

Lehrplan

MODUL	THEMA	BESCHREIBUNG	UE
MODUL 1: Grundlagenausbildung			600
	Mathematische und technische Grundlagen	In diesem Kurs werden den Teilnehmenden die für die Weiterbildung relevanten Tools vorgestellt. Ferner bietet der Kurs eine Einführung zum rechnen in verschiedenen Zahlensystemen.	30
	Linux Grundlagen	Vermittlung von Grundlagenwissen zu Open Source, Open Source Lizenzen, dem Linux Betriebssystem sowie Programmieren mit der BASH entlang des Linux Essentials Curriculum des Linux Professionals Institut (LPI), mit dem Ziel die Linux Essentials Prüfung abzulegen.	200
	IT Netzwerke	Vermittelt die Fähigkeit im Bereich der Konzeption, Implementierung und Sicherung von Computernetzwerken. Es umfasst das Verständnis der Netzwerkarchitektur, Protokolle, Sicherheitsmaßnahmen und Best Practices zur Gewährleistung eines sicheren und effizienten Datenverkehrs innerhalb eines Netzwerks.	30
	Cybersecurity	Schutz von Computernetzwerken vor unbefugtem Zugriff, Missbrauch, Datenverlust oder anderen Sicherheitsbedrohungen. Dies umfasst die Implementierung von Sicherheitsmaßnahmen wie Firewalls, Intrusion Detection Systems (IDS), Verschlüsselung und Zugriffskontrollen, um die Integrität, Vertraulichkeit und Verfügbarkeit von Netzwerkressourcen zu gewährleisten.	10
	Python Basics	Erlernen und Anwenden der Programmiersprache Python zur Entwicklung von Software und Lösungen. Es umfasst das Verständnis der Syntax, Funktionen, Datenstrukturen und Bibliotheken von Python, um Programme zu schreiben, Daten zu analysieren, Webanwendungen zu erstellen und vieles mehr.	30
	Datenbanken	Grundlagen und Prinzipien von Datenbanken, einschließlich der Strukturierung, Speicherung und Organisation von Daten. Es umfasst das Verständnis von Datenbankmodellen, Abfragesprachen und grundlegenden Konzepten wie Tabellen, Datentypen und Beziehungen, um effizientes Datenmanagement und -zugriff zu ermöglichen.	50
	Einführung in das Cloud Computing	Einführung in das Thema Cloud und Abgrenzung. Definition sowie Beispiele aus dem Alltag als Grundlage für AWS. Vorstellung der AWS Cloud sowie der für das bestehen der AWS Cloud Practitioner	30

	Jumpstart AWS	Prüfung relevanten Services.	160
	Prüfungsvorbereitung AWS Cloud Practitioner Prüfung	Wiederholung aller Prüfungsrelevanten Themen.	60
			1100
MODUL II: Hauptqualifizierung			
	Agile Softwareentwicklung mit Scrum	<ul style="list-style-type: none"> - TN kennen unterschiedliche agile Methoden, z.B: SCRUM, Kanban, SAFe - TN kennen das V-Modell als Vorgehensmodell der Softwareentwicklung - TN können die Unterschiede zwischen Scrum und V-Modell beschreiben. <p>Gruppenaufgabe: Es soll ein Kursspezifisches Mini-Projekt umgesetzt werden. Es werden Gruppen gebildet mit 4 Teilnehmern und entsprechender Rollenverteilung. Es wird das agile Projektverfahren im Zeitraffer erlebt und alle wichtigen Stationen erörtert. Das Ergebnis wird am letzten Tag zusammengetragen und vorgestellt.</p>	50
	Versionsverwaltung mit Git	<p>TN können den Begriff Versionsverwaltung erklären</p> <p>TN kennen verschiedenen Versionsverwaltungssysteme: SVN, CVS, Git TN haben Git lokal installiert</p> <p>TN können eigene Repositories anlegen</p> <p>TN können in das lokalen Repository commiten TN können Dateien und Verzeichnisse ignorieren</p> <p>TN können ein Repository via git bundle packen und z.B via E-Mail versenden TN können Github Repositories clonen</p> <p>TN haben einen Github Account</p> <p>TN können öffentliche Github Repositories forken</p> <p>TN können in eigene remote Repositories z.B auf Github publishen</p> <p>TN können die folgenden Begriffe erklären: Commit, Branch, Diff, Merge, Local Repository, Remote Repositories, Patch, Staging</p> <p>TN können Ähnlichkeiten und Unterschiede von Github, Gitlab und "Gerrit Code Review" beschreiben</p>	20

