Kursstruktur: Web Development mit Django, RESTful APIs & React

Django Grundlagen & HTTP-Verständnis

- **Web Development Basics & HTTP-Protokoll**
  - **Skript 1.1: Web Development: Wie funktioniert das Internet? Client, Server, HTTP-Request/Response-Zyklus, URLs, Ports, Endpoints, REST-Konzepte (grundlegend).**
    - Fokus: Das Fundament für alles Weitere. Erläutert `requests`, `routing`, `endpoints` und Datenaustausch.
    - Cheat Sheet: Grundlegende HTTP-Methoden, URL-Struktur.
  - **Leitfaden-Projekt (Demonstration):** Diskussion, wie das Polls-Projekt in diesem Kontext Daten austauschen wird.
  - **Schüler-Projekt (Eigenständig):** Diskussion der Architektur der Recipe Sharing Platform.
  - **Übung:** Kurze Recherche-Aufgabe zu verschiedenen HTTP-Statuscodes und ihrer Bedeutung.
- **Tag 2: Django Setup & Erste Schritte mit URLs**
  - **Skript 1.2: Python für Webentwicklung: Virtuelle Umgebungen (`venv`), `pip`, Projektstrukturierung (dot vs. no-dot).**
    - Fokus: Sauberes Setup für Entwicklung.
    - Cheat Sheet: `python -m venv`, `source activate`, `pip install django`.
  - **Skript 1.3: Django Projekt- & App-Struktur: `settings.py`, `urls.py` Einführung. Erste URL-Routings (`path()`, `name`).**
    - Fokus: Aufbau einer Django-Anwendung verstehen und erste Routen definieren.
    - Cheat Sheet: `INSTALLED_APPS`, `django-admin startproject`, `python manage.py startapp`, `path()`, `name`.
  - **Leitfaden-Projekt (Demonstration):** Initiales Django-Projekt-Setup (Polls) und erste `myapp/urls.py` mit einem simplen View.
  - **Schüler-Projekt (Eigenständig):** Initiales Django-Projekt-Setup (Recipe Sharing) und erste `myapp/urls.py`.
  - **Übung:** Erstelle ein einfaches Django-Projekt mit zwei Apps und definiere für jede App eine simple Route, die einen "Hello World"-String zurückgibt.
- **Tag 3: Django Views & URL-Parameter**
  - **Skript 1.4: Django Views: Funktionale Views (`FBV`), `HttpResponse`, `render()`, Kontext. URL-Routing: Pfad-Konverter (`<str:name>`, `<int:id>`).**
    - Fokus: Wie Anfragen verarbeitet und Antworten gesendet werden, dynamische URLs.
    - Cheat Sheet: `def my_view(request):`, `HttpResponse()`, `render()`, `path('<str:param>/')`.
  - **Leitfaden-Projekt (Demonstration):** Views und URLs für die Umfragenliste und Detailansicht, die Platzhalter verwenden.
  - **Schüler-Projekt (Eigenständig):** Views und URLs für die Rezeptliste und Detailansicht, die Platzhalter verwenden.
  - **Übung:** Implementiere Views, die URL-Parameter (`name`, `id`) auslesen und in der Antwort verwenden. (Analog zu `Django_URL_Routing_and_Query_Parameters_Guide_v2.pdf` Beispiel 1 [cite: 51])
- **Tag 4: Django Query-Parameter & MVT-Architektur**
  - **Skript 1.5: URL-Routing: Query-Parameter (`request.GET`). MVT-Architektur im Detail (Model-View-Template) vs. MVC.**
    - Fokus: Optionale Parameter in URLs und tiefes Verständnis der Django-Architektur.
    - Cheat Sheet: `request.GET.get('key', default_value)`, MVT-Komponenten.
  - **Leitfaden-Projekt (Demonstration):** Implementierung einer einfachen Suchfunktion für Umfragen (Query-Parameter).
  - **Schüler-Projekt (Eigenständig):** Implementierung einer einfachen Suchfunktion für Rezepte (Query-Parameter).
  - **Übung:** Erstelle eine View, die Query-Parameter für Filter- oder Suchfunktionen verarbeitet. (Analog zu `Django_URL_Routing_and_Query_Parameters_Guide_v2.pdf` Beispiel 2 [cite: 62])
- **Tag 5: Django Models & ORM (Grundlagen)**
  - **Skript 1.6: Django Models: Einführung, Felder (`CharField`, `TextField`, `IntegerField` etc.), Migrationen.**
    - Fokus: Wie Daten in der Datenbank abgebildet werden.
    - Cheat Sheet: `models.CharField`, `models.TextField`, `python manage.py makemigrations`, `python manage.py migrate`.
  - **Leitfaden-Projekt (Demonstration):** Definition der `Question` und `Choice` Modelle.
  - **Schüler-Projekt (Eigenständig):** Definition der `Recipe`, `Ingredient`, `Step` Modelle.

