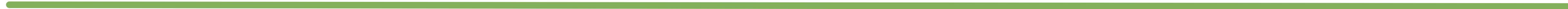


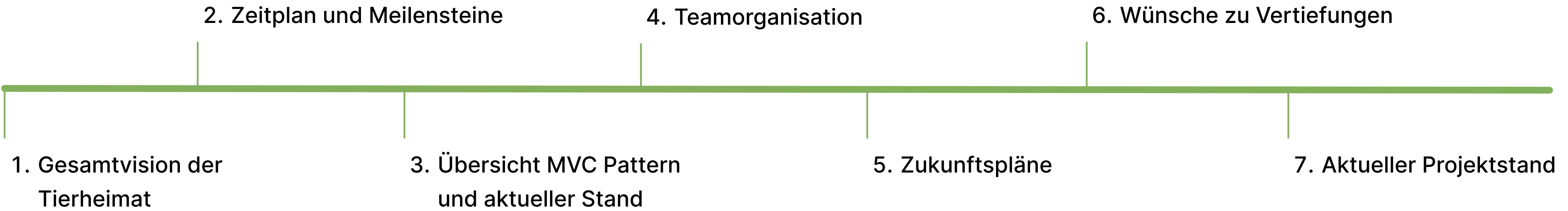


**Dynamische  
Webprogrammierung**  
Entwicklung einer Tierheimwebseite

# Gliederung



# Gliederung



# 1. Gesamtvision der Tierheimat



# 1. Gesamtvision der Tierheimat



# 1. Gesamtvision der Tierheimat

Kernfunktionen:

Zusätzliche Features:



# 1. Gesamtvision der Tierheimat

## Kernfunktionen:

- dynamisches Routing
- laden aller Inhalte aus Datenbank
- Unsere Tiere und vermisste/gefundene Tiere filtern
- Anmeldung und Registrierung Nutzer
- Bearbeitung/Lösung der Einträge (vermisste/gefundene Tiere) durch Admin/angemeldeter Nutzer
- Formulare: Login, Registration, Hilfe anbieten und Vermisste/Gefundene Tiere



## Zusätzliche Features:

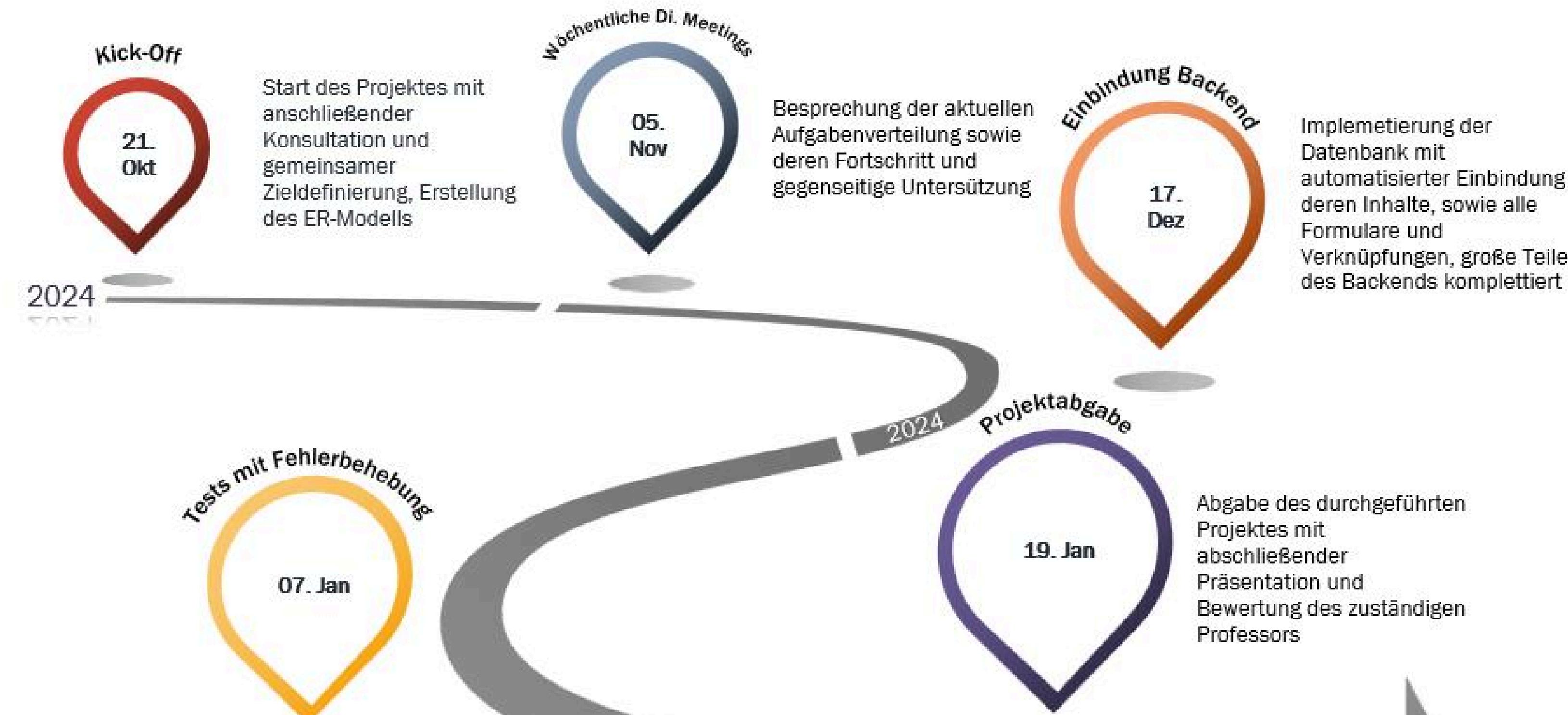
- eine 'Gefällt mir' Funktion von Tieren
- Dynamische Inhalte wie 'Weiterlesen' - Buttons



