

Guide for Managing AMORE Website

Ingebjørg A. Iversen

Table of contents

1	Introduction	2
1.1	Who this guide is for:	3
1.2	Git, Shinyapp, and netlify	3
1.2.1	Git (Version Control & Collaboration)	3
1.2.2	GitHub repository	3
1.2.3	ShinyApps.io (Shiny App Hosting)	3
1.2.4	Netlify (Website Hosting)	4
1.3	RStudio and Quarto	4
1.3.1	The Different Languages	4
1.4	Nettskjema	5
2	Getting started	5
2.0.1	Code block types throughout this guide and what they mean in the document	5
2.1	Prerequisites	6
2.2	Three Access Methods to Git	6
2.2.1	Method 1: Collaborator Access (For Core Lab Members)	7
2.2.2	Method 2: Fork & Pull Request Workflow (For Contributors Without Direct Access)	7
2.2.3	Method 3: RStudio Git Integration (Recommended for R Users)	8
2.2.4	Save and Push Your work to GitHub with command line Git (Method 1 and 2)	9
2.3	Dependencies	11
3	Project structure and file explanations	12
3.1	Root files	13
3.1.1	index.qmd	14
3.1.2	__quarto.yml	14
3.1.3	LICENSE	14
3.1.4	README.md	15
3.1.5	.gitignore	15
3.1.6	.node-version	17
3.1.7	Package.json	17
3.1.8	netlify.toml	18
3.1.9	netlify-setup-quarto.sh	18
3.1.10	install-quarto.sh	19
3.2	.quarto/ folder	20

3.3	assets/ folder	20
3.4	pages/ folder	21
3.5	LMAs/ folder	21
3.6	docs/ folder	22
3.7	scripts/ folder	23
3.8	shiny-meta/ folder	23
3.8.1	How app.R works:	24
3.9	__site/ folder	25
4	Styling system SCSS	26
4.0.1	Overview	26
4.0.2	Structure of styles.scss	27
5	Adding new projects	27
5.1	Getting Started:	28
5.1.1	What do you fill in?	28
5.1.2	Format rules	29
5.2	Build the template	30
5.2.1	Mandatory sections	30
5.2.2	Optional sections	45
5.3	Previewing your work	52
6	Troubleshooting	52
7	Website maintenance	53

Contributors to this guide document:

- Ingebjørg Anjadatter Iversen
- Alina Sartorius
- Heemin Kang

1 Introduction

AMORE is a website created by Ingebjørg Anjadatter Iversen for the behavioral Neuroendocrinology Lab to host living meta-analysis for oxytocin research investigating biobehavioral outcomes.

This document instructs the management of AMORE webpage.

AMORE was built using RStudio and Quarto. Git is used for version control, collaboration and hosting website in GitHub repository. Netlify is deploying the website and accessing the script from the github repository. Shinyapp host the app.R (backend to the directory hosting living meta-analysis on AMORE). Nettskjema delivers contact forms used as medium between platform users and the AMORE team.

1.1 Who this guide is for:

- Lab members adding new LMA projects
- Website administrators maintaining platform

1.2 Git, Shinyapp, and netlify

The AMORE platform consists of three interconnected services:

1.2.1 Git (Version Control & Collaboration)

Repository: iaiversen/AMORE-webpage.

Stores all source code. Tracks changes with commit history. Enables collaboration through pull requests.

Important: Never commit sensitive data (API keys, passwords)

1.2.2 GitHub repository

Repository: AMORE-webpage

Repository URL: <https://github.com/iaiversen/AMORE-webpage>

Created by github account: iaiversen (Ingebjørg Anjadatter Iversen)

Collaborators: dsquintana (Daniel quintana), AMSartorius (Alina Sartorius), HeeminK (Heemin Kang)

1.2.2.1 Repository License

AMORE-webpage repository has the MIT license. It gives open access. The LICENSE file is in the root folder (AMORE-webpage).

1.2.3 ShinyApps.io (Shiny App Hosting)

Hosts the Living Meta-Analysis directory (app.R) Free tier limitations: limited active hours, connection timeouts URL: <https://meta-oxytocin.shinyapps.io/shiny-meta/> Embedded in Living_meta-analysis_Directory.qmd via iframe

1.2.3.1 Shinyapp.io login and important

1.2.4 Netlify (Website Hosting)

Automatically deploys from GitHub Builds site from `_site` folder after Quarto render Domain: `amore-project.org` Configuration in `netlify.toml`

Deployment workflow:

Make changes locally in RStudio. Test by rendering (`quarto render` or Render button) Commit and push to GitHub. Netlify automatically rebuilds and deploys. For Shiny app: Deploy separately using `rsconnect::deployApp()` or push publish button in the right top corner of the terminal.

1.2.4.1 Netlify login and important

two-step authenticator connected to

1.3 RStudio and Quarto

RStudio is your primary development environment. Install:

R (version 4.0.0+) from CRAN RStudio Desktop from Posit Quarto CLI - Usually bundled with RStudio, or install from `quarto.org`

1.3.1 The Different Languages

The AMORE website is built using multiple programming languages and frameworks that work together: Quarto is multilingual, Just specify in the code chunk what language is used.

Languages in use:

- Quarto Markdown (`.qmd`): Primary content files for pages
- SCSS/CSS: Styling and responsive design
- JavaScript: Interactive functionality and client-side behavior
- R: Shiny app backend, data processing, and setup scripts
- HTML: Embedded within Quarto files for custom components Shell scripts: Deployment and automation tasks

1.4 Nettskjema

Editorial right for nettskjema forms:**

- ingebjai@uio.no
- danielqu@uio.no

The users reach out to the AMORE team through Nettskjema. They find the nettskjema through links on the AMORE website or directly on the contact page.

The Nettskjema forms are:

1. Contact form AMORE

- Contact button on home page (index.qmd) transfers platform users to this form.
- Redirects to the other three forms (Propose your project, Update an existing living meta-analysis or Get help or share feedback).

2. Propose your project

3. Get help or share feedback

4. Update an existing living meta-analysis

Emails of responses to any of these forms goes to:

- ingebjai@student.sv.uio.no
- daniel.quintana@psykologi.no

NB! Dan streamlines the nettskjema responses to the AMORE expert steering committee for evaluation of projects or other inquiries.

2 Getting started

This section guides you through setting up your development environment, providing the essential setup instructions.

2.0.1 Code block types throughout this guide and what they mean in the document

Throughout this guide, you'll see code blocks with different labels that indicate where or how to run the code.

2.0.1.1 `{r}` - R block

This code block is used to indicate that the content of the code block belongs to a script or file inside the AMORE.Rproj. It does not necessarily mean that the language used is R, since quarto is multilingual and uses multiple languages in the same files.

2.0.1.2 `{bash}` - bash block

The bash code block indicates that the content is for a command line interface. Bash, powershell and RStudio Terminal are all command line interfaces. (I personally use Windows Powershell as my terminal).

The blocks contains git commands. Git commands work in Bash, Powershell and RStudio Terminal.

Shift+Alt+T or Shift+Option+M on mac opens Terminal in RStudio.

2.0.1.3 `{html}` and `{scss}` blocks

Html or Scss code block indicates that the content in the code block is written using the specified language. The languages are used across different files and scripts all within the AMORE.Rproj project.

2.0.1.4 Quick reference:

- See `{r}` `{html}` `{scss}` → Code belongs inside AMORE.Rproj and if html or scss indicates the language used.
- See `{bash}` → Open Terminal (Git Bash, Windows powershell, or RStudio Terminal)

2.1 Prerequisites

- R (Version 4.0.0 or higher) - programming language
- Rstudio - Integrated Development Environment (IDE) for R.
- Quarto - Document publishing system for creating websites or reports
- Github account
- Git version installed on your computer (not browser version).

2.2 Three Access Methods to Git

Choose the method that matches your access level and preferred workflow:

2.2.1 Method 1: Collaborator Access (For Core Lab Members)

If you are a direct collaborator with write access:

```
{bash}
#| label: collaborator-initial-setup
#| description: Clone repository and install dependencies for collaborators
#| eval: false
#| echo: true

# Clone the repository directly
git clone https://github.com/iaiversen/AMORE-webpage.git
cd AMORE-webpage

# Install R dependencies
Rscript Setup.R
```

Daily Workflow:

```
{bash}
#| label: collaborator-daily-workflow
#| description: Standard git workflow for collaborators with write access
#| eval: false
#| echo: true

# Get latest changes before starting work
git pull origin main

# Make your changes
# ... edit files ...

# Stage your changes
git add .

# Commit with descriptive message
git commit -m "Add new project: [Project Name]"

# Push directly to main repository
git push origin main
```

To check if you're a collaborator: - Visit: <https://github.com/iaiversen/AMORE-webpage/settings>
- If you can access Settings, you're a collaborator - If you get "404 Not Found", use Method 2 below

2.2.2 Method 2: Fork & Pull Request Workflow (For Contributors Without Direct Access)

If you are NOT a direct collaborator:

Initial Setup (One Time):

Step 1: Fork the Repository 1. Go to <https://github.com/iaiversen/AMORE-webpage> 2. Click the Fork button (top right corner) 3. This creates your own copy at: [https://github.com/\[YOUR-USERNAME\]/AMORE-webpage](https://github.com/[YOUR-USERNAME]/AMORE-webpage)

Step 2: Clone YOUR Fork

```
{bash}
#| label: fork-clone-setup
#| description: Clone your forked repository to your local machine
#| eval: false
#| echo: true

# Clone your fork to your computer
git clone https://github.com/[YOUR-USERNAME]/AMORE-webpage.git
cd AMORE-webpage
```

Step 3: Link to Main Repository

```
{bash}
#| label: fork-upstream-setup
#| description: Link your fork to the original repository as upstream
#| eval: false
#| echo: true

# Add the original repository as "upstream"
git remote add upstream https://github.com/iaiversen/AMORE-webpage.git

# Verify your remotes
git remote -v
# Should show:
# origin      https://github.com/[YOUR-USERNAME]/AMORE-webpage.git (your fork)
# upstream    https://github.com/iaiversen/AMORE-webpage.git (main repo)
```

2.2.3 Method 3: RStudio Git Integration (Recommended for R Users)

****Works for both Method 1 and Method 2 users****

RStudio has built-in Git integration that simplifies the workflow. This method is recommended if you're comfortable working in RStudio.