- **Übung:** Erstelle ein einfaches Modell (`Product` oder `Author/Book`) mit verschiedenen Feldtypen, führe Migrationen durch.

Woche 2: Django Datenbank-Interaktion & Templates

- **Tag 6: Django ORM (CRUD) & Admin Interface**
  - **Skript 2.1: Django ORM: Abfragen (`all()`, `get()`, `filter()`, `create()`, `save()`, `update()`, `delete()`), Admin Interface konfigurieren.**
    - Fokus: Praktische Interaktion mit der Datenbank über den ORM und schnelle Datenverwaltung.
    - Cheat Sheet: `objects.all()`, `objects.create()`, `admin.site.register()`, `python manage.py createsuperuser`.
  - **Leitfaden-Projekt (Demonstration):** Registrierung der `Question`/`Choice` Modelle im Admin. Erste Daten im Admin anlegen.
  - **Schüler-Projekt (Eigenständig):** Registrierung der `Recipe`/`Ingredient`/`Step` Modelle. Erste Daten im Admin anlegen.
  - **Übung:** Nutze den Django Shell für CRUD-Operationen auf einem Modell. Fülle den Admin-Bereich mit Beispieldaten.
- **Tag 7: Django Model Relationships (`ForeignKey`, `ManyToMany`, `OneToOne`)**
  - **Skript 2.2: Vertiefung: `ForeignKey`, `ManyToMany`, `OneToOne`-Beziehungen, `on_delete` Optionen, `through` Argument.**
    - Fokus: Modellierung komplexer Datenbeziehungen.
    - Cheat Sheet: `models.ForeignKey(to, on_delete)`, `models.ManyToManyField(to, through)`, `models.OneToOneField(to, on_delete)`.
  - **Leitfaden-Projekt (Demonstration):** Überprüfung und Anpassung der Beziehungen in den Polls-Modellen.
  - **Schüler-Projekt (Eigenständig):** Implementierung der Beziehungen in den Recipe-Modellen (z.B. Rezept hat viele Zutaten über eine ManyToMany-Beziehung mit `through` für Mengeinheiten).
  - **Übung:** Implementiere verschiedene Beziehungen in einem Testprojekt. (`Author`/`Book`, `Student`/`Course`, `User`/`Profile`).
- **Tag 8: Django Templates: Variablen, Tags, Filter**
  - **Skript 2.3: Django Templates: Variablen (`{{ var }}`), Tags (`{% if %}`, `{% for %}`), Filter (`| upper`, `| length`, `| date`).**
    - Fokus: Dynamische Inhalte in HTML.
    - Cheat Sheet: Wichtige Template-Tags und Filter.
  - **Leitfaden-Projekt (Demonstration):** Umfragenliste mit Schleifen und Bedingungen. Detailseite mit formatierten Daten.
  - **Schüler-Projekt (Eigenständig):** Rezeptliste mit Schleifen, Bedingungen und Filtern (z.B. `title` für Rezeptnamen, `date` für Erstellungsdatum).
  - **Übung:** Bearbeite die Aufgabe zum Anwenden von For Loops, If Statements und Filtern in Django Templates[cite: 42].
- **Tag 9: Django Template Inheritance & Includes**
  - **Skript 2.4: Template Inheritance (`{% extends %}`, `{% block %}`, `{{ block.super }}`). Wiederverwendung mit `{% include %}`. Statische Dateien (`static` Tag).**
    - Fokus: Konsistentes Design und Code-Wiederverwendung.
    - Cheat Sheet: `{% extends 'base.html' %}`, `{% block content %}`, `{% include 'partial.html' %}`, `{% static 'path' %}`.
  - **Leitfaden-Projekt (Demonstration):** Erstellung einer `base.html` für das Polls-Projekt mit Header/Footer und Navigation.
  - **Schüler-Projekt (Eigenständig):** Erstellung einer `base.html` für das Recipe-Projekt, alle bestehenden Templates erben von dieser. Einbindung eines einfachen CSS-Frameworks (z.B. Bulma oder TailwindCSS CDN) und erster eigener Styles.
  - **Übung:** Bearbeite die Aufgabe zur Erstellung eines Grundgerüsts mit Template-Inheritance und Navigation (Aufgabe 1 & 2 ohne Formulare/Bilder noch)[cite: 3].
- **Tag 10: Django Forms (Normal Forms)**
  - **Skript 2.5: Django Forms: Einführung, Feldtypen, Widgets, manuelle Validierung. `form.is_valid()`, `form.cleaned_data`.**
    - Fokus: Verarbeitung von Benutzereingaben, wenn kein direktes Model dahintersteht.
    - Cheat Sheet: `forms.Form`, `forms.CharField(widget=forms.TextInput)`, `{{ form.as_p }}`, `{% csrf_token %}`.
  - **Leitfaden-Projekt (Demonstration):** Einfaches Kontaktformular oder Feedback-Formular (nicht an Modell gebunden).