# 2. Zeitplan und Meilensteine



# 2. Zeitplan und Meilensteine

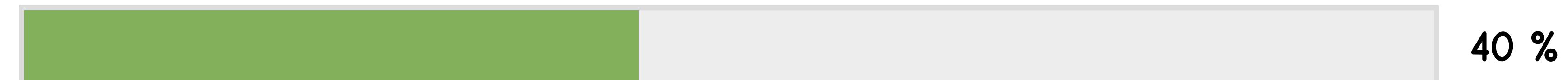


# 3. Übersicht MVC Pattern und aktueller Stand

Live Präsentation



# 3. Übersicht MVC Pattern und aktueller Stand



# 4. Teamorganisation



# 4. Teamorganisation

Gemeinsame Aufgaben:

Aufgaben der Teammitgliedern bisher:



# 4. Teamorganisation

## Gemeinsame Aufgaben:

- Entwurf des ER-Modells
- Nutzung der Tools Jira und WhatsApp zur Kommunikation und Organisation
- Persönliche Treffen jeden Dienstag zur gegenseitigen Unterstützung und Fortschrittsbesprechung

## Aufgaben der Teammitgliedern bisher:



# 4. Teamorganisation

## Gemeinsame Aufgaben:

- Entwurf des ER-Modells
- Nutzung der Tools Jira und WhatsApp zur Kommunikation und Organisation
- Persönliche Treffen jeden Dienstag zur gegenseitigen Unterstützung und Fortschrittsbesprechung

## Aufgaben der Teammitgliedern bisher:

### Stephanie Wachs:

- Entwicklung Routing, Formular "Helfen", dynamisches Laden der Seite "unsere Tiere" (Tiere aus der DB)
- Vereinheitlichung von Menü, Footer und Breadcrumb Navigation auf einer Seite

### Josephina Burger:

- Beginn der Dokumentation
- Umsetzung der MVC-Ordnerstruktur, Formular "Vermisst/Gefunden"