2.2.3.1 Cloning the Repository (First Time Setup)

****Important:**** The AMORE repository already contains an R project file ('AMORE.Rproj'). When you clone the repository, RStudio will automatically recognize and open this existing project—you won't create a nested project.

****Steps:**** 1. In RStudio: ****File → New Project → Version Control → Git**** 2. ****Repository URL:**** - ****If collaborator (Method 1):**** 'https://github.com/iaiversen/AMORE-webpage' - ****If using fork (Method 2):**** 'https://github.com/[YOUR-USERNAME]/AMORE-webpage' 3. ****Project directory name:**** Leave as 'AMORE-webpage' (or rename if desired) 4. ****Create project**

as subdirectory of:** Choose where to save it on your computer (e.g., ‘Documents/’) 5. Click ****Create Project****

****What This Does:**** - Downloads all files from the GitHub repository - Opens the existing ‘AMORE.Rproj’ project file - Automatically connects Git for version control - Sets up your working directory

****Result:**** You’ll have one AMORE project folder with Git already connected—ready to work with!

2.2.3.2 Using Git in RStudio

Once set up, you’ll see a ****Git tab**** in RStudio (usually upper-right pane):

****To commit and push changes:**** 1. Modify files in your project 2. ****Git tab:**** Check boxes next to changed files 3. Click ****Commit**** 4. Write a descriptive commit message 5. Click ****Commit**** button 6. Click ****Push**** to send changes to GitHub

****To pull latest changes:****

- Click the ****Pull**** button (blue down arrow) in the Git tab

****Note:**** If using Method 2 (fork workflow), you’ll need to set up the upstream remote in RStudio Terminal:

```
{bash}
git remote add upstream https://github.com/iaiversen/AMORE-webpage.git
```

2.2.3.3 Opening the Project Later

After initial setup: - File → Open Project → Navigate to **AMORE.Rproj** - Or double-click **AMORE.Rproj** in your file system

RStudio will remember the Git connection automatically.

2.2.4 Save and Push Your work to GitHub with command line Git (Method 1 and 2)

After you are finished working on the files in the project, you must push it to the GitHub repository so the changes are integrated on the website.

Important: Always Pull Before You Push

Critical: Before pushing your changes, always pull the latest updates from the repository. This prevents conflicts when multiple people are working on the website simultaneously.

Complete Git Workflow

1. Pull Latest Changes First

```
{bash}
#| label: git-pull-latest
#| description: Pull latest changes from main repository before making modifications
#| eval: false
#| echo: true

# In RStudio Terminal or Git Bash
git pull origin main
```

Why this matters: Someone else may have added or updated projects since you last synced. Pulling first ensures you're working with the most current version.

2. Stage Your New File

```
{bash}
#| label: git-add-files
#| description: Stage files for commit using various git add options
#| eval: false
#| echo: true

# Add ALL changes in current directory and subdirectories
git add .

# Add specific file from specific folder (using LMAs as example folder)
git add LMAs/yourfile.qmd

# Add all files in a specific folder (using LMAs as example folder)
git add LMAs/

# Add multiple specific files
git add LMAs/file1.qmd LMAs/file2.qmd

# Add all .qmd files
git add *.qmd
```

3. Commit Your Changes

```
{bash}
#| label: git-commit-changes
#| description: Commit staged changes with descriptive message
#| eval: false
#| echo: true

git commit -m "Add new LMA: [Brief but specific description]"
```

4. Push to GitHub

```
{bash}
#| label: git-push-to-github
#| description: Push committed changes to GitHub repository
```

```
#| eval: false
#| echo: true

git push origin main
```

2.3 Dependencies

Install all required packages:

Open `Setup.R` and run the entire script.

```
{r}
#| label: Setup script for dependencies
#| description: copy paste into r console
#| eval: false
#| echo: true

#You need to be in the root folder AMORE
source("scripts/Setup.R") # write this in console
# Or just find the script yoursekf and run it.
```

The `setup.R` script installs:

Core packages:

- `rmarkdown`, `knitr`, `quarto` - Document rendering
- `shiny`, `rsconnect` - Shiny app deployment
- `DT`, `yaml`, `fs`, `httr`, `jsonlite` - Data handling
- `bslib`, `sass` - Styling
- `tinytex` - LaTeX/PDF support

2.3.0.1 Verify Installation

Test that everything works:

```
{bash}
#| label: Verify installation in r console
#| eval: false
#| echo: true

("quarto --version") # Check Quarto system
quarto::quarto_render("index.qmd") # Test rendering a single page
```

3 Project structure and file explanations

```
{bash}
#| label: Root folder (AMORE-webpage) and the file structure
#| eval: false
#| echo: true

AMORE-webpage/
├── index.qmd                                # Homepage (root level)
├── _quarto.yml                             # Main configuration file
├── LICENSE
├── README.md
├── .gitignore
├── .node-version
├── package.json
├── netlify.toml                            # Netlify deployment config
├── netlify-setup-quarto.sh                 # Quarto setup for Netlify
├── install-quarto.sh                       # Quarto installation script
├── AMORE.Rproj                             # RStudio project file
├── .quarto/                               # Quarto build cache (auto-generated)
│   ├── idx/                              # Index files
│   ├── xref/                             # Cross-references
│   ├── listing/
│   ├── preview/
│   └── _freeze/
├── assets/
│   ├── favicons/                         # All favicon files
│   │   ├── favicon.ico
│   │   ├── apple-touch-icon.png
│   │   ├── favicon-96x96.png
│   │   ├── site.webmanifest
│   │   └── amore.favicon.ico
│   ├── images/                           # Logo and visual assets
│   │   ├── logo.unfinished_nowhite- Copy.png
│   │   ├── LiMAs.jpg
│   │   └── [other image files]
│   └── styles/
│       └── styles.scss                    # Main stylesheet
├── pages/                                # All content pages
│   ├── about.qmd
│   ├── contact.qmd
│   ├── guidelines.qmd
│   ├── Living_meta-analysis_Directory.qmd
│   ├── Protocol_checklist.qmd
│   ├── Resources.qmd
│   └── Standardization.qmd
└── LMAs/                                  # Living Meta-Analysis projects
```

```

├── _lma-template.qmd          # Template for new LMAs
└── project pages files

docs/                          # Documentation (working files)
├── _AMORE_Guide.qmd          # Website management guide
├── _recoveryfile.app.qmd
└── _recoveryfile.qmd

scripts/                       # Utility scripts
└── Setup.R                   # R dependencies installer

shiny-meta/                   # Shiny application
├── app.R                     # Main Shiny app
├── deploy.R                   # Deployment script
├── README.md                  # Shiny app documentation
├── .Renvirom                  # Saves shiny meta tokens
├── rsconnect/                # Shiny deployment config
│   ├── shinyapps.io/
│   │   └── meta-oxytocin/
│   │       └── shiny-meta.dcf
└── www/                       # Shiny app assets
    └── amore.favicon.ico

_site/                         # Generated website (DO NOT EDIT)
├── index.html
├── search.json
├── assets/
├── pages/
├── LMAs/
├── shiny-meta/
└── site_libs/

```

3.1 Root files

Root files are located directly in the **AMORE-webpage/** folder. These files control the fundamental behavior, configuration, and metadata of the entire website project. They define how the site is built, deployed, and maintained.

Key characteristics of root files:

- Control project-wide settings and behavior
- Required for the website to function properly
- Should be modified carefully as changes affect the entire site
- Most are configuration or metadata files (not content)

3.1.1 index.qmd

The homepage of the AMORE website. This is the first page visitors see when they navigate to <https://amore-project.org>.

Type: Quarto markdown document (content file)

CSS classes used directly in index.qmd:

- `.content-section` - Main container with transparent background on gradient
- `.header-logo` - Logo image styling (centering, max-width)
- `.subtitle-text` - Centered white subtitle text with specific font size
- `.primary-button` - Call-to-action button styling
- `.contact-buttons` - Button container for centering
- `.BNL-text` - Text styling for lab attribution

Global styles that also affect this page:

- `body` - Gradient animated background (visible behind `.content-section`)
- Typography rules (`p`, `a`, headings)
- `.primary-button:hover` - Hover effects on the CTA button
- Responsive breakpoints that adjust layout on mobile/tablet

3.1.2 __quarto.yml

The main configuration file that controls how the entire website is built and structured.

Type: YAML configuration file

Important notes:

- Changes here affect the entire site
- Always test after editing with `quarto preview`
- Incorrect YAML indentation will break the site
- Navbar order controls menu item order

3.1.3 LICENSE

Defines the legal terms under which the AMORE website code and content can be used, modified, and distributed.

Type: Plain text legal document

License: MIT open source license. [MIT license explanations](#).

3.1.4 README.md

Brief description of the AMORE website project displayed on the GitHub repository page.

Type: Markdown document

Purpose:

- First thing people see on GitHub
- Explains what the project is

3.1.5 .gitignore

Purpose: Tells Git which files and folders to ignore (not track or upload to GitHub).

