

# Task 1: Homepage

- Create the SCSS styles for the Homepage.
- Homepage ist NOT static. It is generated by Payload backend as follows: Under „Pages“ there is a page with slug == home. Inside this page, we use Archive Collection:
  - NEWS are posts with Category „NEWS“
  - ARTICLES are posts with Category „ARTICLE“
  - LITERATURE are products with product categories (see later)
  - VIDEOS are posts from the video collection (see later): Here, the featured image should be clickable and start the video.
  - LATEST are the latest posts (of any category), latest videos and latest products
- Please use Tailwind CSS and shadcn/ui like described here: <https://www.youtube.com/watch?v=sI2nXEeyYlw>
- Also make sure that localization of FRONTEND is supported: <https://payloadcms.com/localization>
- **SUMMARY: The main task here is to develop the CSS styles for the individual blocks. Please make sure that the styles are re-usable. Important question: How to apply the styles to individual blocks on the Homepage? We must link the blocks in Payload backend to the style in the frontend. How do you want to do this? We should discuss this point together.**
- Page should look like this:

Logo

Social media buttons

Unsere Treffen

Beiträge

Video

Audio

Apps

Literature

Sign up

Login

NEWS

75%

LATEST

25%

NEWS

This is the header for the first item of category NEWS

NEWS

Header for post 2

NEWS

Header for post 3

Title

27 July

Title

27 July

Title

27 July

Title

27 July

Title

27 July

ARTICLES

ARTICLE

Header for article 1

ARTICLE

Title of article

27 July

ARTICLE

Title of article

27 July

ARTICLE

Title of article

27 July

LITERATURE

CATEGORY

Title of product

CATEGORY

Title of product

CATEGORY

Title of product

CATEGORY

Title of product

VIDEOS

16:9

VIDEO

Video title

27 July

16:9

VIDEO

Video title

27 July

16:9

VIDEO

Video title

27 July

16:9

VIDEO

Video title

27 July

## Footer

Logo

"Vielmehr seid ihr zu dem Berg Zion gekommen und der Stadt des lebendigen Gottes, dem himmlischen Jerusalem ..." (Hebr. 12:22). Die Gemeinde, in der unser Gott und Vater angebetet werden möchte, ist nicht nur eine Versammlung von Christen mit einem bestimmten Namen. Wir haben den Wunsch, heute das himmlische Jerusalem zu sein und dort unseren Vater-Gott im Geist und in der Wirklichkeit anzubeten. Alle, die diesen Wunsch mit uns teilen, sind herzlich eingeladen!