### Lucas-Manfred Herpe:

- Erstellung der SQL-Befehle, Entwicklung der Datenbankstruktur



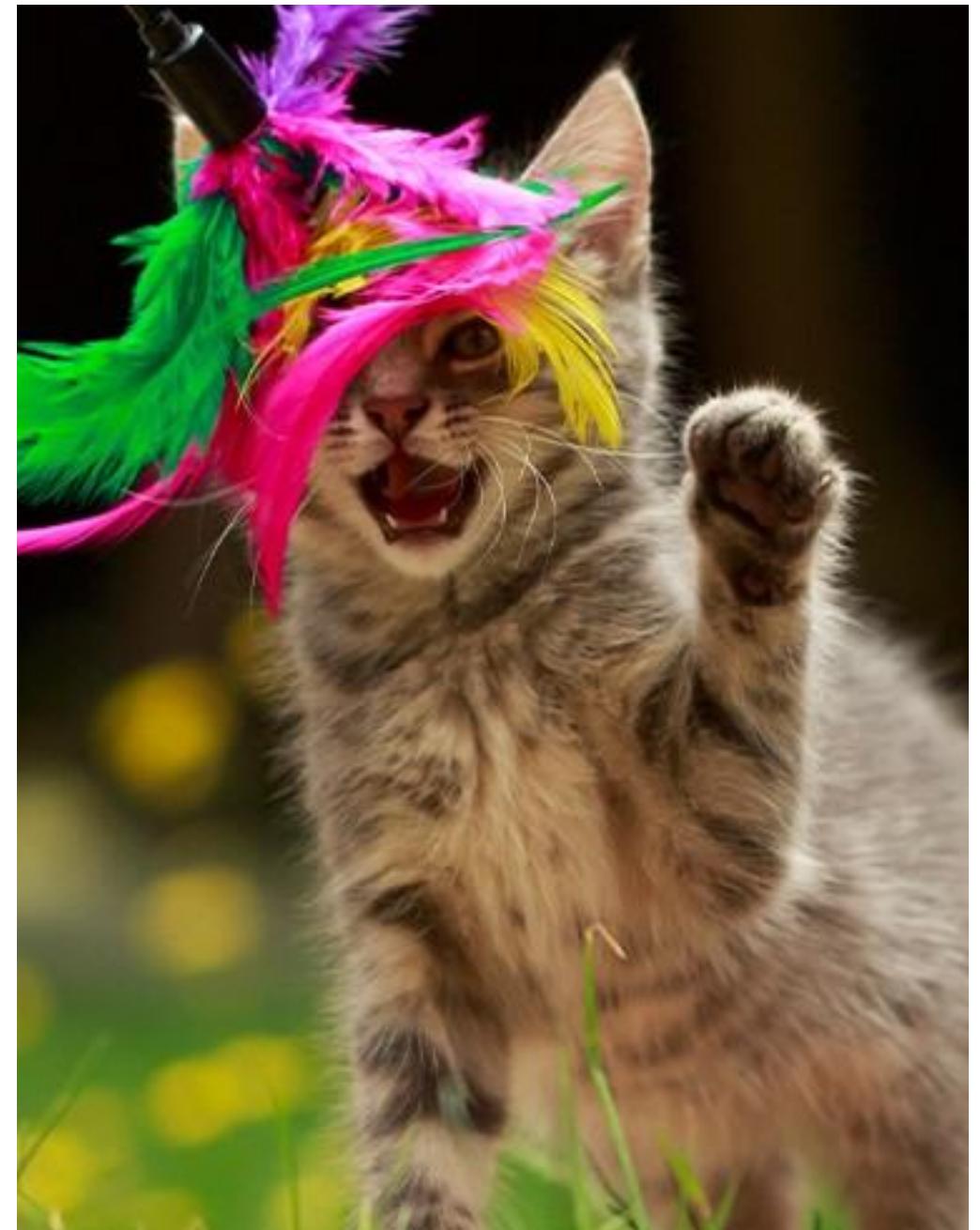
# 5. Zukunftspläne



# 5. Zukunftspläne

Geplante Beendigung des Projekts:

Aktuell sind folgende Arbeitsschritte offen:



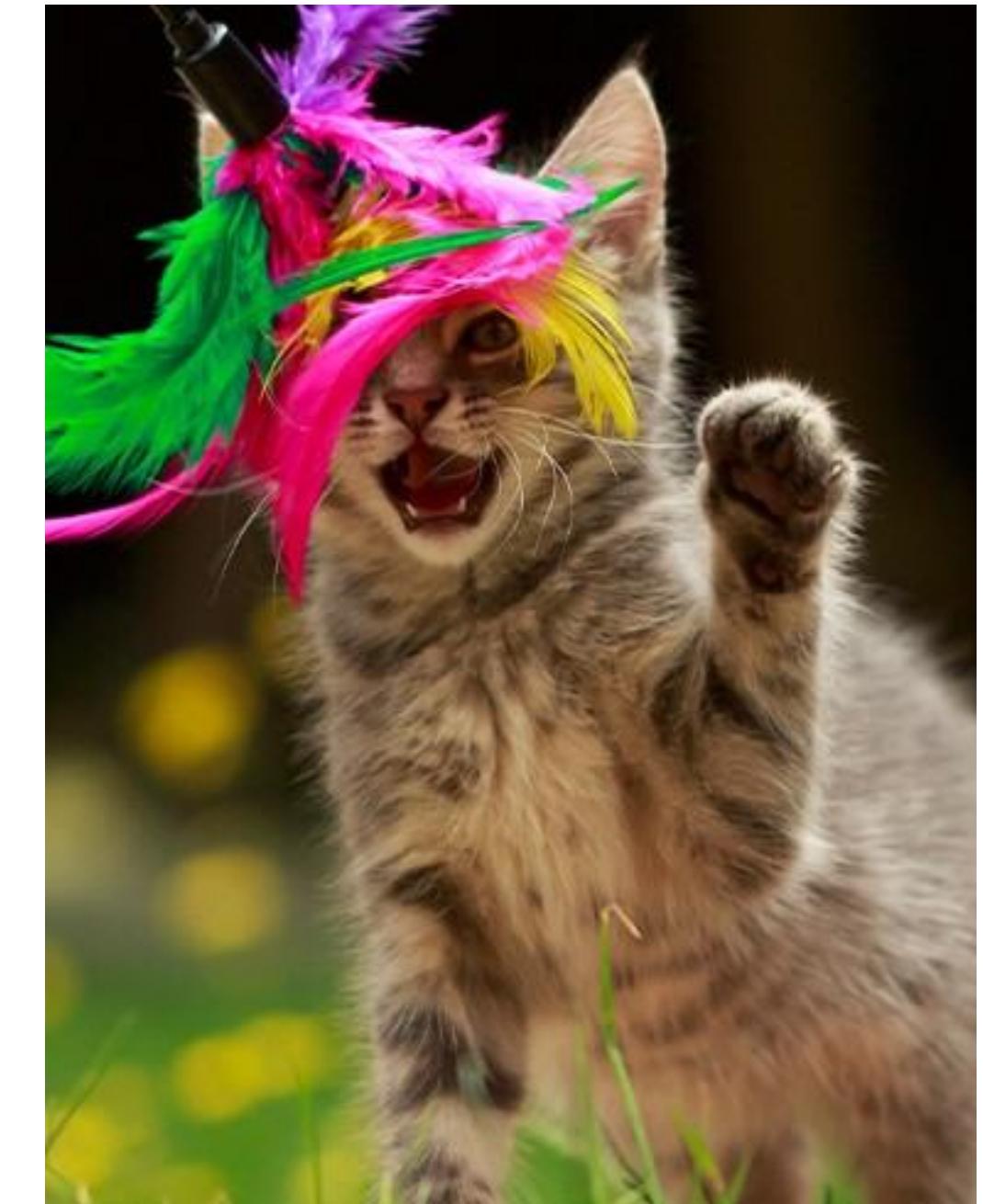
# 5. Zukunftspläne

## Geplante Beendigung des Projekts:

- Orientierung an definierten Meilensteinen für einen erfolgreichen Abschluss
- Geplanter interner Abschluss in KW 01 und 02
- Ausreichend Zeit für interne Absprachen und Fehlerbehebung