	Statische Website mit HTML und CSS	TN können formale Sprachen der IT beschreiben TN können Einsatzzwecke für Programmiersprachen, Auszeichnungssprachen, Modellierungssprachen erklären TN können konkrete formale Sprachen benennen und deren Einsatzzweck erklären TN können einfache Verzeichnisstrukturen für Webprojekte beschreiben und erstellen	40
	Internet Protokolle	TN kennen relevante Web Protokolle: HTTP, HTTPS, FTP, SSL, DNS TN kennen HTTP Funktionen TN können die HTTP Methoden (GET, POST, HEAD PUT, DELETE, TRACE, CONNECT) aufzählen und erläutern TN können relevanten HTTP Codes erklären, z.B 200, 404, 500 TN können die Protokolle	20
	HTTP Server	TN kennen verschiedenen HTTP Server TN können Server aufzählen, z.B Apache, Nginx TN können selbstständig einen HTTP Server einrichten	30
	Prototypen erstellen	TN haben einen kostenlosen Figma Account TN können Figma zum Prototyping von Web UIs nutzen TN haben ein minimales Web-UI mit Figma entworfen	30
	Frontends für das Web entwickeln	Warum Frameworks das Entwicklerleben vereinfachen: Was ist ein Framework? Wie werden sie eingesetzt? Vergleich von verschiedenen Frameworks wie vue, nuxt, next und React.	40

	JavaScript Programmierung	<p>JavaScript Programmierung</p> <p>Grammatik und die Syntax von JavaScript: Deklarationen, Scope, Hosting, Datentypen, Literale - Programablauf/Kontrollfluß mit bedingte Anweisungen zu steuern: if ... else; switch, try/catch/throw, Error, Promis - Schleifen programmieren: for , while, do...while, break/continue, for..in, for..of</p> <p>typische Operatoren Ausdrücke: Zuweisungen, Vergleiche sowie arithmetische, bitwise, logische und bedingte (ternär) Operatoren</p> <p>Programmteile in Funktionen gliedern: deklarieren, definieren, aufrufen, Funktions- Scope, Argumente</p> <p>Nummern und Datumsangaben: Literale, Math, Date</p> <p>Formatierung und einfachen Auswertung von Zeichenketten: String, Template-Literale, I18n, RegEx</p> <p>Datenstrukturen kennen und nutzen: Arrays</p> <p>Datenstrukturen mit Schlüsseln: Map, Set Konventionen</p> <p>zur Quellcode-Formatierung JavaScript Quellcode in externe Dateien ausgelagern</p>	60
	Webentwicklung mit NodeJS	<p>TN können den historischen Abriss der Entstehungsgeschichte von Nodejs wiedergeben</p> <p>TN können Ziele und Nutzen von Nodejs benennen TN</p> <p>können NodeJS lokal installieren und nutzen TN</p> <p>können den interaktiven Modus nutzen</p> <p>TN können das Nodejs Modulsystem beschreiben TN</p> <p>können NodeJS auf der Kommandozeile nutzen</p> <p>TN habe eigene JavaScript "Konsolen-Anwendungen" erstellt und nutzen diese lokal TN</p> <p>kennen NPM und nutzen diesen</p> <p>TN haben eine NodeJS Webserver erstellt</p> <p>TN können Web-Formulare via POST auswerten</p> <p>TN können einfache NodeJS Anwendungen erstellen TN</p> <p>kennen Express als Ergänzung zu NodeJS</p> <p>TN können APIs erstellen und diese via HTTP-Methoden aufrufen TN</p> <p>dokumentieren Ihre API im Quellcode mittels swagger-jsdoc</p> <p>TN kennen Unterschiede zwischen CommonJS und ES6 bezüglich Module und Imports</p> <p>TN kennen das NodeJS Ökosystem, insbesondere npm, express, socket.io, und nutzen dieses</p>	60