- **Schüler-Projekt (Eigenständig):** Implementierung eines allgemeinen Suchformulars für Rezepte (nicht an Modell gebunden).
- **Übung:** Erstelle ein einfaches Formular (z.B. Pizza Order Form), das Benutzereingaben validiert und anzeigt.

Woche 3: Django ModelForms, Authentifizierung & MySQL

- **Tag 11: Django ModelForms**
  - **Skript 3.1: Django ModelForms: Automatisches Generieren von Feldern, Validierung vom Model, `form.save()`, Anpassen von Feldern (`Meta.widgets`).**
    - Fokus: Effiziente Formularerstellung für Datenbankoperationen.
    - Cheat Sheet: `forms.ModelForm`, `class Meta: model = ..., fields = ..., widgets = {}`.
  - **Leitfaden-Projekt (Demonstration):** Formular zum Hinzufügen einer Umfrage/Frage mit ModelForm.
  - **Schüler-Projekt (Eigenständig):** Formular zum Hinzufügen/Bearbeiten von Rezepten mit ModelForm.
  - **Übung:** Bearbeite die Aufgabe 4 aus den Django Template und Model Aufgaben: Produkte über ein Formular hinzufügen[cite: 17].
- **Tag 12: Django Benutzerauthentifizierung & Autorisierung**
  - **Skript 3.2: Benutzerauthentifizierung: Login, Logout, Registrierung (`UserCreationForm`, `AuthenticationForm`). Autorisierung: `login_required` Decorator.**
    - Fokus: Absicherung von Webanwendungen und Benutzerverwaltung.
    - Cheat Sheet: `login()`, `logout()`, `login_required`, `UserRegistrationForm (custom)`.
  - **Leitfaden-Projekt (Demonstration):** Implementierung der Benutzerregistrierung, Login/Logout für das Polls-Projekt.
  - **Schüler-Projekt (Eigenständig):** Implementierung der Benutzerregistrierung und Login/Logout für die Recipe Sharing Platform.
  - **Übung:** Bearbeite die Teile der Django Watchlist Project Übung zur Authentifizierung und Login-Required Views[cite: 69].
- **Tag 13: Medienverwaltung (Bilder) & Statische Dateien Vertiefung**
  - **Skript 3.3: Medien: `MEDIA_ROOT`, `MEDIA_URL`, `ImageField`, Dateiuploads (inkl. `Pillow`). Statische Dateien: `collectstatic` und Deployment-Vorbereitung.**
    - Fokus: Handling von hochgeladenen Dateien (insbesondere Bilder).
    - Cheat Sheet: `MEDIA_ROOT`, `MEDIA_URL`, `models.ImageField(upload_to)`, `pip install Pillow`, `python manage.py collectstatic`.
  - **Leitfaden-Projekt (Demonstration):** Hinzufügen eines Bild-Uploads für Umfragen/Fragen.
  - **Schüler-Projekt (Eigenständig):** Hinzufügen eines `ImageField` zum Rezept-Modell, Implementierung des Uploads und Anzeige auf Listen-/Detailseiten. (Angelehnt an `Django_Template_Exercises.pdf` Aufgabe 5 [cite: 21])
  - **Übung:** Zusätzliche Übung zu Bild-Uploads in einem separaten Modell.
- **Tag 14: Datenbankwechsel zu MySQL**
  - **Skript 3.4: Datenbankwechsel: MySQL-Installation (konzeptionell), `mysqlclient`, `settings.py` Anpassung, Daten `dumpdata` & `loaddata`.**
    - Fokus: Praktischer Umstieg auf eine produktionsnähere Datenbank.
    - Cheat Sheet: MySQL-Befehle (`CREATE DATABASE`, `CREATE USER`, `GRANT PRIVILEGES`), `pip install mysqlclient`, `python manage.py dumpdata`, `python manage.py loaddata`.
  - **Leitfaden-Projekt (Demonstration):** Durchführung des Datenbankwechsels für das Polls-Projekt.
  - **Schüler-Projekt (Eigenständig):** Durchführung des Datenbankwechsels für das Recipe-Sharing-Projekt.
  - **Übung:** Eigene kleine Test-App, um den MySQL-Wechsel und Datenimport zu üben.
- **Tag 15: Modul-Review & Projekt-Konsolidierung (Django-Frontend)**
  - **Skript 3.5: Projekt-Konsolidierung & Best Practices für Django-Frontend.**
    - Fokus: Zusammenführung aller bisher gelernten Django-Konzepte, Überprüfung der Clean-Code-Prinzipien für das Frontend mit Templates.
  - **Leitfaden-Projekt (Demonstration):** Abschluss der Django-Template-basierten Implementierung des Umfragesystems.
  - **Schüler-Projekt (Eigenständig):** Abschluss der Django-Template-basierten Implementierung der Recipe Sharing Platform. Ziel ist ein voll funktionsfähiges Django-Frontend.
  - **Übung:** Code-Review-Session: Schüler tauschen sich aus und geben Feedback zu ihren Projekten. Fehlerbehebung.