## Task 2: Conferences collection

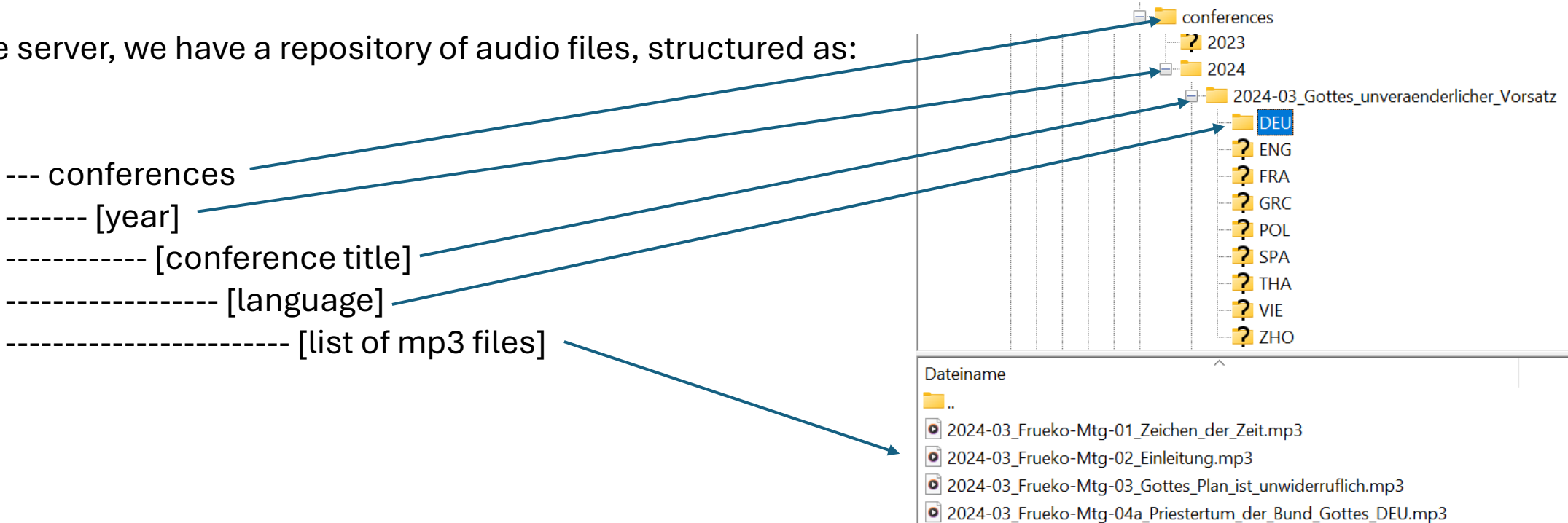
- Implement a new **collection** „Conferences“ in Payload by extending the „posts“ collection (which already exists in the Payload demo)
- In addition to the existing fields, we must add: **location as text** and **date as calendar** to select month/year
- The „Conferences“ collection can then be added to a page by the backend admin via **Page > Content > Layout > Archive**
- Once the admin added a „Conferences“ collection to a page, the frontend shall display the collection as seen below
- For title, location and date implement alphabetical sort functions
- Also implement a search for title
- Add pagination to show 10 conference items per page

	Sort ABC		Sort ABC	Sort date
Image	This is a conference title, which might be a bit long		Stuttgart	April 2024
Image	This is a conference title, which might be a bit long		Stuttgart	April 2023
Image	This is a conference title, which might be a bit long		Stuttgart	April 2022
Image	This is a conference title, which might be a bit long		Stuttgart	April 2021
Image	This is a conference title, which might be a bit long		Stuttgart	April 2020

## Task 3: Audio Playlist block

- **Implement a new layout block called „Audio Playlist“**
- Audio Playlist requires the following fields:
  - Image as upload
  - Title as text
  - Description as text
  - Path to the folder structure on the server

- On the server, we have a repository of audio files, structured as:



- For example, the path looks like:

path: 2024/2024-03\_Gottes\_unveraenderlicher\_Vorsatz

- This means, the Playlist block should have a combobox (or similar) that lets the user select a path within the „conferences“ root on the server
- In the FRONTEND, the Playlist block generates the following block. **It should look like this:** <https://app.theheavenlyjerusalem.org/record/217>

Image

# Title

Description

38 Recordings

Number of items

Play

Dropdown box

All

German

English

French

Greek

Languages of each conference, coming from the server (subfolders of conference):

Server repository:

2024-03\_Gottes\_unveraenderlicher\_Vorsatz

DEU

ENG

FRA

GRC

POL

SPA

THA

VIE

Icon	Title of audio which might be a little long (comes from server) Language		58:39
Icon	Title of audio which might be a little long (comes from server) Language		58:39
Icon	Title of audio which might be a little long (comes from server) Language		58:39

- These items are populated depending on the selection of the combobox. The user can select a language in the combobox, then the files within the respective language folder on the file server will be displayed.
- The content of the combobox is populated via a script on the server. Right now, we have a PHP script: <https://himmlisches-jerusalem.de/audio.php?type=conference> for conference audio files.
- Please transform this PHP file to TypeScript or Javascript and include it in the app.
- Do not change the JSON output format because it is also used in an Android app.
- When the user clicks on the play button, an mp3 player should appear to play the audio.