## Aktuell noch offene Arbeitsschritte:

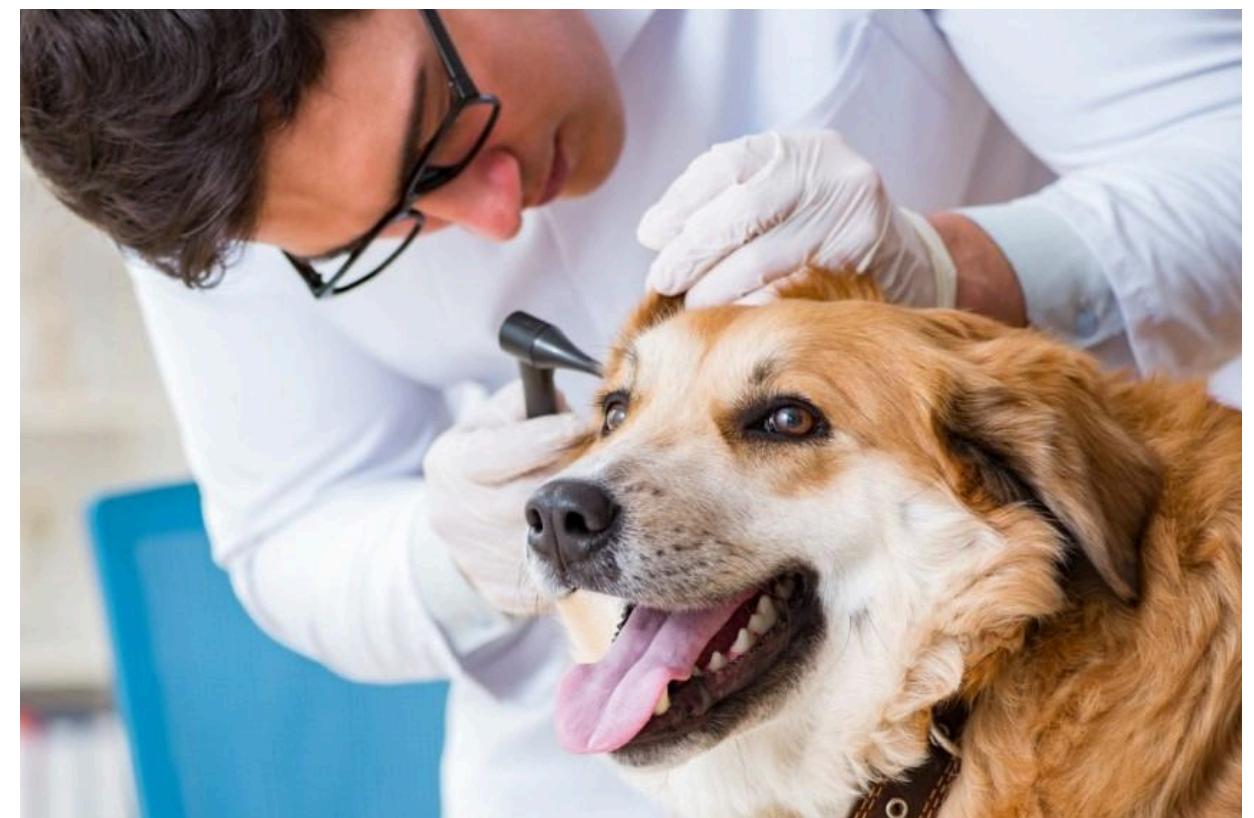
- Einbindung der Datenbank (bereits teilweise umgesetzt)
- Verbindung der Formulare (bereits teilweise umgesetzt)
- Filterung Unsere Tiere (Tierart, ...)
- Umsetzung der PHP-Session (Nutzeranmeldung)
- Registrierung neuer Nutzer
- Fertigstellung der Projektdokumentation
- Voraussichtliche Implementierung von JavaScript (schwer abzuschätzen)



# 5. Zukunftspläne

Herausforderungen:

Lösungsansätze:



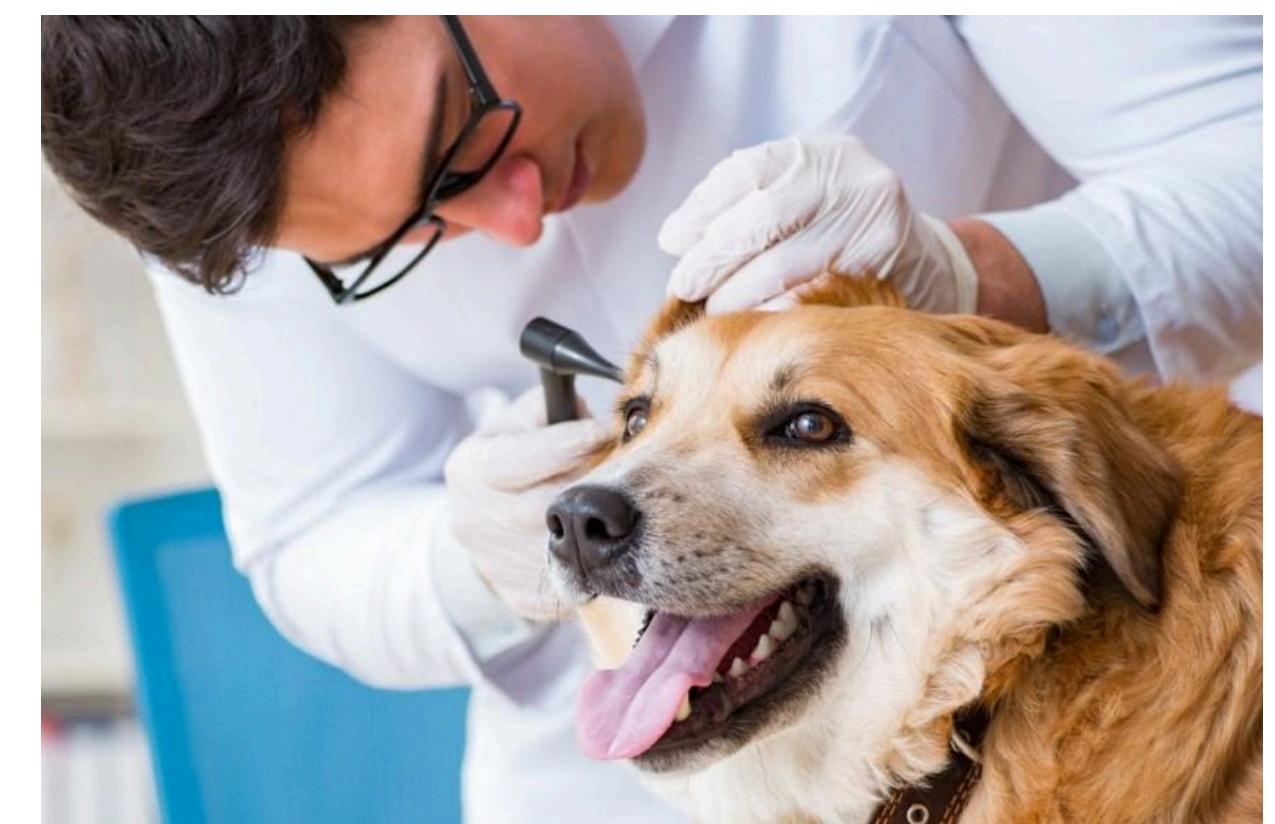
# 5. Zukunftspläne

## Herausforderungen:

- Geringer Lernzeitraum für die optimale Umsetzung des Gelehrten
- Teilweise schwer erreichbare Anforderungen an das Projekt in Bezug auf den Lernfortschritt
- Nicht thematisierte Inhalte, die bereits als umgesetzt vorausgesetzt werden (insbesondere JavaScript)

## Lösungsansätze:

- Nutzung zusätzlicher Ressourcen wie Tutorials
- Teamarbeit, um Wissen gezielt zu teilen
- Setzen von Prioritäten, um Funktionalitäten möglichst effizient umzusetzen



# 6. Wünsche zu Vertiefungen



# 6. Wünsche zu Vertiefungen

Welche Anforderungen sind im Projektverlauf entstanden, die Sie besprechen möchten?

Was möchten Sie gerne vertiefen?



# 6. Wünsche zu Vertiefungen

Welche Anforderungen sind im Projektverlauf entstanden, die Sie besprechen möchten?

- entspricht unser umgesetztes MVC Pattern den gestellten Anforderungen?

Was möchten Sie gerne vertiefen?

- relevante Themen zu Beginn der Vorlesung behandeln, um eine realistische Umsetzung zu gewährleisten (JavaScript)
- Vertiefung der Best Practices für die Verknüpfung von Front- und Backend
- Sicherstellung einer reibungslosen Kommunikation der Schnittstellen



# 6. Wünsche zu Vertiefungen

Technische Vertiefungen

1. Umsetzung von PHP Cookies und Sessions

2. Verwaltung der Nutzerrechte



# 6. Wünsche zu Vertiefungen

## Technische Vertiefungen

### 1. Umsetzung von PHP Cookies und Sessions

- sollen wir Session und Cookies verwenden oder nur eins von beidem?

### 2. Verwaltung der Nutzerrechte

- in der Nutzerrollentabelle (DB) Rechte festlegen?
- zählt der Login als mehrseitiges Formular (mit Session)?



# 7. Aktueller Projekstand

	Anforderung	Beispiel	
1	Mind. 6 Seiten	Zum Beispiel: Startseite, Über Uns, Shop, Kontakt, Impressum, Kontoverwaltung	<input checked="" type="checkbox"/>
2	Mind. 3 Unterseiten	Zum Beispiel unterteilt sich der Punkt Shop, in einem Untermenü, in Männer, Frauen, Kinder.	<input checked="" type="checkbox"/>
3	Einheitliche Navigation über alle Seiten	Sorgen Sie dafür, dass der Nutzer sich wohlfühlt, die Navigation sollte einheitlich gestaltet sein und sich nicht ständig ändern.	<input checked="" type="checkbox"/>
4	Navigation mit Untermenü (2. Ebene)	Zeigen Sie, dass Sie wissen, wie man ein Untermenü auf der Seite einbindet und darstellt.	<input checked="" type="checkbox"/>
5	<b>Sie verwenden das Model-View-Controller-Pattern.</b>	Sie haben eine Controller-Klasse, die ein dynamisches Routing durchführt. Die einzelnen Interaktionen der Nutzer werden auf Actions überführt. Controller und Model (Datenbankinteraktion, Geschäftslogik) sind klar getrennt.	<input type="checkbox"/>
6	mind. 3 Formulare (Login, Registrierung, Kontakt)	Zeigen Sie, dass Sie den Umgang mit Formularen verstanden haben. Formulare gibt bspw. zum Anlegen von Produkten, für die Registrierung, beim Login oder auch im Kontakt.	<input type="checkbox"/>
7	Sie folgen Clean-Code Konventionen.	Sie vermeiden doppelten Code durch Wiederverwendung und folgen auch anderweitig den <u>Clean Code Konventionen</u> .	<input type="checkbox"/>
8	Codestyle und Code-Dokumentation	Sorgen Sie dafür, dass Sie einen verständlichen Codestyle verwenden und dokumentieren Sie ausreichend den Code, um die Wartbarkeit zu gewährleisten.	<input checked="" type="checkbox"/>
9	Keine Verwendung von Frameworks	Die Verwendung von Frameworks wie jQuery oder Bootstrap ist nicht möglich.	<input checked="" type="checkbox"/>