Type: Plain text configuration file

Purpose:

- Keeps repository clean
- Prevents uploading generated/temporary files
- Reduces repository size
- Protects sensitive information

```
{r}
#| label: .gitignore contents
#| eval: false
#| echo: true

# R files
*.log
*.tex
*.aux
.Rproj.user
.Rproj.user/
.Rhistory
.RData
.Ruserdata
*.Rproj

# Quarto and website specific files
_site/
/.quarto/
*_cache/
*.bak
*_error.scss
*.backup
*.ipynb
/_site/
/_freeze/
```

```

/_book/
/*_files/

# Generated HTML files in SOURCE directories (not _site)
/*.html
/pages/*.html
/LMAs/*.html
/docs/*.html

# Generated dependency folders in SOURCE directories
/*_files/
/pages/*_files/
/LMAs/*_files/
/docs/*_files/

# OS files
.DS_Store
Thumbs.db

# Test files

docs/_recoveryfile.qmd
docs/_recoveryfile.app.qmd

# Generated files
_quarto_internal_scss_error.scss

# Backup files
*.backup
*.bak

```

Why exclude these:

- RStudio files are user-specific
- `__site/` is generated, not source code
- Cache files speed up local rendering but aren't needed in repo
- Reduces repository size

Never add to `.gitignore`:

- Source files (`.qmd`, `.R`, `.scss`)
- Configuration files (`__quarto.yml`, `netlify.toml`)
- Assets (images, logos)

3.1.6 .node-version

Specifies which Node.js version to use for building the website.

Type: Plain text version file

Purpose:

- Ensures consistent Node.js version across environments
- Used by Netlify during deployment
- Prevents version compatibility issues

Current content:

- Single line with Node.js version number (e.g., “16”)

When to edit:

- Updating Node.js requirements
- Fixing deployment issues related to Node.js version
- Usually leave unchanged unless necessary

3.1.7 Package.json

Defines npm scripts and Node.js dependencies for building and deploying the website.

Type: JSON configuration file

Purpose:

- Contains build scripts for Netlify automated deployment
- Manages project metadata
- Defines commands to install Quarto and render the site

Key sections:

- **name:** - Project identifier “amore-project”
- **version:** - Project version number
- **scripts:** - Build commands
 - **prebuild** - Installs Quarto before building
 - **build** - Renders the Quarto site

3.1.8 netlify.toml

Configuration file for Netlify deployment settings.

Type: TOML configuration file

Purpose:

- Tells Netlify how to build and deploy the website
- Specifies build commands and output directory
- Sets environment variables

Key settings:

- `publish = "_site"` - Where built files are located
- `command = "npm run build"` - Build command to execute
- `NODE_VERSION = "16"` - Node.js version for building

3.1.9 netlify-setup-quarto.sh

Bash script that installs Quarto on Netlify's build servers.

Type: Shell script (`.sh`)

Purpose:

- Downloads and installs specific Quarto version
- Runs during Netlify build process
- Ensures Quarto is available for rendering

Key actions:

- Downloads Quarto v1.3.450 tarball
- Extracts and installs to `~/.local/quarto`
- Creates symbolic link to binary
- Verifies installation

Version note:

- Script uses v1.3.450 for Netlify deployment
- Local development can use newer Quarto versions
- Version mismatch did not cause issues for myself, but if rendering issues, perhaps look into this

Important:

- Called by `package.json` prebuild script
- Must be executable (`chmod +x`)

3.1.10 install-quarto.sh

Bash script that installs Quarto locally (alternative installation method).

Type: Shell script (`.sh`)

Purpose:

- Downloads and installs Quarto v1.6.40
- For local development setup
- Installs to user's home directory

Key actions:

- Downloads Quarto tarball from GitHub
- Extracts to `$HOME/.local/bin.`, a temporary cache directory created during local Quarto installation. The `$HOME` directory is safe to delete.

Type: System directory

- Adds to `PATH`
- Verifies installation with `quarto --version`

When to use:

- Setting up development environment
- Alternative to downloading from Quarto website
- Linux/Unix systems

When to edit:

- Updating to newer Quarto version

Note:

- Different version than `netlify-setup-quarto.sh`
- Not used in deployment (local only)

3.2 .quarto/ folder

Quarto's build cache and internal processing files (auto-generated).

Type: System cache directory

Purpose:

- Stores rendering cache for faster builds
- Contains cross-reference indexes
- Holds frozen computation results
- Internal Quarto processing files

Structure:

Important notes:

- Auto-generated - recreated on each render
- Already excluded in **.gitignore**
- NB! Never edit manually
- Safe to delete if troubleshooting build issues
- Speeds up subsequent renders

When to delete:

- Fixing strange rendering errors
- Clearing cache after major changes
- Will be recreated automatically on next `quarto render`

3.3 assets/ folder

Contains all static files used across the website (images, styles, icons).

Seperated into three subfolders:

- favicorns
 - Contains all browser icons and Progressive Web App (PWA) assets.
 - Utilized in `__quarto.yml` file
- images
 - all images across the website are saved in this folder
- styles
 - Contains the `styles.scss` sheet that instructs the rendering process of design rules for the website

3.4 pages/ folder

Contains all main content pages of the website (except homepage).

Type: Content directory

Purpose:

- Houses all `.qmd` files for site pages
- Each file = one webpage
- Linked from navbar in `_quarto.yml`

Files:

- `about.qmd` - About AMORE, mission, team, steering committee
- `contact.qmd` - Contact form links
- `guidelines.qmd` - Step-by-step workflow for publishing LMAs
- `Living_meta-analysis_Directory.qmd` - Embedded Shiny app directory
- `Protocol_checklist.qmd` - Interactive checklist tool
- `Resources.qmd` - External resources and tools
- `Standardization.qmd` - AMORE framework requirements

3.5 LMAs/ folder

Contains all Living Meta-Analysis project pages.

Type: Content directory (project pages)

Purpose:

- Individual `.qmd` file for each LMA project
- Template file for creating new projects
- Fetched by Shiny app for directory listing

YAML metadata structure: Each project file contains metadata that the Shiny app reads:

Important notes:

- Shiny app reads these files via GitHub API
- Metadata must follow template structure
- Files starting with `_` are excluded from rendering

3.6 docs/ folder

Contains internal documentation and working files (not published to website).

Type: Documentation directory

Purpose:

- Working documents for website management
- Draft files and recovery backups
- Internal guides and notes

Files:

- `_AMORE_Guide.qmd` - Website management guide (this document)
- `_recoveryfile.qmd` - Backup/recovery file
- `_recoveryfile.app.qmd` - Backup/recovery file

Important notes:

- Files starting with `_` are not rendered to website
- Already excluded in `.gitignore` (for `.html` versions)
- For internal use only
- Not linked in navbar
- Safe to add more documentation files here

When to use:

- Creating internal documentation
- Draft content before moving to `pages/`
- Testing new features
- Reference materials for maintainers

3.7 scripts/ folder

Contains utility scripts for project setup and maintenance.

Type: Utility scripts directory

Purpose:

- Automate setup tasks
- Maintenance scripts
- Helper utilities

Files:

- **Setup.R** - Installs all required R packages and dependencies

Key functions in Setup.R:

- Installs R packages (rmarkdown, knitr, shiny, quarto, etc.)
- Installs TinyTeX for PDF generation
- Checks for existing installations before installing
- Loads required libraries

3.8 shiny-meta/ folder

Contains the Shiny application for the Living Meta-Analysis Directory.

Type: Shiny app directory

Purpose:

- Interactive searchable directory of all LMA projects
- Fetches project data from GitHub
- Filters and displays LMA metadata
- Deployed on shinyapps.io

Structure:

```
{bash}
#| label: shiny-meta directory
#| eval: false
#| echo: true

shiny-meta/
├─ app.R           # Main Shiny application code
├─ deploy.R        # Deployment script for shinyapps.io
├─ .Renvirom       # Keeps shiny-meta tokens safe
└─ README.md       # Shiny app documentation
```

```
└─ rsconnect/      # Deployment configuration
└─ www/           # Static assets (favicon)
```

Key files:

- **app.R** - Complete UI and server logic
- **deploy.R** - Contains shinyapps.io account credentials
- **rsconnect/** - Auto-generated deployment settings
- **.Renviron** ****Security critical file**** containing:
 - * **'SHINYAPPS_ACCOUNT'** - Your shinyapps.io account name
 - * **'SHINYAPPS_TOKEN'** - Authentication token for deployment
 - * **'SHINYAPPS_SECRET'** - Secret key for deployment

3.8.1 How app.R works:

1. app.R fetches **.qmd** files from **LMAs/** folder via GitHub API
2. Extracts YAML metadata from each project
3. Provides filtering by outcomes, population, methodology
4. Displays paginated results with links to project pages

3.8.1.1 Deployment:

- Hosted at: <https://meta-oxytocin.shinyapps.io/shiny-meta/>
- Non-paid server at shinyapps.io, paid server will improve page loading
- Embedded in: **pages/Living_meta-analysis_Directory.qmd**

3.8.1.2 .Renviron

This file is personal with the tokens from the shinyapp account needed to deploy the app. All users that will deploy anew the app.R to the server on the shinyapp account meta-oxytocin must create their own .Renviron file with personal token and secret. Which is collected by login in to the account at shinyapp.io.

The meta-oxytocin account uses unpaid version, to enable multiple account users, the account subscription must be upgraded to professional plan.

****Setting up deployment credentials for shinyapp:****

1. Create a **'Renviron'** file in the **'shiny-meta/'** directory: `“‘bash # .Renviron file structure
SHINYAPPS_ACCOUNT=your-account-name SHINYAPPS_TOKEN=your-token-here
SHINYAPPS_SECRET=your-secret-here “‘`
2. Get your credentials from shinyapps.io: * Log in to <https://www.shinyapps.io/> * Navigate to Account meta-oxytocin → Tokens * Show/Generate token * Copy the account name, token, and secret

3. In R, run ‘deploy.R’ which will use these credentials

Important notes:

- Updates automatically when new LMA files added to GitHub
- Requires shinyapps.io account for deployment

3.9 __site/ folder

Contains the generated/rendered website files (auto-generated, do not edit).