### IMPORTANT:

- The Audio Playlist block can be used in **any page, post or other collection**, not only in conference collection (default behaviour in Payload)
- BUT: If the Playlist block is used within a conference collection item, then the fields **image**, **title**, **location**, and **date** should be pre-filled based on the data in the conference item.
- If the Playlist block is used in another collection type (not conference), the fields image, title, location, and date are empty and required.

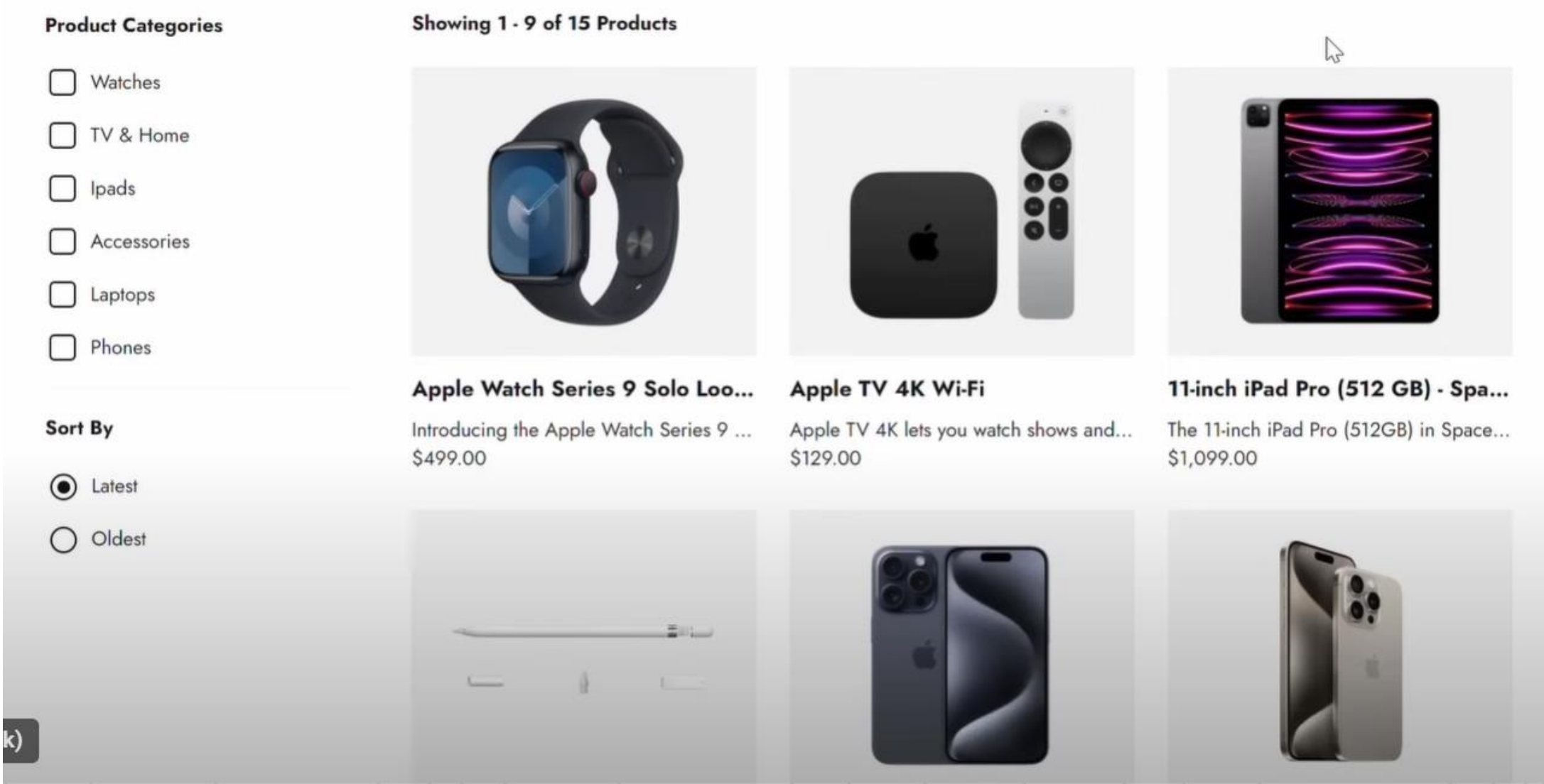


# Task 4: Video collection

- Implement a new **collection** „Videos“ in Payload by extending the „posts“ collection (which already exists in the Payload demo)
- In addition to the existing fields, we must add: **youtube-url as text** and **mobile-url as text**
- The „Videos“ collection can then be added to a page by the backend admin via **Page > Content > Layout > Archive**
- Add a voting mechanims for videos as shown here: <https://www.youtube.com/watch?v=121FJ9mov-4>
- In the FRONTEND, we show video cards with votes
- Video thumbnail image should be 16:9 aspect ratio
- Once the user clicks on a video, a video player opens up and plays the video from the youtube-url.
