

# Projektbeschreibung Fachinformatiker-Anwendungsentwicklung | Modul Javascript | FI-AE 2023

## Vorbemerkungen:

Bearbeitungszeit:	29. Januar bis 2. Februar 2024
Erlaubte Hilfsmittel:	uneingeschränkter Zugang zu allen Ressourcen
Abgabe:	2. Februar 2024, 15:30 Uhr (MEZ)

## Framework

Erstellen sie ein Portfolio entsprechend nachfolgender Kriterien. Coden sie in Javascript und verwenden sie für Variablen, Konstanten, Bezeichner weitgehend die Camel-Notation. Verwenden sie den ECMA-Standard und beziehen sie sich auf das World-Wide-Web-Consortium ([w3c](https://www.w3c.org/)). Die digitale Bezeichnung der Projektmappe sollte der Form myProjekt.[VornameNachname] erfolgen. Lesen sie die Anweisungen sorgfältig durch und beachten sie diese. Die Ausnahme bestätigt die Regel – nicht umgekehrt. Die nachfolgende Sitemap (Abbildung 1) ist die Vorlage, die sowohl inhaltlich als auch strukturiert umgesetzt werden soll.

## SiteMap

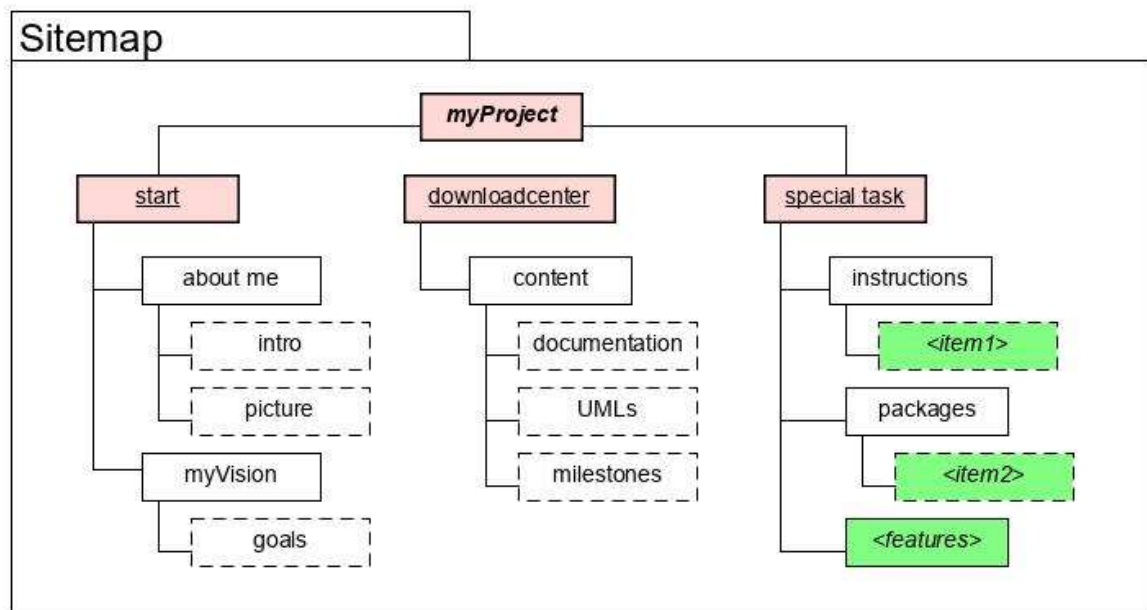


Abbildung 1.1

### about me:start

Der Inhalt **intro** bezieht sich auf die Vorstellung ihrer Person und den wichtigen persönlichen Daten [Vorname, Name, Alter respektive Geburtsdatum]. Außerdem sollte die Vorstellung Hinweise darauf geben, die im Kontext auf

- die aktuelle Ausbildung bei der CBM
- die Motivation zu diesem Beruf
- die relevanten fachbezogenen Erfahrungen
- als auch mögliches gesellschaftliches Engagement reflektieren

Hinweis: Die Items dürfen in einem Absatz oder differenziert beschrieben werden.

Der Inhalt **pictures** ist eine Bildergalerie, bestehend aus mindestens 3 Bildern, wobei ein Bild ihr Portrait zeigen sollte. Das Portraitbild ist kein zwingendes Merkmal. Beachten sie das Urheberrecht und die Quellenangabe. Bevorzugen sie OER-Materialien. Verwenden sie für die Elemente Bezeichnungen, die selbsterklärend sind und der Konvention entsprechen. Bei zusammengesetzten Begriffen ist die Camel-Notation anzuwenden.