Type: Build output directory

Purpose:

- Final HTML files ready for deployment
- Compiled CSS and JavaScript
- Copied assets and resources
- Generated during build process

Structure:

```
{bash}
#| label: __site directory
#| eval: false
#| echo: true

__site/
├─ index.html          # Rendered homepage
├─ search.json         # Search index
├─ assets/             # Copied assets
├─ pages/              # Rendered page HTML files
├─ LMAs/               # Rendered LMA project pages
├─ shiny-meta/         # Copied Shiny app files
└─ site_libs/          # Quarto dependencies (Bootstrap, JS)
```

Important notes:

- Auto-generated - recreated on every **quarto render**
- Never edit files here - changes will be overwritten
- Already excluded from Git (in **.gitignore**)

Two separate build processes:

Local build (your computer):

- Purpose: Preview and test changes
- Command: **quarto preview** or **quarto render**

- Creates: `_site/` folder locally
- Result: For testing only, not deployed

Netlify build (production):

- Purpose: Deploy live website
 - Triggered: Automatically on GitHub push
 - Process: Netlify clones repo → runs `npm run build` → generates fresh `_site/`
 - Result: This is what visitors see at `amore-project.org`
-

4 Styling system SCSS

4.0.1 Overview

The AMORE website uses SCSS (Sassy CSS) for styling, which provides more power and flexibility than regular CSS. The main stylesheet is located at `assets/styles/styles.scss`. This file is automatically compiled to CSS when you render the Quarto site.

[Here you find scss rules](#)

What `styles.scss` controls:

- Global variables (colors, fonts)
- Typography system
- Layout components (hero sections, grids, cards)
- Navigation and footer
- Interactive elements (tabs, filters, buttons)
- Page-specific styles (index, contact, project pages)
- Responsive breakpoints (mobile, tablet, desktop). This means the layout adjusts to different screen sizes.
- Animations (gradient shifts, hover effects)

Linked in:

- `_quarto.yml` under `theme:` and `css:`
- Applied to all pages automatically

Important notes:

- Changes affect entire website

- Test thoroughly after edits and before committing
- Uses SCSS features (variables, nesting)
- Compiled during `quarto render`

File Paths in SCSS

When referencing images or other files in your SCSS, understanding file paths is crucial:

```
{scss}
#| label: file path
#| eval: false
#| echo: true

/* CORRECT - Goes up one folder level from assets/styles/ to assets/ */
background-image: url('../images/logo.png');

/* INCORRECT - Would look in assets/styles/images/ */
background-image: url('images/logo.png');
```

4.0.2 Structure of styles.scss

The file is organized into logical sections marked with comments:

1. Global Variables - Colors, fonts, and sizing used throughout
2. Body & Animations - Background effects and page-level styling
3. Typography - All text styling (headings, paragraphs, links)
4. Components - Reusable pieces (buttons, cards, navigation)
5. Page-Specific - Styles for particular pages (contact, about, etc.)
6. Responsive Design - Adjustments for different screen sizes

5 Adding new projects

This section guides you through adding a new living meta-analysis project to the AMORE website. Each project requires a `.qmd` file with properly formatted metadata that allows the Shiny app directory to filter and display your project correctly.

5.1 Getting Started:

1. Locate the template: Find `_lma-template.qmd` in the LMAs folder of your project directory
2. Create your project file:
 - Duplicate the template file and rename it with a descriptive name.
 - Naming convention: Use lowercase letters with underscores, no spaces
 - Examples: `smith_social_cognition.qmd` | `Smith Social Cognition.qmd`
3. Save location: Keep the new file in the LMAs folder alongside other project files
4. Alternatively: create new `.qmd` file that you save in the LMAs folder. Write scripts yourself or copy paste from this guide or template the necessary code components that you want to include.

Do NOT push to GitHub repository before the project page is finished. If you need to push your work for version control and to secure your work, add a `'_'` prefix to the name. Like this `_smith_social_cognition.qmd`. The prefix prevents the page from rendering correctly. The file loses access to the styles instructions from `styles.scss`, this means that the prefix should only be added after the preview but before pushing to Git repository.

5.1.1 What do you fill in?

1. Mandatory and Optional information
2. Fill inn the brackets [with the information requested inside the brackets] or under headers where you are explicitly told to provide information. Leave the rest of the code as it is. `### Authors` or `### Timeline` are headers and not comments inside the template and should not be removed.
3. Scripts: Everything under `#| echo: true` in the `{r}` code blocks should be included. If you are uncertain use the `_lma-template.qmd` to visualise what the script should look like.
4. When the script is created, Render to preview

5.1.1.1 Mandatory information (must be included):

- YAML header - Contains all metadata for filtering and display
- Information container - The blue panel at the top with project details (timeline, team, links)
- Abstract - Brief overview of the meta-analysis
- Resources - Links to preregistration, data, scripts, and publications

5.1.1.2 Optional information (include if relevant):

- Current Results
- Methodology Summary
- Inclusion and Exclusion Criteria
- Search Strategy
- Study Characteristics
- References

5.1.2 Format rules

```
{r}
#| label: YAML header format rules
#| description: This is not supposed to be included anywhere! It is just for explanatory
  ↳ purposes. Below is the YAML header as it should be included.
#| eval: false
#| echo: true

# The YAML format is very important. Incorrect indentation will result in error, or the
  ↳ filter system of the app.R (directory of projects) will be unable to correctly read the
  ↳ necessary information.

## if a metadata class is irrelevant for the project there are several steps to ignore the
  ↳ class:
oxytocin: #example class
  route: null          # ← Won't show
  dosage: NA           # ← Won't show
  assessment_method:   # ← Empty/missing, won't show
# XinterventionX      # remove class

YAML array format: # All three options should in theory work

# List with dashes without quotes
analytical_framework:
  - Frequentist
  - Bayesian
  - Mixed methods

# List with dashes AND quotes
analytical_framework:
  - "Frequentist"
  - "Bayesian"
  - "Mixed methods"

# Inline array with brackets AND quotes
analytical_framework: ["Frequentist", "Bayesian", "Mixed methods"]

# ❗ DO NOT MODIFY the format section at the bottom
```

```
# This section is required for proper page rendering:
format:
  html:
    page-layout: full
    toc: true
    # ... rest of format settings
```

5.2 Build the template

5.2.1 Mandatory sections

5.2.1.1 YAML header

```
{r}
#| label: YAML header
#| description: This should be at the very beginning of the project file. The filters
  ↳ options are matched with app.R. If you want to include new options this must also be
  ↳ added to the app's filter logic.# When research groups propose new projects through
  ↳ nettskjema "Propose your project", they are instructed to VOLUNTARY choose metadata
  ↳ options for their project to increase their visibility. If they choose not to answer
  ↳ this part of the form leave the sections blank or write null.
#| eval: false
#| echo: true

---
title: "Write title here"
analytical_framework: # Choose one or more
  - "Frequentist"
  - "Bayesian"
  - "Mixed methods"
oxytocin:
  intervention: # if applicable, choose one or more:
    - "Intranasal oxytocin administration"
    - "Oral oxytocin administration"
    - "Intravenous/injection oxytocin administration"
    - "Environmental/behavioral oxytocin manipulation"
    - "Perinatal oxytocin exposure"
  assessment_method: # if applicable, choose one or more:
    - "Behavioral assessment"
    - "Physiological response"
    - "Biological sample collection"
    - "Genetic studies"
    - "Neural/imaging measurement"
  route: # Choose one or more:# if applicable, choose one or more:
    - "Central"
    - "Peripheral"
    - "Various administration routes"
    - "Administration method unspecified"
  dosage: # if applicable, choose one or more:
    - "8 IU"
```

```

- "16 IU"
- "24 IU"
- "32 IU"
- "40 IU"
- "Variable dosage"
population:
  status: # if applicable, choose one or more:
    - "Healthy"
    - "Clinical"
    - "Mixed"
  age_group: # if applicable, choose one or more:
    - "Children"
    - "Adolescents"
    - "Adults"
    - "Older Adults"
    - "Mixed Age Groups"
  clinical_type: # Describe if applicable (e.g., "Autism Spectrum Disorder", "Depression")
outcomes:
  biological: # if applicable, choose one or more:
    - "Cardiovascular"
    - "Neuroendocrine"
    - "Neurological"
    - "Metabolic"
    - "Immune and Inflammatory"
    - "Pain and Sensory"
    - "Sleep and Circadian"
  psychological_behavioral: # if applicable, choose one or more:
    - "Mood and Emotion"
    - "Cognition and Memory"
    - "Stress and Coping"
    - "Eating and Appetite"
    - "Risk and Decision-Making"
    - "Sleep Behavior and Quality"
    - "Bonding and Attachment"
    - "Trust and Cooperation"
    - "Communication and Empathy"
    - "Aggression and Conflict"
  clinical: # if applicable, choose one or more:
    - "Neurodevelopmental"
    - "Mood Disorders"
    - "Psychotic Disorders"
    - "Addiction and Substance Use"
    - "Eating Disorders"
update-frequency: # Choose one:
  - "Every 6 months"
  - "Every 12 months"
  - "Every 18 months"
  - "Every 24 months"
dois:
  preregistration: "10.XXXX/template-prereg"
  preprint: "10.XXXX/template-preprint"
  publication: "10.XXXX/template-publication"
format: ### NB! This is important to include ##

```

```

html: ### NB! This is important to include ##
page-layout: full ### NB! This is important to include ##
toc: true ### NB! This is important to include ##
toc-title: "On this page" ### NB! This is important to include ##
toc-location: right ### NB! This is important to include ##
toc-depth: 3 ### NB! This is important to include ##
---
```