- The mobile-url is not used by the Website (you can ignore it for now)

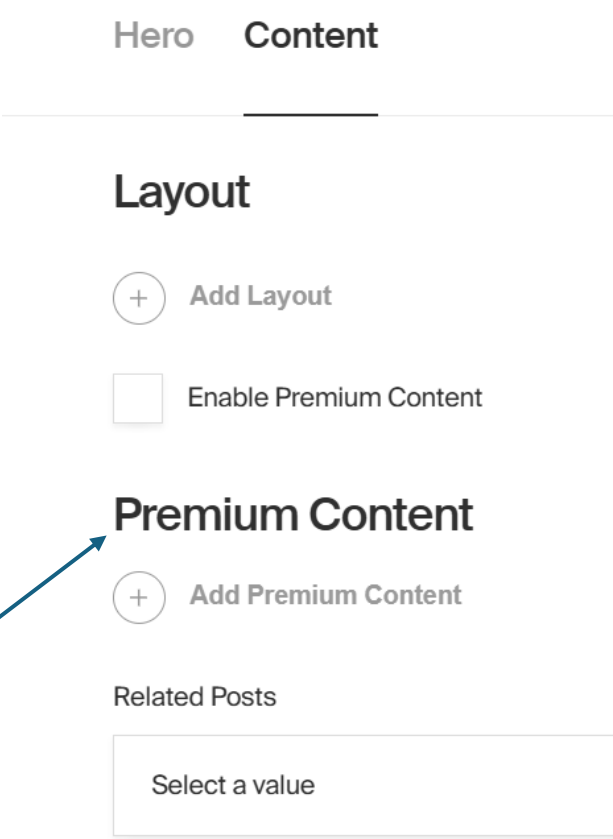
# Task 5: Ecommerce

- Payload provides a fully-functional ecommerce template including cart, checkout, stripe for payments etc. However, the ecommerce template for Payload 3.0 is not fully ready yet. But all the core functions (cart, checkout, stripe...) are already fully functional now. You can obtain the source code from here: <https://github.com/payloadcms/payload/tree/feat/ecommerce-3>
- Add „Products“ collection and „Product Categories“
- Please use the ecommerce functions from the Payload git repository. Don’t implement it yourself.
- For styling the FRONTEND, please use this tutorial: <https://www.youtube.com/watch?v=3JUsg-WsU9o&t=213s>
- Careful: This tutorial is made with Payload 1.0! We are using Payload 3.0!
- Simply clone the styles of this tutorial to style our shop page (including all the features presented in the tutorial)
- For stipe payments, I already have an account with stripe. Just contact me once you need it.
- Here is a screenshot from the product styling of the tutorial, including sidebar for filtering. Please use this example to style our shop:



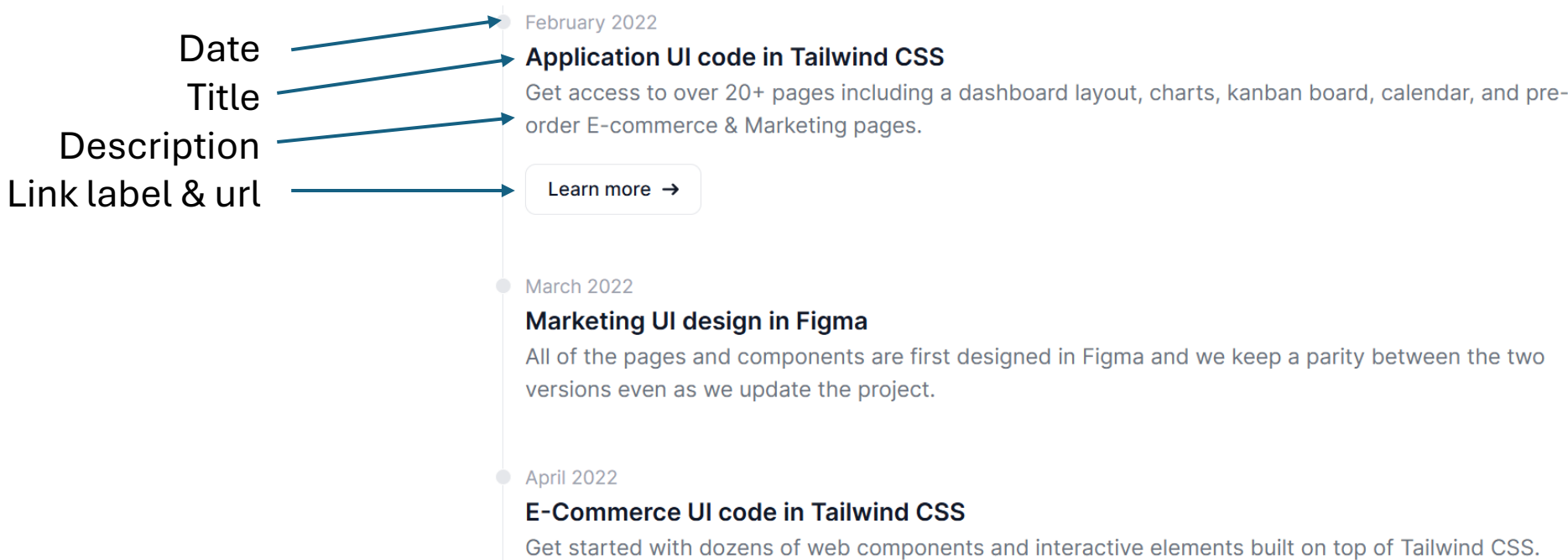
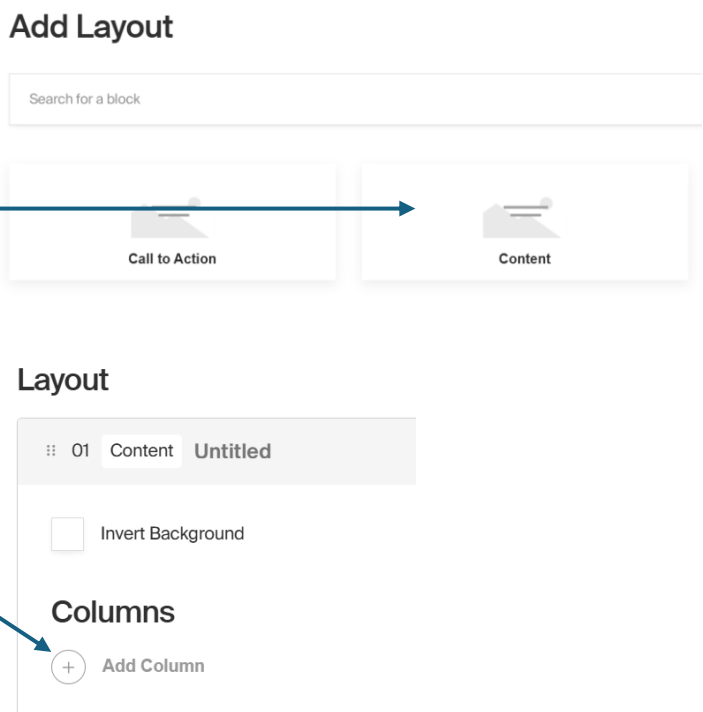
# Task 6: User role management

- We need the following user roles: admin, customer, priest, uploader, none (= guest)
- When a user registers on the website (not in shop), he has no role (= guest)
- The admin must then assign a role manually in the Payload backend
- If the user purchases something in the shop, he is a „customer“ automatically
- A user can have multiple roles, for example, customer AND priest
- There will be role-based access: For example, the page „Fellowships“ can only be viewed if the user has role == priest. Payload already provides the functionality for „Premium content“. We will use this functionality.



