Project DRAFT & Structure

StudyHub

Collaboration and resource platform for college students.

Overview

- **Purpose:** StudyHub is a web platform designed to help university students share notes, learning materials, and exam preparation resources.
- **Main Goal:** Enhance both individual and collaborative study organization and effectiveness among students.
- **Target Audience:** University students aged 18–30, enrolled in bachelor’s and master’s degree programs.

Main sections

Sections present in the horizontal Navbar (locked) to which you are redirected by clicking on the relevant writing (with the exception of the Home writing which is not directly present in the Navbar since you are redirected to it every time you click directly on the writing with the name of the application located in the Navbar on the left):

Profile

Description: this page contains all the user's personal information along with the documents uploaded, saved, or marked as favorites on the platform.

Structure:

Static page with personal information, consisting of:

- (image) and information such as name and surname, email, academic institute, academic course, academic year
- Section for both notifications and general settings

- Main area with respectively a section regarding the documents uploaded to the platform and one regarding the documents present on the platform and marked as "favorites"

Home

Description: Main page which serves as an introduction to the application in which to welcome the user who will find all the basic information needed to get to know the application and learn how to move within it.

Structure:

It is a scrolling page where under the navbar there is:

- The introduction to the application, with a short and concise explanation of the objective of the application and its functions
- It guides you on how to move within the app by briefly presenting and showing each section and its functioning and purpose, also with the help of embedded representations for clear and practical examples. For instance, horizontal scrolling bar displays key details of an embedded document.

Upload

Description: With this page you upload documents with your notes to the platform.

Structure:

- INTRO with brief description of the section.
- Guidelines on how to correctly upload and how the note document to be uploaded must be created and structured.
- Box where you will be asked to enter information about the document and the content of the document:
  - Document Title
  - Academic Institute
  - Academic year
  - Academic course

    - Subject
    - Author
    - Brief description of the document
    - Button to upload the document

View

Description: With this page you can search the platform for the notes documents you need.

Structure:

- INTRO with brief description of the section.
- Search bar with option to filter by various attributes or needs:
- Academic institution
- Academic year
- Academic course
- Subject
- Author
- Document Appreciation Rating
- The documents that appear with the search will present a preview of all the attributes previously listed
- Horizontal sections, one regarding the most recently used documents and one regarding the most used documents overall

Form

Description: A form to fill out to describe questions, doubts or problems to be resolved and to which you will receive an answer.

Structure:

- Select reference subject
- Enter email

- Section to describe doubt or problem
- Button to send the form to the platform

Footer

In the footer of the web application there will be information such as Contacts, FAQ and Social, etc.

Color palette

- Lemon (#fde910) for background and predominant color;
- White (#FFFFFF) and Black (#000000) for titles and subtitles.

Languages, Framework, Tools...

- **Front-end**: HTML, CSS, Javascript
  - **Bootstrap:** CSS framework for responsive design and layout.
  - **jQuery:** Client-side scripting for DOM manipulation and event handling.
- **Back-end**: Python
  - **Flask:** Backend framework handling routing and business logic.
  - **Jinja2:** Templating engine rendering HTML views with dynamic data.
  - **AJAX:** Asynchronous requests (e.g., document upload, dynamic form updates).

Basic hypothetical project development structure

```
StudyHub/
├── run.py                # Script di avvio dell’app Flask
├── config.py              # Configurazione generale (senza .env)
├── requirements.txt        # Dipendenze Python
├── README.md              # Documentazione progetto
├── uploads/               # Cartella per file caricati dagli utenti
├── app/                  # Package principale (solo codice Python)
```

```
│   ├── __init__.py        # Factory function, registra blueprint
│   ├── main/             # Blueprint 'main': homepage, gestione errori
│   │   ├── __init__.py
│   │   ├── routes.py      # Home, pagine statiche
│   │   └── errors.py      # Gestione errori 404, 500
│   ├── auth/             # Blueprint 'auth': login, registrazione, profilo
│   │   ├── __init__.py
│   │   ├── routes.py      # Login, logout, registrazione, profilo utente
│   │   └── forms.py       # Form di login, registrazione, modifica profilo
│   ├── upload/           # Blueprint 'upload': caricamento documenti
│   │   ├── __init__.py
│   │   ├── routes.py      # Pagina upload, gestione form upload
│   │   └── utils.py       # Funzioni di supporto per upload (es. validazioni)
│   ├── view/             # Blueprint 'view': ricerca e visualizzazione documenti
│   │   ├── __init__.py
│   │   ├── routes.py      # Ricerca, filtri, visualizzazione documenti
│   │   └── utils.py       # Funzioni di supporto per ricerca e filtri
│   ├── form/             # Blueprint 'form': gestione form domande/dubbi
│   │   ├── __init__.py
│   │   ├── routes.py      # Pagina form, invio richieste
│   │   └── forms.py       # Form per domande/dubbi
├── static/               # File statici (CSS, JS, immagini)
│   ├── css/
│   │   └── style.css      # Include palette colori (lemon, bianco, nero)
│   ├── js/
│   │   └── script.js      # Eventuali script JS, AJAX
│   └── images/
│       └── logo.png
└── templates/            # Template HTML suddivisi per blueprint e partials
    ├── partials/         # Template parziali riutilizzabili
    │   ├── _navbar.html    # Navbar inclusa in base.html
    │   └── _footer.html    # Footer incluso in base.html
    ├── main/
    │   ├── base.html       # Template base che include partials di navbar e footer
    │   ├── home.html       # Pagina principale
    ├── auth/
    │   ├── login.html
    │   ├── register.html
    │   └── profile.html    # Pagina profilo utente (info, documenti, notifiche)
    ├── upload/
    │   ├── upload.html     # Pagina upload documenti con istruzioni e form
    ├── view/
    │   ├── search.html     # Pagina ricerca documenti con filtri e risultati
    ├── form/
    │   ├── question_form.html # Form per domande/dubbi
    └── errors/
       └── 404.html       # Pagina errore 404 personalizzata
```

Strumenti e framework indicati e loro ruolo

Tecnologia	Ruolo nel progetto
HTML, CSS, JavaScript	Base per costruire il frontend, layout, stili e interattività.
Bootstrap	Framework CSS per design responsivo e componenti UI pronti, accelera lo sviluppo frontend.
jQuery	Libreria JS per manipolazione DOM e gestione eventi, semplifica AJAX e interazioni client-side.
AJAX	Tecnica per richieste asincrone, utile per upload file, aggiornamenti dinamici senza ricaricare pagina.
Python + Flask	Backend leggero e flessibile per routing, logica applicativa, gestione richieste HTTP.
Jinja2	Motore di template integrato in Flask per generare HTML dinamico con dati dal backend.

Strumento	**Dove si trova nella struttura?**	**Come si usa?**
Bootstrap	Via CDN in template HTML (oppure in `static/css/`)	Inclusione `<link>` e `<script>` nel base.html
jQuery	Via CDN in template HTML (oppure in `static/js/`)	Inclusione `<script>` nel base.html
AJAX	Nei file JS in `static/js/`	Funzioni JS che fanno chiamate asincrone
Jinja2	Nei file HTML in `templates/`	Motore template usato da Flask per renderizzare