Woche 4: RESTful APIs mit DRF (Gestraffte Version)

- **Tag 16: RESTful API-Konzepte & DRF-Einführung mit ModelViewSets**
    - **Skript 4.1: Einführung in RESTful APIs: Konzepte (Ressourcen, Statelessness, Idempotenz), HTTP-Methoden, Statuscodes. Django REST Framework (DRF) Basics: Installation, Serializer, und die Einfachheit von `ModelViewSet` für CRUD-Operationen.**
        - Fokus: Schneller Einstieg in vollständige APIs mit den mächtigen DRF-Standardtools.
        - Cheat Sheet: REST-Prinzipien, `pip install djangorestframework`, `rest_framework` in `INSTALLED_APPS`, `serializers.ModelSerializer`, `ModelViewSet`.
    - **Leitfaden-Projekt (Demonstration):** Erstellung von Serializern und einem `ModelViewSet` für die Polls-Ressourcen (Fragen, Auswahlmöglichkeiten).
    - **Schüler-Projekt (Eigenständig):** Erstellung von Serializern und einem `ModelViewSet` für die Recipe-Ressourcen (Rezepte, Zutaten, Schritte).
    - **Übung:** Erstelle ein einfaches DRF-Projekt mit einem Modell und einem `ModelViewSet`, um die CRUD-Funktionalität über die API zu testen (z.B. mit Postman/Insomnia oder Browser-API).
- **Tag 17: Authentifizierung & Berechtigungen in DRF**
    - **Skript 4.2: Authentifizierung & Berechtigungen in DRF: `TokenAuthentication`, `IsAuthenticated`, `IsAdminUser`, `IsOwnerOrReadOnly` Custom Permissions.**
        - Fokus: Absicherung von APIs.
        - Cheat Sheet: `DEFAULT_AUTHENTICATION_CLASSES`, `DEFAULT_PERMISSION_CLASSES`, `TokenAuthentication`, `IsAuthenticated`.
    - **Leitfaden-Projekt (Demonstration):** Absicherung der Umfragen-API mit Token-Authentifizierung und Berechtigungen.
    - **Schüler-Projekt (Eigenständig):** Absicherung der Recipe-API, z.B. nur registrierte Nutzer dürfen Rezepte erstellen, nur der Ersteller darf sie bearbeiten/löschen.
    - **Übung:** Absicherung einer Test-API, sodass nur authentifizierte Benutzer Schreibzugriff haben.