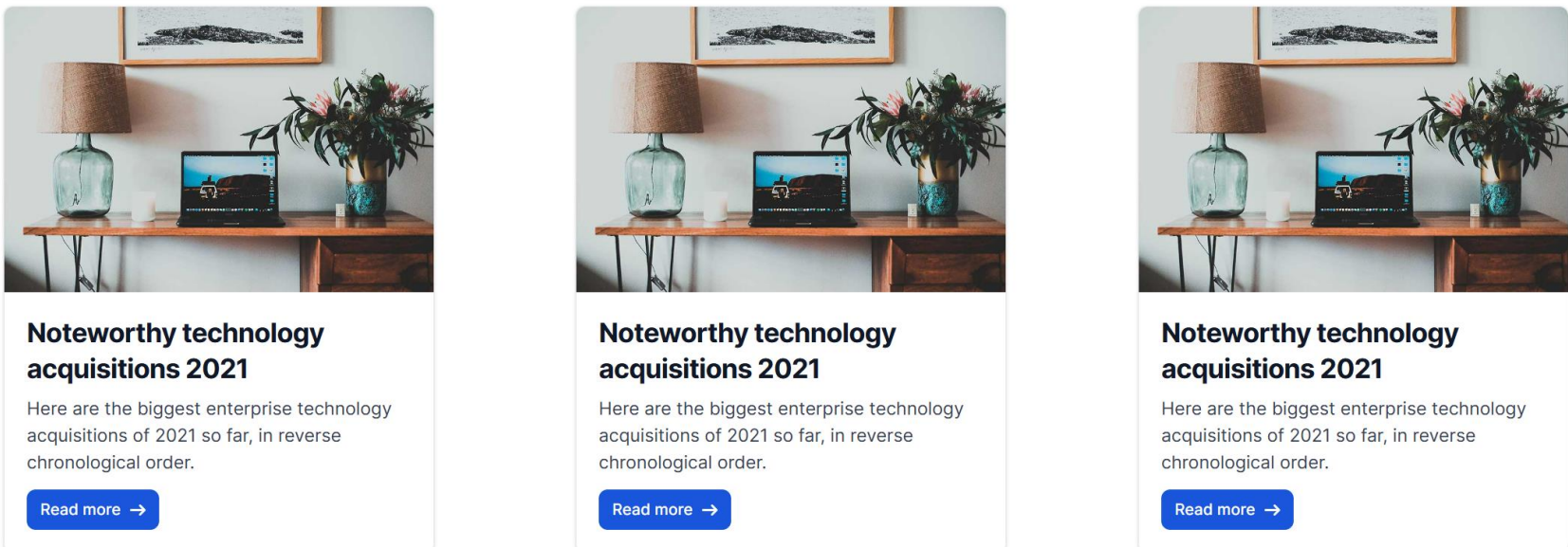
# Task 7: Timeline

- Implement a new block called „**Timeline**“, which can be added to any page
- This block is similar to the existing Content block
- In the Timeline block, the admin can add „columns“
- Each column is an entry in the timeline
- Each column has the following fields:
  - Date as text
  - Title as text
  - Description as text
  - Link label as text
  - Link url as text
- In the FRONTEND, these fields shall generate the following UI (you can get the HTML code of this UI here: <https://flowbite.com/docs/components/timeline/#default-timeline>):



