|  |
| --- |
|  |

**Lyfetree**

The tree of life.

Concept

Table of contents

[Requirements 3](#_Toc147348549)

[Landingpage 3](#_Toc147348550)

[Authentication 3](#_Toc147348551)

[“Lyfetree“ page 3](#_Toc147348552)

[Technical structure 4](#_Toc147348553)

[Django framework 4](#_Toc147348554)

[Templates 4](#_Toc147348555)

[URLs 4](#_Toc147348556)

[Models 4](#_Toc147348557)

[Forms 4](#_Toc147348558)

[Settings 4](#_Toc147348559)

# Requirements

## Landingpage

When visiting the website/app, users who are not logged in should be immediately redirected to the registration/login page.

If a user is already logged in or has already done so, he or she will be redirected to the index view (index.html) which - adding on top of the base.html template – includes the personal "lyfetree", i.e. an structured and styled overview of the stored data in the user-specific database (tree-like structure of the data).

## Authentication

Users must be able to register or login on a designated website/view. For the login, a "forgotten password" function should also be considered.

## “Lyfetree“ page

On the personal Lyfetree page, users should be able to interact with their created milestones/memories/achievements. They shall have the possibility to create new milestones via forms and to revise/edit already existing milestones in their database, which should be immediately displayed after posting to the database. They should also be able to delete milestones.

## Database

A tree-like structure is provided for the user-specific database (“journey”). Users can create supercategories ("milestones") when initially creating a database, which are then filled with subcategories (parent/child relationship). For subcategories, in turn, further subcategories can be created.

# Technical structure

## Django framework

### Apps

|  |  |
| --- | --- |
| App | Description |
| journey | This app contains user-specific databases for the display of the Lyfetree |

### Templates

All templates are stored in the “templates” folder. Authorization templates are in “templates/registration”.

|  |  |
| --- | --- |
| Template | Description |
| login.html |  |
| signup.html |  |
| index.html |  |
| base.html |  |

### URLs

All url-patterns are configured in the urls.py of the django project.

|  |  |
| --- | --- |
| URL path | Description |
| “” |  |
| “admin/” |  |
| “login/” |  |
| “logout/” |  |
| “password\_reset/” |  |
| “password\_reset/done” |  |
| “reset/<uidb64>/<token>/” |  |
| “reset/done/” |  |
| “register/” |  |
| “create\_milestone/” |  |
| “edit\_milestone/” |  |
| “rename\_milestone” |  |

### Models

All data models are configured in the models.py file.

|  |  |
| --- | --- |
| Class | Description |
| Tag |  |
| Milestone |  |

### Forms

All forms are configured in the forms.py file.

|  |  |
| --- | --- |
| Form | Description |
| MilestoneCreateForm |  |
| MilestoneEditForm |  |

### Settings

All project settings are configured in the settings.py file.

|  |  |
| --- | --- |
| Setting | Description |
| SECRET\_KEY = config(“SECRET\_KEY”) |  |
| LOGIN\_URL = “login” |  |
| LOGOUT\_REDIRECT\_URL = “login” |  |

## Local setup

* Pipenv + .env-file
* Pip
* Python 3