5.2.1.2 Information container

```

{r}
#| label: information container
#| description: This code chunk is the title and the information box visible under the
  ↳ project hero title. Insert the required information inside the [ ] brackets
#| eval: false
#| echo: true

::: {.project-hero-section}
::: {.project-hero-content}
::: {.project-hero-title}

Write title here

:::
:::
:::

::: {.project-info-panel}
::: {.project-info-grid}

::: {.info-card .authors-card}
### Authors
<div id="author-list">
  <div id="visible-authors">
    <strong>[First Author Name]</strong> ([Affiliation])<br> ## Write authors and
    <strong>[Second Author Name]</strong> ([Affiliation])<br> # affiliations inside
    <strong>[Senior Author Name]</strong> ([Affiliation]) ##### the brackets []
  </div>

  <div id="hidden-authors" style="display:none">
    <strong>[Fourth Author]</strong> ([Affiliation])<br>
    <strong>[Fifth Author]</strong> ([Affiliation])<br>
    <strong>[Sixth Author]</strong> ([Affiliation])<br>
    <strong>[Seventh Author]</strong> ([Affiliation])<br>
    <strong>[Eighth Author]</strong> ([Multiple Affiliations | Separated by Pipes])<br>
    <strong>[Additional Authors as needed...]</strong><br>
  </div>

```



```

<a href="#" id="author-toggle" class="toggle-link" onclick="toggleAuthors(); return
↪ false;">+ Show more authors</a>

<script>
function toggleAuthors() {
  var hiddenAuthors = document.getElementById("hidden-authors");
  var toggleButton = document.getElementById("author-toggle");

  if (hiddenAuthors.style.display === "none") {
    hiddenAuthors.style.display = "block";
    toggleButton.textContent = "- Show fewer authors";
  } else {
    hiddenAuthors.style.display = "none";
    toggleButton.textContent = "+ Show more authors";
  }
}
</script>
</div>
...

... {.info-card .timeline-card} ## Fill information inside the [] brackets
### Timeline
**Pre-registered:** [Month DD, YYYY]
**Last update:** [Status of current analysis]
**Next update:** [Planned date or "To be determined" (XX-month update cycle)]
...

... {.info-card .identifiers-card} ## Fill information inside the [] brackets
### Identifiers
**DOI:** [https://doi.org/ACTUAL_DOI](https://doi.org/ACTUAL_DOI)

**Keywords:** [keyword1], [keyword2], [keyword3], [methodology], [population]
...

... {.info-card .contact-card} ## Fill information inside the [] brackets
### Contact
**Principal Investigator:** [PI Full Name]
**Email:** [pi.email@institution.edu](mailto:pi.email@institution.edu)
...

...
...

```

5.2.1.3 Abstract

```

{r}
#| label: Abstract section of main content area
#| description: This is where you fill in the abstract. It should be the first header after
↪ the information container box.
#| eval: false
#| echo: true

```

```
## Abstract {#abstract}

write abstract under abstract header
```

5.2.1.4 Resources

```
{.html}
#| label: Resources section of main content area.
#| description: This code chunk is the resources section. The resources section is written
  ↪ as HTML code, with JavaScript for tab functions.
#| eval: false
#| echo: true

## Resources {#resources}
<body>
  <div class="resources-container">

    <!-- TAB NAVIGATION BUTTONS -->
    <div class="tab-navigation">
      <button class="tab-button active" onclick="switchTab('initial')">Initial
        ↪ Publication</button>
      <button class="tab-button" onclick="switchTab('update1')">Update 1</button>
      <button class="tab-button" onclick="switchTab('update2')">Update 2</button>
    </div>

    <!-- INITIAL PUBLICATION TAB -->
    <div id="initial" class="tab-content active">

      <!-- Timeline indicator for initial publication -->
      <div class="timeline-indicator">
        <div class="status-badge status-published">Published</div>
        <span>[Month Year]</span>
      </div>

      <!-- PROTOCOL & PUBLICATION CATEGORY -->
      <div class="resource-category">
        <div class="category-header">
          <svg class="category-icon" fill="none" stroke="currentColor" viewBox="0
            ↪ 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
              ↪ stroke-width="2" d="M9 12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0
              ↪ 012-2h5.586a1 1 0 01.707.293l5.414 5.414a1 1 0 01.293.707V19a2
              ↪ 2 0 01-2 2z"></path>
          </svg>
          <h3 class="category-title">Protocol & Publication</h3>
        </div>

        <!-- Study Protocol Item -->
```

```

<div class="resource-item">
  <div class="resource-header">
    <div class="resource-info">
      <div class="resource-title">Study Protocol</div>
      <div class="resource-description">[Write short description of study
        ↪ protocol, e.g., "Preregistered analysis plan outlining
        ↪ inclusion criteria and analytical approach"]</div>
      <div class="resource-links">
        <a href="[INSERT_FULL_URL_HERE]" class="resource-link"
          ↪ target="_blank">
          <div class="link-title">[Link name, e.g.,
            ↪ "Preregistration on OSF"]</div>
          <div class="link-url">[Clean URL, e.g.,
            ↪ "osf.io/8fzdy"]</div>
        </a>
      </div>
    </div>
  </div>
</div>

<!-- Published Article Item -->
<div class="resource-item">
  <div class="resource-header">
    <div class="resource-info">
      <div class="resource-title">Published Article</div>
      <div class="resource-description">[Write description of published
        ↪ article, e.g., "Peer-reviewed article published in
        ↪ Molecular Psychiatry"]</div>
      <div class="resource-links">
        <a href="[INSERT_FULL_DOI_URL_HERE]" class="resource-link"
          ↪ target="_blank">
          <div class="link-title">[Link name, e.g., "Molecular
            ↪ Psychiatry Article"]</div>
          <div class="link-doi">[DOI, e.g.,
            ↪ "10.1038/s41380-024-02871-4"]</div>
        </a>
      </div>
    </div>
  </div>
</div>

<!-- Preprint Version Item -->
<div class="resource-item">
  <div class="resource-header">
    <div class="resource-info">
      <div class="resource-title">Preprint Version</div>
      <div class="resource-description">[Write description, e.g., "Open
        ↪ access preprint with full supplementary materials"]</div>
      <div class="resource-links">
        <a href="[INSERT_FULL_URL_HERE]" class="resource-link"
          ↪ target="_blank">
          <div class="link-title">[Link name, e.g., "OSF
            ↪ Preprint"]</div>
        </a>
      </div>
    </div>
  </div>
</div>

```

```

        <div class="link-url">[Clean URL, e.g.,
        ↪ "osf.io/hjyfjygg"]</div>
    </a>
</div>
</div>
</div>
</div>
</div>
</div>

<!-- DATA & ANALYSIS CATEGORY -->
<div class="resource-category">
    <div class="category-header">
        <svg class="category-icon" fill="none" stroke="currentColor" viewBox="0
        ↪ 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
            ↪ d="M4 7v10c0 2.21 3.589 4 8 4s8-1.79 8-4V7M4 7c0 2.21 3.589 4 8
            ↪ 4s8-1.79 8-4M4 7c0-2.21 3.589-4 8-4s8 1.79 8 4"></path>
        </svg>
        <h3 class="category-title">Data & Analysis</h3>
    </div>

    <!-- Complete Dataset Item -->
    <div class="resource-item">
        <div class="resource-header">
            <div class="resource-info">
                <div class="resource-title">Complete Dataset</div>
                <div class="resource-description">[Short description of data,
                ↪ e.g., "Raw and cleaned data files with codebook"]</div>
                <div class="resource-links">
                    <a href="[INSERT_FULL_URL_HERE]" class="resource-link"
                    ↪ target="_blank">
                        <div class="link-title">[Link name, e.g., "OSF Data
                        ↪ Repository"]</div>
                        <div class="link-url">[Clean URL, e.g.,
                        ↪ "osf.io/6hsjfg"]</div>
                    </a>
                </div>
            </div>
        </div>
    </div>
</div>

    <!-- Analysis Scripts Item -->
    <div class="resource-item">
        <div class="resource-header">
            <div class="resource-info">
                <div class="resource-title">Analysis Scripts</div>
                <div class="resource-description">[Description of scripts, e.g.,
                ↪ "R scripts for all meta-analytic models and sensitivity
                ↪ analyses"]</div>
                <div class="resource-links">
                    <a href="[INSERT_FULL_URL_HERE]" class="resource-link"
                    ↪ target="_blank">

```

```

        <div class="link-title">[Link name, e.g., "R Analysis
        ↪ Scripts"]</div>
        <div class="link-url">[Clean URL or DOI]</div>
    </a>
</div>
</div>
</div>
</div>
</div>
</div>

<!-- RELATED RESEARCH CATEGORY -->
<div class="resource-category">
    <div class="category-header">
        <svg class="category-icon" fill="none" stroke="currentColor" viewBox="0
        ↪ 0 24 24">
            <path stroke-linecap="round" stroke-linejoin="round"
            ↪ stroke-width="2" d="M13.828 10.172a4 4 0 00-5.656 0l-4 4a4 4 0
            ↪ 105.656 5.656l1.102-1.101m-.758-4.899a4 4 0 005.656 0l4-4a4 4 0
            ↪ 00-5.656-5.656l-1.1 1.1"></path>
        </svg>
        <h3 class="category-title">Related Research</h3>
    </div>

    <!-- Related Research Item -->
    <div class="resource-item">
        <div class="resource-header">
            <div class="resource-info">
                <div class="resource-title">Theoretical Background</div>
                <div class="resource-description">[Description, e.g., "Key
                ↪ theoretical papers informing this meta-analysis"]</div>
                <div class="resource-links">
                    <a href="[INSERT_FULL_URL_HERE]" class="resource-link"
                    ↪ target="_blank">
                        <div class="link-title">[Paper title or description]</div>
                        <div class="link-doi">[DOI, e.g., "10.1234/example"]</div>
                    </a>
                    <a href="[INSERT_FULL_URL_HERE]" class="resource-link"
                    ↪ target="_blank">
                        <div class="link-title">[Second paper title or
                        ↪ description]</div>
                        <div class="link-doi">[DOI, e.g., "10.5678/example"]</div>
                    </a>
                </div>
            </div>
        </div>
    </div>
</div>
</div>

</div>

<!-- UPDATE 1 TAB -->

