt und eine kleine dunkelgraue Umrandung um die Karten herum erstellt.

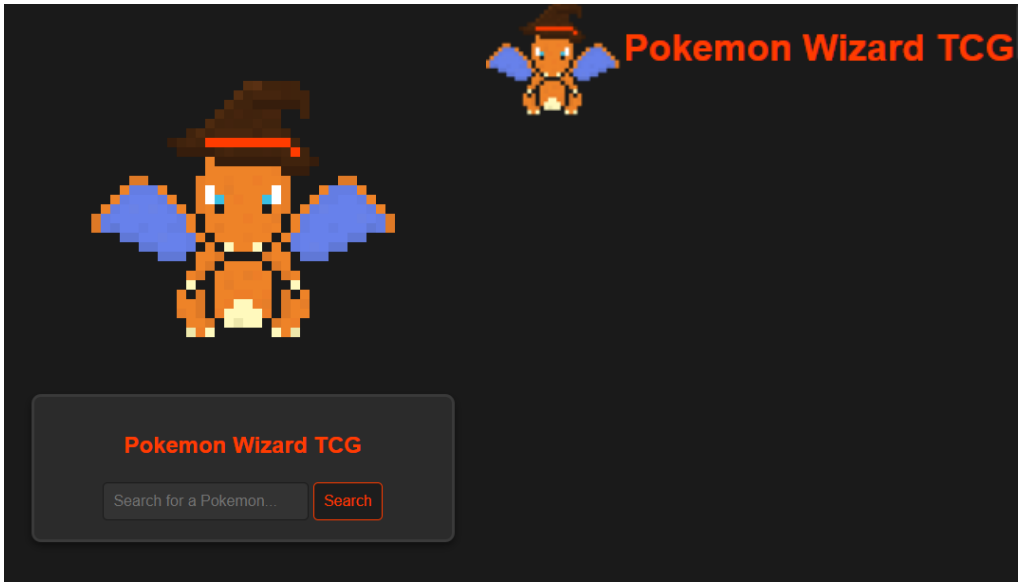


Als Nächstes wollten wir verschiedene Informationen über die Karten visuell ansprechend anzeigen. Zuerst wollten wir für jede Karte eine Info-Seite generieren lassen, wenn auf die Karte draufgeklickt wird. Wir haben uns aber für Modals entschieden, also ein kleines Fenster vor den Karten.



Um den Modal zu generieren, habe ich die Funktion showCardDetails() erstellt, die eine Variable card braucht. Diese Funktion wird ausgelöst, wenn auf einer Karte draufgeklickt wird. Diese Funktion beinhaltet eine in HTML programmiertes Modal. Damit aber die Informationen im Modal sich je nach Karte ändern, haben wir den card Object benutzt. Dies wird von der API bereitgestellt und hat alle Informationen über eine Karte. Wenn man z.B. wissen will, wie eine Karte heißt, benutzt man card.name. Danach haben wir dem HTML Modal einen div Element zugewiesen mithilfe von modal.innerHTML. Daraufhin haben wir den Knopf programmiert, womit man den Modal schließen kann. Damit der Modal auch vor den Karten angezeigt wird, mussten wir in CSS bestimmte Attribute ändern. Wir haben den Background schwarz gemacht damit es aussieht, als ob der Modal vor den Karten liegt und den z-index auf 1 gestellt. Damit die Informationen und Karten visuell ansprechend angezeigt werden, haben wir beim Modal den overflow Attribut zu auto gestellt, den border-radius Attribut verändert damit wir überall abgerundete Ecken haben, und Abstände zwischen Karten und Informationen im Modal mit den Attributen margin und padding gehalten. Zum Schluss haben wir mithilfe der @media Regel in CSS Responsive Webdesign hinzugefügt, so dass z.B.,wenn der Bildschirm des Benutzers kleiner als 900 px werden nur 2 Karten pro Reihe angezeigt.

Darüber hinaus haben wir in den Ferien unser Maskottchen erstellt und die Farben angepasst. Zuerst waren wir uns nicht sicher wie oder wo wir es auf der Website zeigen sollten. Wir hatten zwei Ideen einmal konnte es klein oben in der linken Ecke abgebildet sein oder groß in der Mitte über der Suche.



Letztendlich haben wir uns für die größere Version entschieden (linkes Bild). Zudem haben wir noch ein GIF hinzugefügt welches gezeigt wird, wenn die Maus des Nutzers über dem Glurak Bild ist.

Quellen/Tools:

Visual Studio Code -> Code Editor

Pixilart -> Pixel Programm

Pixelspeechbubble -> Sprechblasen Generator

Chatgpt -> Farben Harmonisierung und Boxshadow

W3Schools -> Javascript Modals, fetch() und map()