### myVision:start

Dieser Inhalt bezieht sich auf ihre individuellen Ziele (**goals**) und Vorstellungen. So beschreiben sie hier

- ihre Hard- und Softskills
- ihre beruflichen Ziele
- die Kompetenzen, die sie zukünftig erwerben möchten
- und die Identifikation mit dem Berufsbild Development

### content:downloadcenter

**documentation** bedeutet die Bereitstellung ihrer verwendbaren Dokumente, Erläuterungen, Diagramme und Materialien als zusammengefasster Bericht im Format \*.pdf (Portable Document Format). Das Dokument ist mit einem Deckblatt und einem Inhaltsverzeichnis zu vervollständigen. Die Sitemap ist entsprechend [packages:special tasks](#) zu ergänzen. Die Dokumentation ist als Ergänzung zu ihrem Code zu verstehen und sollte ein schmales Format besitzen. Zur Orientierung sind nachfolgende Schwerpunkte gegeben, welche die Herangehensweise erleichtern soll. Grundsätzlich geben sie hier Hinweise zu komplexen Strukturen und stellen das gewählte Diagramm vor. Für die Bearbeitung des Projektes stellt dieser Punkt nur eine untergeordnete Funktion dar.

	Deckblatt
	Inhaltsverzeichnis
1.	Projektbeschreibung
1.1	Motivation zur Wahl der Aufgabe (0,5 .. 1 Seite)
2.	Coding
2.1	Allgemeiner Aufbau des Programms (1 Seite)
2.2	API - Connections (0,5 .. 1 Seite)
2.2	Erläuterungen zu der gewählten Notation (0,5 .. 1 Seite)
3.	Fazit
3.1	Zusammenfassung (0,5 Seite)
4.	Quellenverzeichnis
5.	Anlagen
	Diagramme
	vervollständigtes Sitemap

Abbildung 2 | Beispiel

**UML-**notierte [Diagramme](#) sind vorzugsweise in Umllet respektive [Umletino](#) zu erstellen. Platzieren sie die Diagramme als Miniaturansicht auf ihrem html-Dokument nebeneinander. Konvertieren sie die Bilder in ein entsprechendes Dateiformat. Programmieren Sie ein günstiges Mausereignis, welches die Ansicht des betroffenen Bildes so vergrößert, dass der Betrachter die entscheidenden Details erkennen kann.

**milestones** ist eine Portable-Network-Grafik, die am Ende des Projekts vervollständigt wird und beschreibt, welche Inhalte im Zeitraum von 5 Arbeitstagen bewältigt wurden. Erstellen sie einen Zeitstrahl, der chronologisch die wichtigen Erfolgspunkte zeigt und binden sie diesen in das Dokument downloadcenter ein.

### instruction:special tasks

Der Content gibt Auskunft zu ihrem Projekt, mögliche Anweisungen für den Nutzer, Spielregeln und dergleichen, um einen zweifelsfreien Umgang mit dem Ergebnis zu gewährleisten. Die Veröffentlichung der Aufgabenstellung ist optional.

### packages:special tasks

impliziert die Elemente <features> und <item x> der [Sitemap](#). Hier wird das individuelle Projekt aufgerufen. Alle benutzerdefinierten Seiten, Grafiken, built-in-Elemente und weitere Bausteine werden hier strukturiert eingepflegt. Visualisieren sie die Struktur ihrer Projektaufgabe, indem sie die Sitemap mit Umllet (Umlertino) vervollständigen und als Anlage ihrer Dokumentation anhängen. Die Sitemap ist einfach zu strukturieren. Die Übersicht soll die Dokumentnamen und wichtige Referenzen zeigen, die im Projekt verarbeitet bzw. erstellt wurden. Weiterführende Elemente wie, Bildbeschreibung, Hyperlinks und dergleichen bleiben unberücksichtigt.

### Deckblatt documentation kontextbezogene Vorlage

Schriftart, Schriftgrößen und Zeilenabstand sind frei wählbar. Insgesamt gelten allgemeine, formale Regeln. Informieren sie sich bei <https://www.1a-studi.de/bachelorarbeit/vorlage> unter anderem. Der Dokumentation ist zwingend ein Inhaltsverzeichnis einzupflegen, um die Recherche und den Umgang ihrer Ausführungen besser folgen zu können. Fügen sie ihrer Dokumentation ein Quellenverzeichnis bei, dass alle verwendeten Ressourcen, einschließlich AI-basierten Tools erwähnt. Das Verzeichnis hat der Norm DIN ISO 690 zu entsprechen.

### Abgabe

Zur Bearbeitung des Projektes stehen ihnen 5 Arbeitstage zur Verfügung. Die Abgabe hat in digitaler Form zu erfolgen. Das Projektverzeichnis ist exakt der vorgeschriebenen Bezeichnung der Projektmappe zu beschriften und mit allen Materialien in einem Archiv zu komprimieren. Die Abgabe einer selbstentpackenden exe-Datei ist nicht erlaubt.