```

```

<div id="update1" class="tab-content">

  <!-- Coming soon notice -->
  <div class="coming-soon-notice">
    <strong>Update 1 Coming Soon</strong> • Expected: [Month Year] (updates
    ↳ every [XX] months)
  </div>

  <!-- Timeline indicator for update 1 -->
  <div class="timeline-indicator">
    <div class="status-badge status-coming-soon">In Development</div>
    <span>Next update within 24 months</span>
  </div>
</div>

<!-- UPDATE 2 TAB -->
<div id="update2" class="tab-content">

  <!-- Future notice -->
  <div class="future-notice">
    <strong>Future Update</strong> • Expected: [Month Year]
  </div>

  <!-- Timeline indicator for update 2 -->
  <div class="timeline-indicator">
    <div class="status-badge status-future">Future</div>
    <span>Long-term update planning</span>
  </div>
</div>

</div>

<!-- JAVASCRIPT FOR TAB SWITCHING -->
<script>
  function switchTab(tabId) {
    // Hide all tab contents
    const contents = document.querySelectorAll('.tab-content');
    contents.forEach(content => content.classList.remove('active'));

    // Remove active class from all buttons
    const buttons = document.querySelectorAll('.tab-button');
    buttons.forEach(button => button.classList.remove('active'));

    // Show selected tab content
    document.getElementById(tabId).classList.add('active');

    // Add active class to clicked button
    event.target.classList.add('active');
  }
</script>

```

```

</body>
</html>

```

Screenshots from the template of the resources html section: The neon green text is where you can or need to customize the html code.

```

## Resources {#resources}
```{=html}
<body>
 <div class="resources-container">
 <div class="tab-navigation">
 <button class="tab-button active" onclick="switchTab('initial')">Initial
Publication</button>
 <button class="tab-button" onclick="switchTab('update1')">Update 1</button>
 >
 <button class="tab-button" onclick="switchTab('update2')">Update 2</button>
 >
 </div>

 <div id="initial" class="tab-content active">
 <div class="timeline-indicator">
 <div class="status-badge status-published">Published</div>
 January 2025
 </div>

 <div class="resource-category">
 <div class="category-header">
 <svg class="category-icon" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
 <path stroke-linecap="round" stroke-linejoin="round" stroke
-width="2" d="M9 12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0 012-2h5.586a1 1 0 01.707.293l5
.414 5.414a1 1 0 01.293.707V19a2 2 0 01-2 2z"/>
 </svg>
 <h3 class="category-title">Protocol & Publication</h3>
 </div>

```

Figure 1: screenshot of resources section. Neongreen font is where you need to customize the code.

```

-width="2" d="M9 12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0 012-2h5.586a1 1 0 01.707.293l5
.414 5.414a1 1 0 01.293.707V19a2 2 0 01-2 2z"/>
</svg>
<h3 class="category-title">Protocol & Publication</h3>
</div>

<div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Study Protocol</div>
 <div class="resource-description">Write very short
description here of study protocol. E.g. preregistered analysis plan</div>
 <div class="resource-links">
 <a href="https://osf.io/insert_doi_here" class
="resource-link" target="_blank">
 <div class="link-title">What name you give the
link e.g. Preregistration on OSF</div>
 <div class="link-url">clean DOI e.g. osf.io/8fzdy
</div>

 </div>
 </div>
 </div>
</div>

<div class="resource-item">

```

Figure 2: screenshot of resources section. Neongreen font is where you need to customize the code.

```

<div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Published Article</div>
 <div class="resource-description">Write description of
published article here like what journal</div>
 <div class="resource-links">
 <a href="https://doi.org/insert/doi/here" class
="resource-link" target="_blank">
 <div class="link-title">Title of the link, e.g.
Molecular Psychiatry Article</div>
 <div class="link-doi">DOI: 67435y4835</div>

 </div>
 </div>
 </div>
</div>

<div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Preprint Version</div>
 <div class="resource-description">Write description e.g.
Open access preprint with full supplementary materials</div>
 <div class="resource-links">
 <a href="https://osf.io/link_here" class="resource
-link" target="_blank">
 <div class="link-title">E.g. OSF Preprint</div>
 <div class="link-url">e.g. osf.io/hjyfjygg</div>

 </div>
 </div>
 </div>
</div>
</div>

```

Figure 3: screenshot of resources section. Neongreen font is where you need to customize the code.



```

<div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Published Article</div>
 <div class="resource-description">Write description of
published article here like what journal</div>
 <div class="resource-links">
 <a href="https://doi.org/insert/doi/here" class
="resource-link" target="_blank">
 <div class="link-title">Title of the link, e.g.
Molecular Psychiatry Article</div>
 <div class="link-doi">DOI: 67435y4835</div>

 </div>
 </div>
 </div>
</div>

<div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Preprint Version</div>
 <div class="resource-description">Write description e.g.
Open access preprint with full supplementary materials</div>
 <div class="resource-links">
 <a href="https://osf.io/link_here" class="resource
-link" target="_blank">
 <div class="link-title">E.g. OSF Preprint</div>
 <div class="link-url">e.g. osf.io/hjyfjygg</div>

 </div>
 </div>
 </div>
</div>
</div>

```

Figure 4: screenshot of resources section. Neongreen font is where you need to customize the code.

```

<div class="resource-category">
 <div class="category-header">
 <svg class="category-icon" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
 <path stroke-linecap="round" stroke-linejoin="round" stroke
-width="2" d="M4 7v10c0 2.21 3.589 4 8 4s8-1.79 8-4V7M4 7c0 2.21 3.589 4 8 4s8-1.79 8
-4M4 7c0-2.21 3.589-4 8-4s8 1.79 8 4"></path>
 </svg>
 <h3 class="category-title">Data & Analysis</h3>
 </div>

 <div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Complete Dataset</div>
 <div class="resource-description">short descriptions of
data like raw data, cleaned data</div>
 <div class="resource-links">
 <a href="https://osf.linkhere/" class="resource-link"
target="_blank">
 <div class="link-title">e.g. OSF Data Repository
</div>
 <div class="link-url">e.g. osf.io/6hsjfg66sdjs
</div>

 </div>
 </div>
 </div>
 </div>
</div>

```

Figure 5: screenshot of resources section. Neongreen font is where you need to customize the code.

```

 <div class="resource-category">
 <div class="category-header">
 <svg class="category-icon" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
 <path stroke-linecap="round" stroke-linejoin="round" stroke
-width="2" d="M4 7v10c0 2.21 3.589 4 8 4s8-1.79 8-4V7M4 7c0 2.21 3.589 4 8 4s8-1.79 8
-4M4 7c0-2.21 3.589-4 8-4s8 1.79 8 4"></path>
 </svg>
 <h3 class="category-title">Data & Analysis</h3>
 </div>

 <div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Complete Dataset</div>
 <div class="resource-description">short descriptions of
data like raw data, cleaned data</div>
 <div class="resource-links">
 <a href="https://osf.linkhere/" class="resource-link"
target="_blank">
 <div class="link-title">e.g. OSF Data Repository
</div>
 <div class="link-url">e.g. osf.io/6hsjfg66sdjs
</div>

 </div>
 </div>
 </div>
 </div>
 </div>

```

Figure 6: screenshot of resources section. Neongreen font is where you need to customize the code.

```

 <div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Analysis scripts</div>
 <div class="resource-description">description of what is
here, like Python analysis scripts</div>
 <div class="resource-links">
 <a href="https://linkhere/" class="resource-link"
target="_blank">
 <div class="link-title">e.g. Python scripts</div>
 <div class="link-url">e.g. OSF
repository DOI</div>

 </div>
 </div>
 </div>
 </div>

 <div class="resource-category">
 <div class="category-header">
 <svg class="category-icon" fill="none" stroke="currentColor"
viewBox="0 0 24 24">
 <path stroke-linecap="round" stroke-linejoin="round" stroke
-width="2" d="M13.828 10.172a4 4 0 00-5.656 0l-4 4a4 4 0 105.656 5.656l1.102-1.101m
-.758-4.899a4 4 0 005.656 0l4-4a4 4 0 00-5.656-5.656l-1.1 1.1"></path>
 </svg>
 <h3 class="category-title">Related Research</h3>
 </div>

```

Figure 7: screenshot of resources section. Neongreen font is where you need to customize the code.

```

 <h3 class="category-title">Related Research</h3>
 </div>

 <div class="resource-item">
 <div class="resource-header">
 <div class="resource-info">
 <div class="resource-title">Theoretical Background</div>
 <div class="resource-description">Key theoretical papers
informing this meta-analysis</div>
 <div class="resource-links">
 <a href="https://linktorlatedpaper" class="resource
-link" target="_blank">
 <div class="link-title">Title of paper</div>
 <div class="link-doi">DOI: of paper</div>

 <a href="https://linkheresecondpaper" class="resource
-link" target="_blank">
 <div class="link-title">title of second paper</div>
 <div class="link-doi">DOI:tosecondpaper-x</div>

 </div>
 </div>
 </div>
 </div>
</div>

```

Figure 8: screenshot of resources section. Neongreen font is where you need to customize the code.

```

<div id="update1" class="tab-content">
 <div class="coming-soon-notice">
 Update 1 Coming Soon • Expected: Month and year
(updates every XX months)
 </div>

 <div class="timeline-indicator">
 <div class="status-badge status-coming-soon">In Development</div>
 Next update within 24 months
 </div>
</div>

<div id="update2" class="tab-content">
 <div class="future-notice">
 Future Update • Expected: month and year
 </div>

 <div class="timeline-indicator">
 <div class="status-badge status-future">Future</div>
 Long-term update planning
 </div>
</div>
</div>

```

Figure 9: screenshot of resources section. Neongreen font is where you need to customize the code.