10	<b>Projektdokumentation</b>	Die ausführliche Projektdokumentation ist als strukturierte HTML-Seite im Impressum verlinkt. Sie enthält auch Informationen zum Aufwand und der Arbeitsteilung im Team. Details finden Sie in den Folien zur Projektaufgabe.	<input type="checkbox"/>
11	Installationshinweise	Hinweise zur Installation und Kennwörter sind als Textdokument oder PDF beigelegt.	<input type="checkbox"/>
12	Datenbank-Export	Die Datenbank ist einschl. Beispieldaten als SQL-Export gut auffindbar in der Abgabe enthalten. Idealerweise beschreiben Sie den Import in der Installationsanleitung und wo die Datei(en) zufinden sind.	<input type="checkbox"/>
13	Test	Das Projekt wurde mit den Abgabedaten auf einem anderen Rechner installiert und mit verschiedenen Bildschirmauflösungen getestet.	<input type="checkbox"/>
14	Passende, abwechslungsreiche Gestaltung	Die inhaltliche und grafische Gestaltung mit Texten, Bildern, Grafiken und Tabellen sollte zum gewählten Thema passen und abwechslungsreich sein. Fließtext und sich wiederholende Elemente können als Blindtext, Dummy-Grafiken u.ä. ausgeführt sein, solange die prinzipielle Intention erkennbar ist.	<input checked="" type="checkbox"/>
15	<b>Interaktivität schaffen mit CSS/JavaScript</b>	Schaffen Sie Nutzerfeedbacks, Interaktivität. Zum Beispiel mit einem Submenu gesteuert durch CSS und Hover-Effekte oder durch Animationen mit JavaScript	<input type="checkbox"/>
16	<b>Quelltext Optimierung durch Metas,Alt,Title,Noscript. Sinnvolle Verwendung von HTML5 (div,nav,header,article,section ...)</b>	Schaffen Sie eine Barrierefreiheit für Bots und Hilfstoools. Nutzen Sie HTML5 Elemente und strukturieren Sie Ihren Code entsprechend.	<input checked="" type="checkbox"/>



# 7. Aktueller Projekstand

17	Einbindung von Bildern	Zeigen Sie den Umgang mit Bildern.	<input checked="" type="checkbox"/>
18	Responsive Webdesign (Verwendung von Gridlayouts, Flexbox, float, ...)	Sorgen Sie dafür, dass Ihre Webseite auch auf mobilen Endgeräten funktioniert. Denken Sie an wichtige Meta-Informationen, die Sie benötigen für korrekte Darstellung auf mobilen Endgeräten.	<input checked="" type="checkbox"/>
19	Animation (optional)	Verwenden Sie Animationen und Überblendungen (Transitions) für eine angenehme, interessante User Experience.	<input checked="" type="checkbox"/>
20	Stilangabe für Druckausgabe (optional)	Erstellen Sie für mind. eine geeignete Seite (z.B. Produktübersicht oder -detail) gesonderte Stile für die Ausgabe auf dem Drucker. Allgemein für DWP	<input type="checkbox"/>
21	Formulare werden mittels PHP/JavaScript/HTML behandelt und Fehler angezeigt	Nutzen Sie die Möglichkeiten von HTML, JavaScript und PHP zur Validierung der Felder und aller anderen Kommunikationsmöglichkeiten mit dem Backend. Prüfen Sie, ob alle Felder ausgefüllt sind bzw. ob alle Felder mit validen Werten gesetzt sind und geben Sie entsprechende Fehlermeldungen aus. Blenden Sie ggf. mit JavaScript Formular-Felder ein und aus. Aktivieren Sie den Absenden-Button erst dann, wenn alle Felder korrekt gefüllt sind. Nutzen Sie JavaScript für komplexere semantische Überprüfungen der Eingaben. Befüllen Sie wenn möglich Eingabefelder vor, indem Sie geeignete Werte aus der Datenbank abfragen. Vermeiden Sie Fehlermeldung in PHP durch Zugriff auf Schlüssel in einem Array, die nicht vorhanden sind. Escapen Sie Nutzerinput, bevor Sie ihn für Datenbankabfragen verwenden. Validieren Sie per PHP immer die Daten, die der Nutzer an den Server sendet.	<input type="checkbox"/>
22	Nutzeranmeldung	Es ist möglich, sich mit einem Account anzumelden, es gibt einen Login dafür.	<input type="checkbox"/>
23	Rollen-/Rechtekonzept	Sie berücksichtigen mindestens 3 verschiedene Nutzerarten, die bezüglich der Datenschnittstellen Create-Read-Update-Delete unterschiedliche Rechte (auf unterschiedlichen Tabellen) haben. Dies wird im Code überprüft und erzwungen.	<input type="checkbox"/>