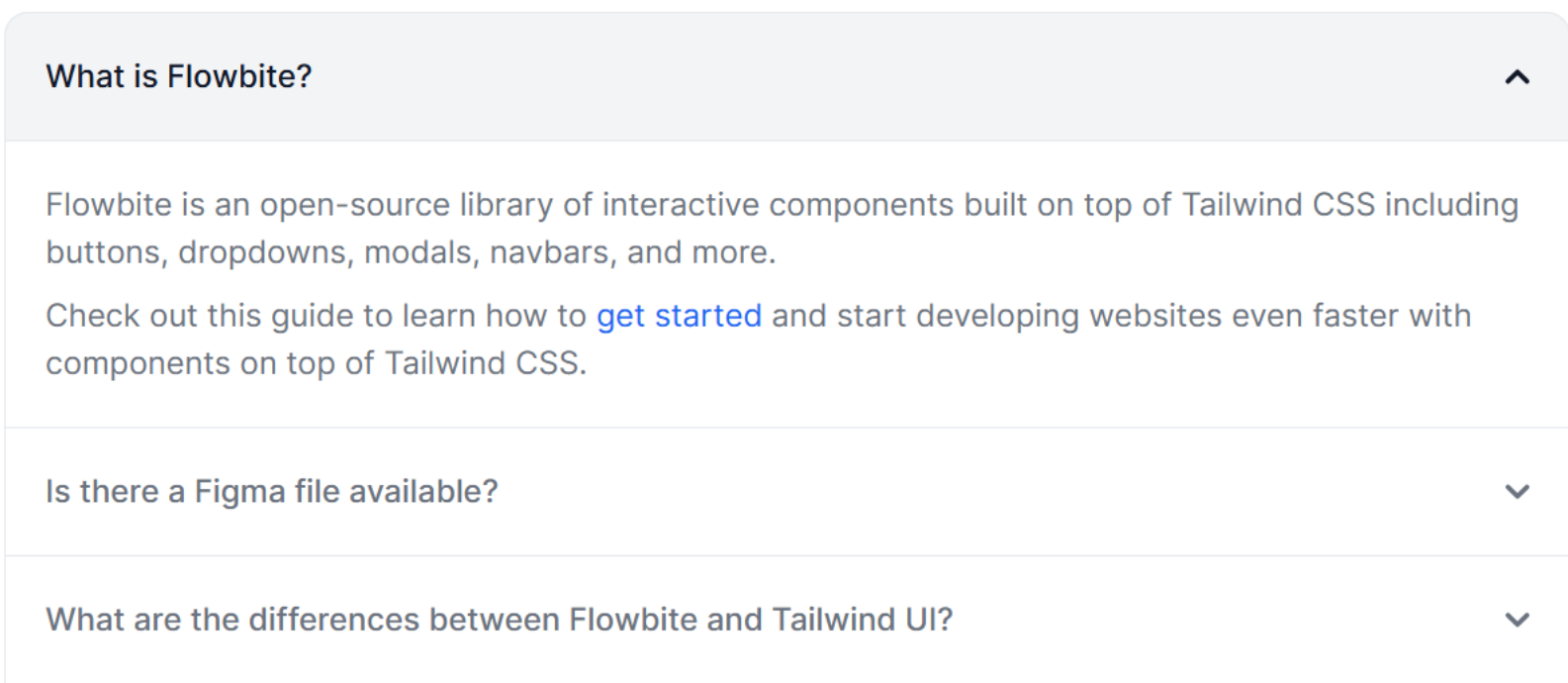
# Task 8: Chart Download

- Implement a new block called „**Chart Download**“, which can be added to any page
- Fields:
  - Image as upload
  - Title as text
  - Description as text
  - Link url as text with label „Download →“
- Link url is a path to a PDF file on the server (for example [this](#)).
- In the FRONTEND, these fields shall generate the following UI (you can get the HTML code of this UI here: <https://flowbite.com/docs/components/card/#card-with-image> )
- When the user clicks „Download“, the PDF file is downlaoded