	Web Apps mit React entwickeln	Einführung in React und die Entwicklung einer Web-Application.	50
	Unit Testing mit JestJS	<p>Einführung in das Softwaretesting:</p> <p>Was bedeutet Softwaretesting? Welche Vor- und Nachteile gibt es? Unit Tests z.B mit jest</p> <p>Frontend Testing / Automation mit Selenium</p> <p>Systemtest mit cucumber</p> <p>Testing (TDD, BDD basics)</p> <p>Entwicklung einen einfachen unit-Test für die Web-Anwendung aus dem Praxis Projekt</p> <p>Entwicklung einen http basierten Integrationstest für die Web-Anwendung aus dem Praxis Projekt</p>	50
	Testautomatisierung mit Selenium	<p>Wiederholung und Vertiefung von Softwaretesting und Automation umfassen das Verständnis grundlegender Testkonzepte, die Fähigkeit zur Anwendung von Testautomatisierungstools, das Wissen über Best Practices im Softwaretesting, die Bewertung von Testergebnissen und die Kenntnisse von Testmanagement-Tools und Praktiken.</p> <p>Unit-Tests Integrationstest</p> <p>Systemtests</p> <p>Funktionstests</p> <p>Akzeptanztests</p> <p>Regressionstests</p> <p>Last- und Performancetests</p> <p>Sicherheitstests</p> <p>Usability-Tests</p>	40
	ISTQB Basics für CTFL	Softwaretesting nach Industrie-Standards des "International Software Testing Qualifications Board"	40
	Prüfungsvorbereitung CTFL	Vorbereitung und Prüfungssimulation zum Certified Tester - Foundation Level v4.0	20

	PRAXIS: Web App Development	<p>TN sollen selbständig ein eigenes Web Projekt realisieren</p> <p>TN haben ein oder mehrere Repositories zur Verwaltung des Web Projekts erstellt TN pflegen eine Projektdokumentation z.B unter Github</p> <p>TN haben im Vorfeld eine Ideen-Skizze erstellt TN haben das Projekt umgesetzt</p> <p>TN haben nach Abschluss des Projekt offene Punkte erstellt</p> <p>TN haben eine Projektpräsentation erstellt, z.B mittels Google Slides TN haben die Präsentation im Rahmen des Kurses gehalten</p> <p>TN haben evtl. die Präsentation aufgezeichnet</p> <p>TN haben einen "Ordner" mit allen Unterlagen erstellt und abgegeben</p>	150
	API Design mit Postman	<p>TN können Unterschied zwischen URL und URI erklären TN können REST erläutern</p> <p>TN können HTTP-Methoden aufzählen und beschreiben TN haben Postman lokal installiert</p> <p>TN haben einen eigenen Postman Account</p> <p>TN können GET Aufrufe via Postman absetzen und interpretieren TN können eigne APIs mittels Postman definieren</p> <p>TN können APIs mittels Postman manuell testen</p> <p>TN können APIs mittels Postman automatisiert testen TN können APIs mittels Postman dokumentieren</p>	50
	Grafana Basics	<p>Zentrales Logging und Auswerten und Softwaremetriken mittels Grafana (Open-Source-Dashboarding- und Visualisierungstool) und Loki (Open-Source-Log-Aggregationssystem) Schwerpunkte:</p> <p>Einführung in Grafana</p> <p>Erstellen, konfigurieren und Anpassen von Dashboards</p> <p>Integration von CloudWatch</p> <p>Einführung in Loki</p> <p>Installation und Konfiguration des Log-Aggregationssystems Durchsuchung von Log-Daten</p> <p>Erstellen von eigenen Überwachungs- und Analyseumgebung Daten visualisieren</p> <p>Log-Daten überwachen</p>	50

	Prometheus Basics	Vergleich und Anwendung von CloudWatch (AWS Monitoring-Service) und Prometheus (Open-Source-Framework für das Monitoring und das Alerting). Schwerpunkte: Integration von Prometheus in Softwarekomponenten Erfassen von Metriken und Log-Daten Eigene Monitoring-Lösungen mit Prometheus aufbauen. Interpretation von Metriken	50
	Microsoft Azure Basics	Vorstellung der Microsoft Azure Cloud sowie der für das bestehen der AZ-900 Prüfung relevanten Services	160
	Vergleich & Gegenüberstellung AWS und Microsoft Azure	Wo sind die Unterschiede und Gemeinsamkeiten der beiden Cloud Anbieter? Wie man sich für die richtige Cloud entscheiden kann.	15
	Microsoft Azure Prüfungsvorbereitung	Vorbereitung und Prüfungssimulation zur AZ-900 Prüfung	75
			600
MODUL III: Spezialisierung und Arbeitsmarktvorbereitung			