Woche 5: React Frontend & Integration (5 Tage)

- **Tag 18: Einführung in React: Komponenten, JSX, Props**
    - **Skript 5.1: Einführung in React: Komponenten (`Function Components`), JSX, Props, `useState` Hook (Grundlagen).**
        - Fokus: Grundlegende React-Konzepte verstehen.
        - Cheat Sheet: `function MyComponent(props)`, JSX-Syntax, `useState`.
    - **Leitfaden-Projekt (Demonstration):** Initiales React-Setup mit `Vite` oder `create-react-app` für ein simples React-Frontend zum Polls-Projekt. Erste statische Komponenten.
    - **Schüler-Projekt (Eigenständig):** Initiales React-Setup für die Recipe Sharing Platform. Erstellung der ersten statischen Komponenten für die Rezeptliste.
    - **Übung:** Erstelle eine kleine React-App mit mehreren verschachtelten Komponenten und Props.
- **Tag 19: React State & Events, Datenabruf (`Workspace` API)**
    - **Skript 5.2: React State (`useState` vertiefen), Event-Handling, Controlled Components. Datenabruf: `Workspace` API, `useEffect` Hook.**
        - Fokus: Interaktive React-Komponenten und Kommunikation mit dem Backend.
        - Cheat Sheet: `useEffect`, `Workspace()`, `.then()`, `.catch()`.
    - **Leitfaden-Projekt (Demonstration):** React-Komponenten, die Umfragedaten von der Django-API abrufen und anzeigen. Abstimmung über Buttons (POST-Request).
    - **Schüler-Projekt (Eigenständig):** React-Komponenten, die Rezeptdaten von der Django-API abrufen und anzeigen (Liste und Details).
    - **Übung:** Baue ein einfaches Formular in React, das über `Workspace` Daten an einen Dummy-API-Endpunkt sendet.
- **Tag 20: React: Formulare & CRUD-Operationen mit API**
    - **Skript 5.3: React Formulare: Formular-Handling, Senden von POST/PUT/DELETE-Anfragen an die DRF-API.**
        - Fokus: Benutzerinteraktion mit dem Backend über Formulare in React.
        - Cheat Sheet: `Workspace` mit `method: 'POST'`, `headers: {'Content-Type': 'application/json'}`, `JSON.stringify()`.
    - **Leitfaden-Projekt (Demonstration):** Formulare in React zum Erstellen/Bearbeiten von Umfragen.
    - **Schüler-Projekt (Eigenständig):** Formulare in React zum Erstellen, Bearbeiten und Löschen von Rezepten über die Django-API.

    - **Übung:** Erstelle eine React-Komponente, die es ermöglicht, einen neuen Eintrag über ein Formular an deine DRF-API zu senden und die Liste danach zu aktualisieren.

- **Tag 21: React Routing & Grundlegende Navigation**
    - **Skript 5.4: React Routing (`react-router-dom`: `BrowserRouter`, `Routes`, `Route`, `Link`, `useParams`).**
        - Fokus: Navigation im Single-Page Application (SPA) Stil.
        - Cheat Sheet: `react-router-dom` Komponenten, `Link`, `useNavigate`.
    - **Leitfaden-Projekt (Demonstration):** Implementierung grundlegender Navigation im React-Frontend für das Polls-Projekt.
    - **Schüler-Projekt (Eigenständig):** Implementierung von Navigation zwischen Rezeptliste, Rezeptdetails und Formularen im Recipe Sharing Frontend.
    - **Übung:** Erweitere eine bestehende React-App um einfaches Routing zu verschiedenen Seiten.

- **Tag 22: Projekt-Integration & Deployment**
    - **Skript 5.5: Finales Projekt: Django REST Backend mit React Frontend verbinden. Deployment-Konzepte: Statische Dateien über Django servieren, Nginx/Gunicorn (konzeptionell), Build-Prozesse (`npm run build`).**
        - Fokus: Die volle Integration von Backend und Frontend und der Schritt in die "Produktion".
        - Cheat Sheet: `npm run build`, `python manage.py collectstatic`, Überblick über Deployment-Schritte.
    - **Leitfaden-Projekt (Demonstration):** Abschluss der Integration des Polls-Projekts.
    - **Schüler-Projekt (Eigenständig):** Abschluss der Recipe Sharing Platform, Überprüfung der gesamten Funktionalität. Kurzer Überblick über mögliche Deployment-Szenarien.
    - **Übung:** Abschließende Code-Review-Session und Vorbereitung für ein einfaches Deployment (z.B. Django als Host für das React Build).