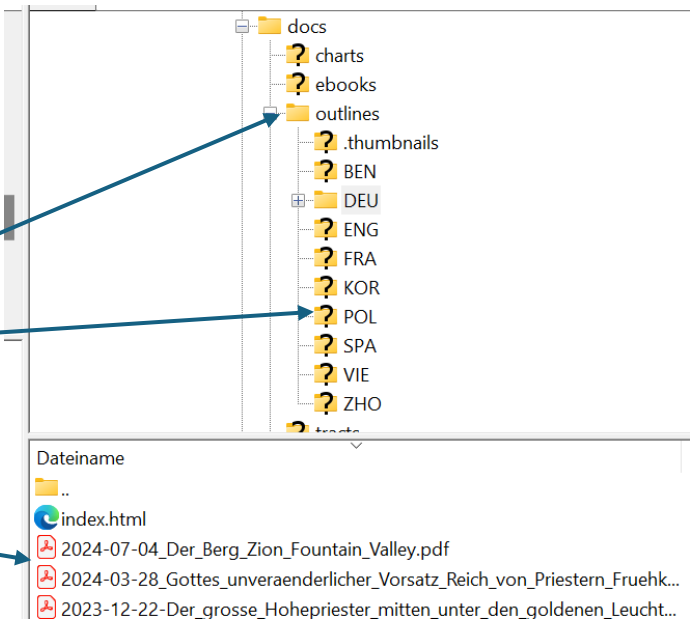
Task 9: FAQ block

- Implement a new block called „**FAQ**“, which can be added to any page
- This block is similar to the existing Content block
- In the FAQ block, the admin can add „columns“
- Each column is an entry in the FAQ
- Each column has the following fields:
  - Question as text
  - Answer as text (must allow hyperlinks)
- In the FRONTEND, we use this Accordion (<https://flowbite.com/docs/components/accordion/#default-accordion>):



Task 10: Outline block

- Implement a new block called „**Outlines**“, which can be added to any page
- In principle it is the same block as Playlist block, except for PDF files (not audio)
- Admin can select a path in the file system, for example: **docs/outlines**
- Inside this folder, we have many languages (like in Playlist)
- Inside each language, we have PDF files
- We have a PHP script which reads the contents of this folder: <https://himmlisches-jerusalem.de/docs.php?type=outline>
- Please do not change the JSON format
- In JSON, select all entries with „album == outlines“ and read the „title“



- In the FRONTEND, we use a similar layout like Playlist:

Image

Outlines

Description

38 Items

Number of items

Dropdown box

All

German

English

French

Greek

Server repository:

docs

charts

ebooks

outlines

.thumbnails

BEN

DEU

ENG

FRA

KOR

POL

SPA

VIE

ZHO

Icon

2024-07-04-Title1\_which\_can\_be\_long

Language

Download

Icon

2024-07-04-Title2

Language

Download

Icon

2024-07-04-Title3

Language

Download

Addition to Task 3:

To add audio, we have 2 use cases:

**Use case 1:**  
We upload audio manually, for example with FileZilla (not via backend/frontend). Once new audio has been uploaded and the user refreshes the page, we query the new audio files via the new js script (I have already sent you the old PHP script).

**Use case 2:**  
We upload audio via an upload area (drag-drop) on frontend. This means: If a user with role >= „Uploader“ is signed in, he can see an upload area here:

Image

Title

Description

38 Recordings

Number of items

Play

Dropdown box

All

German

English

French

Greek

Languages of each conference, coming from the server (subfolders of conference):

Server repository:

2024-03\_Gottes\_unveraeenderlicher\_Vorsatz

DEU

ENG

FRA

GRC

POL

SPA

THA

VIE

This upload area is only visible for users who are signed in with role >= Uploader

Drag and Drop assets here

Or

Browse

Icon

Title of audio which might be a little long (comes from server)

Language

58:39

Icon

Title of audio which might be a little long (comes from server)

Language

58:39

Icon

Title of audio which might be a little long (comes from server)

Language

58:39

For exampe, if in the backend, the path 2024/2024-03\_Gottes\_unveraenderlicher\_Vorsatz is defined and the „Uploader“ selected „German“ as language in the dropdown box, all audio will be stored to 2024/2024-03\_Gottes\_unveraenderlicher\_Vorsatz/DEU on the server.