	DevOps Basics	TN können das Berufsbild des DevOps Engineer beschreiben TN können das Akronym CALMS erklären (Culture - Automation - Lean - Measurment - Sharing) TN können den kulturellen Unterschied zwischen klassischen Organisationen und "DevOps" Organisationen erklären TN können die Begriffe Pipeline, Stage, Build, Test, Release erörtern TN können einen DevOps Lifecycle aufzeichnen und beschreiben TN können typische Methoden und Werkzeuge aufzählen TN können Methoden und Werkzeuge im DevOps Lifecycle einordnen TN können die Begriffe DevSecOps und BisDevOps einordnen	30
	Docker	Was sind Containertechnolgien? Unterschied zwischen Container und Virtuellen Maschinen? Wie erstelle ich einen Container? Wie starte / stoppe ich Container? Was sind stateful und stateless container? Kubernetes Containerisierung von AWS	100
	Übersicht Container Orchestrierung	TN können Container Orchestrierung erklären TN können verschiedenen Orchestrierung Werkzeuge benennen. Z.B Kubernetes, Open Shift, Docker Swarm	10
	Kubernetes	Kubernetes	60
	Terraform	Verwaltung und Automatisierung von Infrastruktur durch Code mithilfe des Tools Terraform. Einführung in Terraform Nutzung von Terraform CLI Erstellung und Bereitstellung von Infrastrukturressourcen in AWS Verwendung/Interaktion von Modulen Zustandsverwaltung für eine automatisierte und skalierbare Infrastukturverwaltung Automatisierung durch CI/CD Systeme wie github Actions	50

	Ansible	<p>Konfigurationsmanagement- und Orchestrierungstool, das die Automatisierung von IT-Infrastruktur und Anwendungen ermöglicht. Die Lehrinhalte umfassen das Erlernen der</p> <p>Ansible-Syntax Erstellen von Playbooks Verwaltung von Inventaren Konfigurieren von Systemen Automatisieren von Aufgaben für effiziente und konsistente.</p>	50
	CICD Grundlagen	<p>Begriff CICD beschreiben:</p> <p>automatisierten Prozess der Continuous Integration (CI) und Continuous Deployment (CD) von Software. Dieser Prozess umfasst das Einrichten von Pipelines, die Code-Builds, automatisierte Tests, Code-Reviews und die automatische Bereitstellung von Software in einer produktionsfähigen Umgebung. Ziel von CICD ist es einen schnellen und effizienten Entwicklungs- und Bereitstellungszyklus zu ermöglichen. Vergleich verschiedener Systeme wie</p> <p>Jenkins github actions Bamboo</p> <p>Erweiterte Nutzung und Konfiguration von Pipelines als Automatisierungswerkzeug für den Softwareentwicklungsprozess in Verbindung mit der Versionskontrollplattform GitHub. Es umfasst die Erstellung und Anpassung von Pipelines zur Automatisierung von Build-, Test- und Bereitstellungsschritten und die Integration mit GitHub für ein effizientes und reibungsloses Continuous Integration und Continuous Devlivery (CI/CD) Workflow. Automatisierung der Praxisprojekte</p>	20
	CICD mit Jenkins	<p>Die bekannteste Lösung auf dem Prüfstand - wie arbeitet man mit Jenkins on-premise.</p>	40

	CICD mit Github Actions	Die Cloud Alternative mit github Actions: Konfiguration, Nutzung und Einbindung auch für Open Source Projekte.	40
	Kursabschluss	Praxis Projekt aus der Sicht des DevOps Ingenieurs: Umsetzung eines agilen Softwareprojekts sowie der Vollständigen Automatisierung des Software-Lebenszyklus und der Infrastruktur als Code. Präsentation am letzten Tag vor Arbeitgeber-Gästen.	200