Die Abgabefrist endet am Freitag, dem **2. Februar 2024 um 15:30 Uhr MEZ**.

### Projektaufgaben

Insgesamt gibt es 3 verschiedene, voneinander unabhängige Projektaufgaben. Diese entsprechen hinsichtlich ihrer Schwierigkeit Level 1, 2 und 3, wobei Level 1 positiv, Level 2 komperativ und Level 3 superlativ zu betrachten ist. Entscheiden sie sich für eine der Aufgaben als Pflichtaufgabe. Optionale Entscheidungen für die zusätzliche Bearbeitung der anderen Aufgaben dürfen nicht getroffen werden. Die Wahl des Levels hat keinen Einfluss auf die Benotung. Streben sie eine souveräne und strukturierte Form der Bearbeitung an, um einen fehlerfreien Code zu erzielen. Beachten sie die zusätzlichen Anweisungen und das Bewertungsschema und halten sie sich an die Vorgaben.

### Diagramme

dem Projekt ist mindestens ein Diagramm beizufügen, dass direkten Bezug zu ihrem Code aufweist. Eine objektorientierte Betrachtungsweise wird empfohlen, jedoch nicht vorausgesetzt. Die Umsetzung des Javasript als imperative Programmierung, sogenannter Spaghetticode ist gleichwertig erlaubt. Erstellen sie zu einem klassennotierten Code oder einer Ablaufstruktur mindestens eine Grafik. Folgende UML-Darstellungen können gewählt werden:

- Objektdiagramm
- Klassendiagramm
- Struktogramm, insbesondere
  - Einfach- oder Mehrfachverzweigung
  - Switch-Case-Anweisung
  - kopf- oder fußgesteuerte Schleife

### Kommentare

Der Quellcode ist zu kommentieren. Erklären sie im Kommentar so knapp wie möglich die Sequenz, die Funktion, die Struktur, das Schlüsselwertpaar oder das Array. Verzichten sie soweit dies möglich ist, auf mehrzeilige Kommentare. Versuchen sie die richtigen Termini, wie Literal, Token und Parser zu verwenden. Es ist keine Dokumentenverarbeitung mit AI vorgesehen, so dass kein DocParserContent erwartet wird. Erklären sie ihren Code umgangssprachlich.

# Wahlaufgaben

Quelle: Manasa Ramakrishnan: „Wie Sie mit diesen spannenden Projektideen Ihr JavaScript-Potenzial steigern können“ (09.12.2022), (URL: <https://emeritus.org/blog/coding-javascript-project-ideas/>)

## 1. Einfache Wetteranwendung Level 1

Erstellen Sie eine Wetteranwendung, die Wetterdaten von einer API (Application Programming Interface) abrufen und dem Benutzer anzeigen. Entwerfen Sie zunächst eine einfache Benutzeroberfläche mit HTML und eine schmale CSS. Verwenden Sie dann JavaScript, um eine API-Anfrage zu stellen und die Wetterinformationen basierend auf dem Standort des Benutzers oder einer ausgewählten Stadt dynamisch zu aktualisieren. Darüber hinaus können Funktionen wie die Anzeige der aktuellen Wetterbedingungen, der Temperatur, der Luftfeuchtigkeit und einer Fünf-Tage-Vorhersage hinzugefügt werden.

## 2. Fitness-Tracking-Anwendung Level 2

Eine Fitness-Tracking-Anwendung, die es Benutzern ermöglicht, ihre Trainingsroutinen zu verfolgen, Ziele zu setzen und den Fortschritt zu überwachen, ist eine großartige Projektidee für Entwickler auf mittlerer Ebene. Implementieren Sie zunächst Funktionen wie Trainingsprotokollierung, Trainingskategorisierung, Trainingsverlauf und Leistungsstatistiken. Integrieren Sie auch APIs oder tragbare Geräte, um Echtzeitdaten wie Herzfrequenz oder zurückgelegte Distanz während des Trainings zu erfassen.

## 3. Kollaborativer Code-Editor Level 3

Erstellen Sie einen Code-Editor für die Zusammenarbeit in Echtzeit, in dem mehrere Benutzer Code in einem gemeinsamen Arbeitsbereich schreiben und bearbeiten können. Implementieren Sie im Idealfall Funktionen wie Syntaxhervorhebung, automatische Codevervollständigung und Echtzeit-Cursorsynchronisierung. Verwenden Sie dazu Technologien wie WebSockets oder ShareDB für die Zusammenarbeit in Echtzeit und sorgen Sie für Sicherheitsmaßnahmen zum Schutz des Benutzercodes.

## Impressum

Diplom-Berufsschullehrerin Jana Koch (TU) im Auftrag der **cbm GmbH**, Wegesende 3 – 4  
28195 Bremen. Kontakt: [info@cbm-bremen.de](mailto:info@cbm-bremen.de) | Januar 2024.