```

<div id="update1" class="tab-content">
 <div class="coming-soon-notice">
 Update 1 Coming Soon • Expected: Month and year
 (updates every XX months)
 </div>

 <div class="timeline-indicator">
 <div class="status-badge status-coming-soon">In Development</div>
 Next update within 24 months
 </div>

<div id="update2" class="tab-content">
 <div class="future-notice">
 Future Update • Expected: month and year
 </div>

 <div class="timeline-indicator">
 <div class="status-badge status-future">Future</div>
 Long-term update planning
 </div>
</div>

```

Figure 10: screenshot of resources section. Neongreen font is where you need to customize the code.

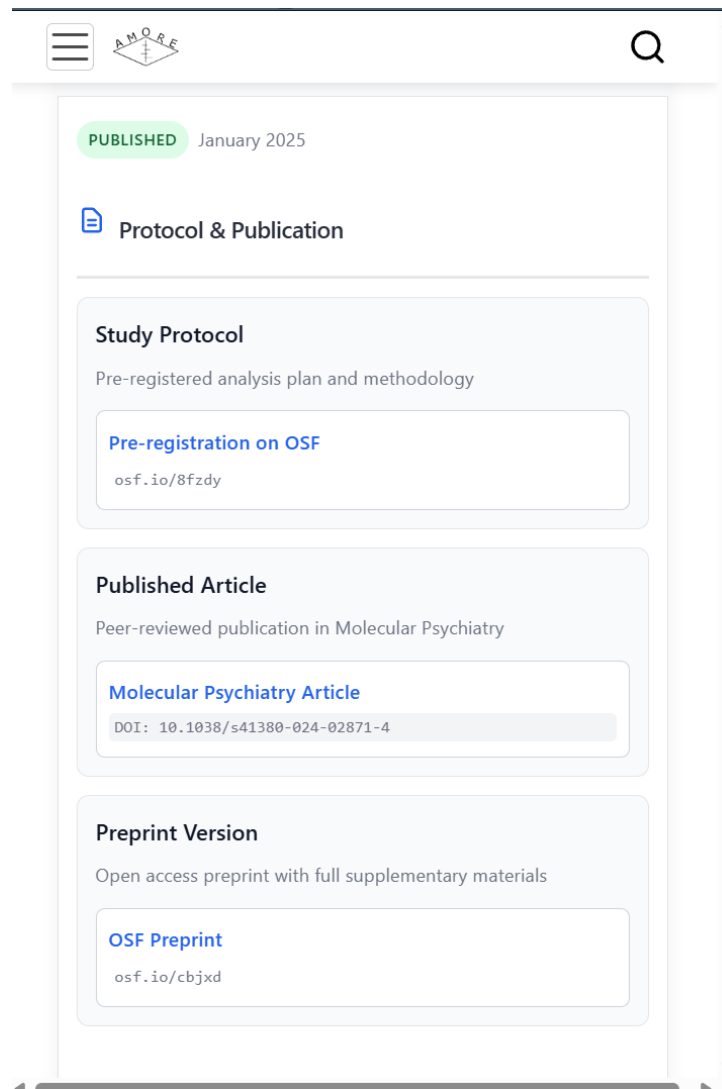


Figure 11: Visual of resources section

## 5.2.2 Optional sections

### 5.2.2.1 Current results

```
{.html}
#| label: Current results section of main content area
#| description: HTML code for the current results section for project pages. Pre-built HTML
 ↳ code chunks to mix and match. Different containers to show significant negative or
 ↳ positive findings or null findings. Needs to be wrapped in the div class
 ↳ "results-container"
#| eval: false
#| echo: true
```

```

Current Results {#current-results}
<body>
 <div class="results-container">

 <!-- OVERALL FINDINGS OPTIONS -->

 <!-- Option A: No Overall Effect -->
 <div class="overall-findings" style="background: #f0f9ff; border-left: 4px solid
 ↪ #3b82f6;">
 <p>The meta-analysis found no overall significant effect of
 ↪ oxytocin administration on [OUTCOME] (Hedges' g = [VALUE], 95% CI [[LOWER],
 ↪ [UPPER]], p = [P-VALUE]).
 </div>

 <!-- Option B: Significant Positive Effect -->
 <div class="overall-findings" style="background: #f0fdf4; border-left: 4px solid
 ↪ #10b981;">
 <p>The meta-analysis found a significant positive effect of
 ↪ oxytocin administration on [OUTCOME] (Hedges' g = [VALUE], 95% CI [[LOWER],
 ↪ [UPPER]], p = [P-VALUE]).</p>
 <p>This effect was [consistent across domains / varied by subdomain], with
 ↪ [ADDITIONAL CONTEXT].</p>
 </div>

 <!-- Option C: Significant Negative Effect -->
 <div class="overall-findings" style="background: #fef2f2; border-left: 4px solid
 ↪ #ef4444;">
 <p>The meta-analysis found a significant negative effect of
 ↪ oxytocin administration on [OUTCOME] (Hedges' g = [VALUE], 95% CI [[LOWER],
 ↪ [UPPER]], p = [P-VALUE]).</p>
 <p>This effect suggests that oxytocin [INTERPRETATION OF NEGATIVE EFFECT].</p>
 </div>

 <!-- Option D: Mixed/Heterogeneous Effects -->
 <div class="overall-findings" style="background: #fffbeb; border-left: 4px solid
 ↪ #f59e0b;">
 <p>The meta-analysis revealed substantial heterogeneity in
 ↪ oxytocin's effects on [OUTCOME] (Hedges' g = [VALUE], 95% CI [[LOWER],
 ↪ [UPPER]], I2 = [VALUE]%).</p>
 <p>Effects varied significantly by [MODERATOR], suggesting that oxytocin's
 ↪ impact is highly context-dependent.</p>
 </div>

 <!-- KEY FINDINGS SECTION WRAPPER -->

```

```

<div class="key-findings-section">
 <h3 class="key-findings-title">Key Findings by Domain</h3>

 <div class="findings-grid">

 <!-- FINDING CARD OPTIONS -->

 <!-- Card Type A: Significant Positive Effect -->
 <div class="finding-card significant" style="background: #f0fdf4;">
 <div class="finding-domain">
 [DOMAIN NAME]
 Significant
 ↳ Effect
 </div>
 <div class="finding-effect">Hedges' g = [VALUE]</div>
 <div class="finding-significance">p = [P-VALUE] • Statistically
 ↳ significant positive effect</div>
 </div>

 <!-- Card Type B: No Significant Effect -->
 <div class="finding-card non-significant" style="background: #f9fafb;">
 <div class="finding-domain">
 [DOMAIN NAME]
 No Effect
 </div>
 <div class="finding-effect">Hedges' g = [VALUE]</div>
 <div class="finding-significance">No significant effect detected</div>
 </div>

 <!-- Card Type C: Significant Negative Effect -->
 <div class="finding-card negative-significant" style="background: #fef2f2;">
 <div class="finding-domain">
 [DOMAIN NAME]
 Significant Negative
 ↳ Effect
 </div>
 <div class="finding-effect">Hedges' g = [NEGATIVE VALUE]</div>
 <div class="finding-significance">p = [P-VALUE] • Statistically
 ↳ significant negative effect</div>
 </div>

 <!-- Card Type D: Trending/Marginal Effect -->
 <div class="finding-card non-significant" style="background: #fffbeb;">
 <div class="finding-domain">
 [DOMAIN NAME]

```

```

 Trending
 </div>
 <div class="finding-effect">Hedges' g = [VALUE]</div>
 <div class="finding-significance">p = [P-VALUE] • Marginally significant
 ↳ trend</div>
 </div>

</div>
</div>

<!-- BAYESIAN ANALYSIS OPTIONS -->

<!-- Option A: Evidence Against Effect (Null Result) -->
<div class="bayesian-analysis" style="background: linear-gradient(135deg, #dbeafe
↳ 0%, #bfdbfe 100%);">
 <div class="bayesian-title">
 <svg width="20" height="20" fill="currentColor" viewBox="0 0 20 20">
 <path fill-rule="evenodd" d="M6 2a2 2 0 0-2 2v12a2 2 0 02 2h8a2 2 0
 ↳ 002-2V4a2 2 0 00-2-2H6zm1 2a1 1 0 00 2h6a1 1 0 100-2H7zm6 7H7a1 1 0
 ↳ 100 2h6a1 1 0 100-2zm0 4H7a1 1 0 100 2h6a1 1 0 100-2z"
 ↳ clip-rule="evenodd"/>
 </svg>
 Bayesian Meta-Analysis Results
 </div>
 <div class="bayesian-content">
 The Robust Bayesian Meta-Analysis provided moderate support for the
 ↳ absence of an effect (BF10 = [VALUE]) and [moderate/strong] evidence
 ↳ against publication bias (BFPB = [VALUE]). These findings strengthen
 ↳ confidence in the null result.
 </div>
</div>

<!-- Option B: Evidence For Effect -->
<div class="bayesian-analysis" style="background: linear-gradient(135deg, #d1fae5
↳ 0%, #a7f3d0 100%);">
 <div class="bayesian-title">
 <svg width="20" height="20" fill="currentColor" viewBox="0 0 20 20">
 <path fill-rule="evenodd" d="M6 2a2 2 0 0-2 2v12a2 2 0 02 2h8a2 2 0
 ↳ 002-2V4a2 2 0 00-2-2H6zm1 2a1 1 0 00 2h6a1 1 0 100-2H7zm6 7H7a1 1 0
 ↳ 100 2h6a1 1 0 100-2zm0 4H7a1 1 0 100 2h6a1 1 0 100-2z"
 ↳ clip-rule="evenodd"/>
 </svg>
 Bayesian Meta-Analysis Results
 </div>

```



