

1. عنوان و اطلاعات کلی پروژه

نام پروژه: سامانه مدیریت پایان نامه (Thesis Manager – CLI)

نام و نام خانوادگی: هدی نظرکرده

شماره دانشجویی: 402103239

کد ملی: 0927479605

آدرس ریویزیتوری:

2. خلاصه پروژه

هدف این پروژه ساخت یک سامانه خط فرمان (CLI) برای مدیریت چرخه‌ی پایان‌نامه‌ی کارشناسی ارشد است. سامانه دو نقش اصلی «دانشجو» و «استاد راهنما» را پوشش می‌دهد و عملیات زیر را فراهم می‌کند:

* ورود (Login) دانشجو و استاد.

* ثبت درخواست پایان‌نامه توسط دانشجو نزدیک استاد.

* تأیید/رد درخواست توسط استاد راهنما با در نظر گرفتن ظرفیت.

* ایجاد رکورد پایان‌نامه و انتساب آن به دانشجو/استاد.

* تعیین داور داخلی، داور خارجی و زمان دفاع.

* ثبت نتیجه و نمره دفاع توسط استاد راهنما.

* جست‌وجو در بانک پایان‌نامه‌ها براساس عبارت، سال و استاد.

3. نیازمندی‌ها

* نسخه‌ی Python3.13.5

* کتابخانه‌ها / پکیج‌های استفاده‌شده: dataclasses, typing, hashlib, json, os, datetime, getpass.

* سیستم عامل هدف: همه (Windows / macOS / Linux)

4. ساختار پروژه (Project Structure)

/project-root

app.py — | # نقطه‌ی ورود برنامه (CLI)

/core — |

models.py — | | # تعریف داده‌مدل‌ها: Student, Professor, Thesis

utils.py — | | # توابع کمکی (hash، تولید ID، ورودی غیر خالی)

store.py — | | # خواندن/نوشتن فایل‌های JSON در پوشه /data

db.py — | | # لایه دسترسی داده + seed اولیه

services.py — | | # منطق کسب‌وکار (login، workflow پایان‌نامه)

/data — | | # پایگاه داده فایل — محور (JSON)

students.json — | |

professors.json — | |

theses.json — | |

5. توضیح کلاس‌ها و توابع

Student (1)

ویژگی‌ها:

*student_id: str

*name: str

*year: int

*semester: str

*password_hash: str

*advisor_id: Optional[str] = None

*thesis_id: Optional[str] = None

*thesis_request_status: str = "none" → one of none, requested, approved, rejected

Professor (2)

ویژگی‌ها:

*professor_id: str, name: str, password_hash: str

*capacity_advisor: int (ظرفیت راهنمایی)

*capacity_judge: int (ظرفیت داوری)

*fields: List[str] (حوزه‌های تخصصی)

* current_advisees: List[str] (لیست student_id دانشجویان تحت راهنمایی)

* current_judgings: List[str] (لیست thesis_id های در حال داوری)

Thesis (3

ویژگی‌ها:

- *thesis_id: str, title: str, keywords: List[str], abstract: str
- *fulltext_link: Optional[str]
- *student_id: str, advisor_id: str
- *year: int, semester: str
- *judge_internal_id: Optional[str] = None
- *judge_external: Optional[str] = None
- *defense_date: Optional[str] = None
- *status: str = "draft ← یکی از draft, scheduled, defended
- *grade: Optional[str] = None, result: Optional[str] = None

Utils (utils.py)

- *prompt_nonempty(msg): دریافت ورودی اجباری از کاربر.
- *next_id(prefix, existing_ids): تولید شناسه متوالی مانند T1, T2, ...
- *sha256(text): تولید هش SHA-256 برای گذرواژه.

Store (store.py)

- *load_json(filename, default): بارگذاری امن JSON از ./data
- *save_json(filename, payload): ذخیره JSON در ./data

DB (db.py)

* کلاس DB با نگهداری دیکشنری‌های students, professors, theses.