24	Nutzerregistrierung (Vorname, Nachname, E-Mail, Passwort, Telefonnummer, ...)	Es ist möglich, sich ein Account für den Login anzulegen.	<input type="checkbox"/>
25	Funktionsbereitstellung mit und ohne JavaScript	Stellen Sie sicher, dass Sie JavaScript als Optimierung sehen und dass die Seite auch ohne JavaScript ausreichend funktioniert, so dass alle Inhalte und Funktionen erreicht werden.	<input type="checkbox"/>
26	Verwenden Sie Ajax-Methoden an geeigneten Stellen.	Sorgen Sie dafür, dass wenig Traffic auf der Seite entstehen, in dem Sie Formulare mittels asynchronem JavaScript absenden. Denken Sie aber immer daran, der Nutzer kann falsche Daten übermitteln, validieren Sie daher auf PHP-Seite und übermitteln Sie Fehlermeldungen. Zeigen Sie, dass Sie wissen wie Ajax funktioniert an mindestens einer Stelle (bspw. Registrierungsformular).	<input type="checkbox"/>
27	Seiten bauen sich dynamisch anhand von Daten aus der Datenbank aus (mind. eine Liste von Daten)	Laden Sie Inhalte aus einer Datenbank, zeigen Sie dies an mind. einer Liste. Zeigen Sie dabei den Umgang mit SQL Produktansichten o.ä. können nach mind. 3 Parametern (z.B. Größe, Preis, Datum, Autor,...) gefiltert und sortiert werden. Filterparameter müssen logisch entweder mit UND oder ODER verknüpft werden können.	<input type="checkbox"/>
28	Nutzen Sie PHP-Sessions und Cookies an geeigneter Stelle zur Persistierung eines Verbindungsstatus.	Sie bieten bspw. einen Warenkorb an, den der Nutzer immer sieht, wenn er eingeloggt ist.	<input type="checkbox"/>



# 7. Aktueller Projekstand

29	Seiten können Inhalte dynamisch/asynchron über JavaScript nachladen	Laden Sie Seiteninhalte dynamisch über JavaScript nach, zum Beispiel könnten Sie durch den Klick auf einen “Load-More” Button in einer Liste, die nächsten Inhalte dafür laden und darstellen. Anders könnten Sie den Warenkorb eines Shops in einem Popover anzeigen und den Inhalt mittels Ajax laden. Sie können auch Filter hinzufügen und entfernen, oder zeigen Sie Suchergebnisse, Vorschläge an.	<input type="checkbox"/>
30	Fehlerbehandlung auch für nicht bekannte Seiten (404 Error)	Fangen Sie Fehler ab, vor allem wenn versucht wird auf Seiten zu navigieren die nicht vorhanden sind.	<input checked="" type="checkbox"/>
31	Es ist eine Datenbank angebunden	Binden Sie eine Datenbank an Ihr PHP-Script an, ideal über PDO oder mysqli. Die Datenbank sollte ausreichend komplex sein.	<input checked="" type="checkbox"/>
32	Daten aus der Datenbank werden gelesen und geschrieben.	Bsp.: Laden Sie Inhalte wie Nutzer, Login-Daten oder Produkte aus der Datenbank. Schreiben Sie Inhalte wie Nutzer, Login-Daten oder Produkte in eine Datenbank.	<input type="checkbox"/>
33	Sie verwenden standardkonformes HTML	Sie haben Ihre dynamisch generierten Seiten mittels <u><a href="#">HTML Validator</a></u> überprüft.	<input type="checkbox"/>



# Vielen Dank für Ihre Aufmerksamkeit



PfotenDesign