```

 <div class="bayesian-content">
 The Robust Bayesian Meta-Analysis provided strong evidence for the
 ↪ effect (BF10 = [VALUE]), supporting the frequentist findings. [ADDITIONAL
 ↪ INTERPRETATION].
 </div>
 </div>

<!-- Option C: Inconclusive Bayesian Evidence -->
<div class="bayesian-analysis" style="background: linear-gradient(135deg, #fef3c7
 ↪ 0%, #fde68a 100%);">
 <div class="bayesian-title">
 <svg width="20" height="20" fill="currentColor" viewBox="0 0 20 20">
 <path fill-rule="evenodd" d="M6 2a2 2 0 0-2 2v12a2 2 0 02 2h8a2 2 0
 ↪ 002-2V4a2 2 0 00-2-2H6zm1 2a1 1 0 00 2h6a1 1 0 100-2H7zm6 7H7a1 1 0
 ↪ 100 2h6a1 1 0 100-2zm0 4H7a1 1 0 100 2h6a1 1 0 100-2z"
 ↪ clip-rule="evenodd"/>
 </svg>
 Bayesian Meta-Analysis Results
 </div>
 <div class="bayesian-content">
 The Robust Bayesian Meta-Analysis provided inconclusive
 ↪ evidence (BF10 = [VALUE]), suggesting the data cannot clearly distinguish
 ↪ between the presence and absence of an effect.
 </div>
</div>

<!-- STATISTICAL NOTES OPTIONS -->

<!-- Power Analysis Note -->
<div class="statistical-note" style="background: #f3f4f6; border-left: 3px solid
 ↪ #6b7280;">
 <div class="statistical-note-title">Statistical Power Analysis</div>
 <div class="statistical-note-content">
 Individual studies were generally underpowered ([PERCENT]%
 ↪ median power) to detect the observed summary effect size. This highlights the
 ↪ importance of meta-analytic approaches for detecting small but meaningful effects.
 </div>
</div>

<!-- Heterogeneity Note -->
<div class="statistical-note" style="background: #fef3c7; border-left: 3px solid
 ↪ #f59e0b;">
 <div class="statistical-note-title">Heterogeneity Analysis</div>
 <div class="statistical-note-content">

```

```

 Substantial heterogeneity was observed across studies ($I^2 = [VALUE]\%$, $\tau^2 =$
 ↪ $[VALUE]$), suggesting that $[INTERPRETATION\ OF\ HETEROGENEITY\ SOURCE]$.
 </div>
</div>

<!-- Publication Bias Note -->
<div class="statistical-note" style="background: #dbeafe; border-left: 3px solid
 ↪ #3b82f6;">
 <div class="statistical-note-title">Publication Bias Assessment</div>
 <div class="statistical-note-content">
 [Evidence of publication bias was/was not detected] using $[METHOD]$.
 ↪ $[INTERPRETATION\ AND\ IMPLICATIONS]$.
 </div>
</div>

<!-- FIGURE SECTION -->
<div class="figure-section" style="margin: 3rem 0; text-align: center;">
 <div class="figure-container" style="background: white; border: 1px solid
 ↪ #e5e7eb; border-radius: 12px; padding: 2rem; box-shadow: 0 2px 8px rgba(0,
 ↪ 0, 0, 0.05);">

 <div class="figure-caption" style="font-size: 0.9rem; color: #4b5563;
 ↪ line-height: 1.6; text-align: left; margin-top: 1rem;">
 Figure $[NUMBER]$: $[FIGURE\ TITLE]$. $[DETAILED\ CAPTION]$
 ↪ $[DESCRIBING\ THE\ FIGURE\ AND\ KEY\ FINDINGS]$
 </div>
 </div>
</div>

<!-- No results yet -->
<div class="current-results-container">
 <div class="preregistered-notice">
 <p>This is a pre-registered protocol. Initial analysis has not
 ↪ yet been conducted. Results will be posted when the analysis is completed
 ↪ and the pre-print is published.</p>
 </div>

</div>
</body>

```

### 5.2.2.2 Inclusion and exclusion criteria

```

{r}
#| label: Inclusion and Exclusion Criteria
#| description: Here you can create two separate boxes with the inclusion and exclusion
 ↳ criteria for the meta-analysis.
#| eval: false
#| echo: true

Inclusion and Exclusion Criteria {#criteria}

::: {.criteria-box}
Inclusion Criteria

- Type of Studies: [Study designs included, e.g., "Randomized controlled trials",
 ↳ "Original research written in English"]
- Intervention: [Specific intervention criteria, e.g., "Intranasal oxytocin
 ↳ administration"]
- Population: [Target population, e.g., "Healthy adults aged 18-65"]
- Outcomes: [Primary and secondary outcomes, e.g., "Social behavior measures",
 ↳ "Validated scales for trust"]
- Data Requirements: [Statistical requirements, e.g., "Must report means and standard
 ↳ deviations or effect sizes"]
:::

::: {.criteria-box}
Exclusion Criteria

- [Specific exclusion criteria relevant to the research question]
- [Additional exclusion criteria]
- [Overlap/duplicate publication handling]
:::

```

### 5.2.2.3 Methodology summary

```

{r}
#| label: Methodology summary of main content area
#| description: A section handy to explain a summary of the methodology of the living
 ↳ meta-analysis.
#| eval: false
#| echo: true

Methodology Summary {#methodology}

This living meta-analysis follows the PRISMA guidelines for systematic reviews.

The quality of included studies will be assessed using [specific quality assessment tool,
 ↳ e.g., "Cochrane Risk of Bias tool for RCTs"]. Studies will be independently rated by two
 ↳ reviewers [describe rating categories and scoring system].

```

```
Statistical analyses will be conducted using [statistical software] with the [specific
↳ packages]. [Type of effects model] will be used with [specific estimator].
↳ Heterogeneity will be evaluated by calculating [heterogeneity measures]. Publication
↳ bias will be assessed using [bias assessment methods].
```

#### 5.2.2.4 Search strategy

```
{r}
#| label: Search strategy of main content area
#| description: Write the search strings used on different search engines like PubMed or
↳ similar for reproducibility. It is very nice to have the search strings on the project
↳ page as this is apparently not always documented as makes reproducibility more
↳ difficult.
#| eval: false
#| echo: true

Search Strategy {#search-strategy}

::: {.search-strategy-box}
write database here:
write [search AND string] here

EMBASE:
'Pitocin®' OR 'Syntocinon®' OR 'synthetic oxytocin' AND 'labour/labor' OR 'birth' OR
↳ 'perinatal' OR 'prenatal' OR 'obstetric'

Web of Science:
('Pitocin®' OR 'Syntocinon®' OR 'synthetic oxytocin' AND 'labour/labor' OR 'birth' OR
↳ 'perinatal' OR 'prenatal' OR 'obstetric') AND ('delivery' AND 'augment*' OR 'induc*')

MEDLINE:
'Pitocin®' OR 'Syntocinon®' OR 'synthetic oxytocin' AND 'labour/labor' OR 'birth' OR
↳ 'perinatal' OR 'prenatal' OR 'obstetric' AND 'antepartum'

:::
```

### 5.3 Previewing your work

---

## 6 Troubleshooting

This section provides solutions to common problems you may encounter when creating or updating project pages.

Include:

- `__recoveryfile.qmd`, `__recoveryapp.R`, `styles.scss.backup`
  - `__recoveryapp.R`, is used to test updates to `app.R` before publishing to `shinyapp.io`, once updates are complete, copy paste script to `app.R`, always keep backup with bigger changes.
  - If you want to use any of these files to preview changes, the underscore `__` must be removed from the name temporary so that the file gets access to `styles.scss` when rendering, once the changes are as you wish, include the `__` at the beginning of the file name to hide it again. Files starting with `__` has been added to `.gitignore`
- Styling Problems
  - [To be expanded: Issues with formatting, colors, or layout]
  - Inspect element
- cache and restart
- YAML syntax error
  - [To be expanded: Common YAML formatting mistakes and how to fix them]
- Image Display Issues
  - [To be expanded: Problems with figures and images not showing]
- Search/Filter Not Working
  - [To be expanded: How project metadata connects to the Shiny app search]
- Do you wrap the code correctly? Especially `html` is very strict here

---

## 7 Website maintenance

You find the correct `scss` classes for the different pages under the 3.1 File-by-file breakdown under each file explanation.

This section covers maintenance tasks for the AMORE website.

- Adding New Projects
  - [To be expanded: Complete workflow from proposal to publication]
- Updating Existing Projects
  - [To be expanded: How to add new data, update results, or revise content]
- Managing the Shiny App Directory
  - [To be expanded: How the search/filter app reads project metadata]

- Version Control Best Practices
  - [To be expanded: Git workflow, branching, and pull requests]
- Reviewing and Approving Changes
  - [To be expanded: Steering committee review process]
- Handling Retractions
  - [To be expanded: Process for removing or updating retracted research]
- Monitoring Site Performance
  - [To be expanded: Checking for broken links, load times, and errors]
- Backup and Recovery
  - [To be expanded: How data is backed up and how to restore if needed]
- Advanced Customization
  - This section covers advanced modifications for experienced users.
- Custom CSS Modifications
  - [To be expanded: How to modify styles.scss for custom designs]
- Adding New Filter Categories
  - [To be expanded: Updating the Shiny app to include new metadata fields]
- Creating New Page Components
  - [To be expanded: Building reusable HTML components for project pages]
- Modifying the Project Template
  - [To be expanded: Updating the base template for all new projects]
- Nettskjema
- netlify
- shinyapps server and user

---

Missing critical content in this document:

1. Emergency contacts - Who to contact if site goes down
2. Rollback procedures - How to undo problematic changes
3. Testing protocols - How to test changes before going live

4. Domain management - How to manage amore-project.org
5. Access management - How to add/remove collaborators