* متدها: load(), save(), ensure_seed().

* Seed اولیه (در اولین اجرا):

* دانشجویها:

* Alice / 40110001 (رمز: 1234)

* Bob / 40110002 (رمز: 1234)

* استادها:

* Prof. Smith / 9001 (رمز: admin1) ظرفیت راهنما 5، ظرفیت داور 10، حوزه‌ها: AI, IR

* Prof. Johnson / 9002 (رمز: admin2) ظرفیت راهنما 5، ظرفیت داور 10، حوزه‌ها: DB, Software

Services (services.py)

*login_student(student_id, password)

*login_professor(professor_id, password)

*student_request_thesis(student_id, advisor_id, title, keywords, abstract)

* بررسی ظرفیت استاد (در زمان درخواست).

* ایجاد Thesis جدید با سال/ترم دانشجو و وضعیت اولیه.

*professor_review_requests(professor_id)

*** لیست دانشجویانی که وضعیت درخواستشان requested است.**

***professor_approve_request(professor_id, student_id, approve)**

*** در صورت تأیید: تغییر وضعیت دانشجو به approved و افزودن به current advisees.**

*** در صورت رد: بازنشانی advisor_id و حذف thesis موقت.**

***assign_judges_and_schedule(professor_id, thesis_id, judge_internal_id, judge_external, defense_date)**

*** فقط استاد راهنما به کنترل ظرفیت داور داخلی مجاز است.**

*** تنظیم داوران، تاریخ دفاع و تغییر وضعیت به scheduled.**

***record_defense_result(professor_id, thesis_id, grade, result)**

*** فقط استاد راهنما مجاز است. وضعیت به defended تغییر می کند.**

***search_thesis_bank(query: str = "", year: Optional[int] = None, advisor: Optional[str] = None)**

*** جست و جو روی عنوان، کلیدواژه ها، نام استاد و دانشجو.**

برنامه ی اصلی (app.py)

منوهای CLI برای دانشجو و استاد را فراهم می کند:

*** منوی دانشجو: درخواست پایان نامه، مشاهده وضعیت، جست و جو.**

*** منوی استاد: بررسی درخواست ها، تعیین داور و زمان دفاع، ثبت نمره، جست و جو.**

6. جزئیات پیاده‌سازی

1. تأیید ظرفیت‌ها

* هنگام ثبت درخواست، اگر `len(adv.current_advisees) >= capacity_advisor` باشد، درخواست رد و خطای «Advisor capacity full» پرتاب می‌شود.

* هنگام تعیین داور داخلی، اگر `len(j.current_judgings) >= capacity_judge` باشد، خطای «Internal judge capacity full» داده می‌شود.

2. حفظ سازگاری داده

* تمامی تغییرات پس از عملیات موفق با `db.save()` در فایل‌های JSON ذخیره می‌شود.

* در رد درخواست، رکورد پایان‌نامه موقت حذف و اشاره‌های دانشجو بازنشانی می‌شود.

3. امنیت ساده

* گذرواژه‌ها به صورت SHA-256 ذخیره می‌شوند. (نمونه‌ها در Seed مشخص شده‌اند.)

* دسترسی به عملیات حساس (تعیین داور / ثبت نمره) محدود به استاد راهنمای همان پایان‌نامه است (PermissionError).

4. جست‌وجو

* متن قابل جست‌وجو از ترکیب عنوان، کلیدواژه‌ها، نام استاد و دانشجو ساخته می‌شود و با `query` کوچک‌سازی Trim می‌شود.

7. نحوه اجرا

1. مطمئن شوید ساختار پوشه مانند بخش «ساختار پروژه» است.

```
python -m venv .venv  
  
# Windows: .venv\Scripts\activate  
  
# macOS/Linux: source .venv/bin/activate
```

3. اجرای برنامه:

```
Python app.py
```

4. در اولین اجرا، داده‌های اولیه ساخته می‌شود. سپس از منوی اصلی نقش را انتخاب کنید:

* دانشجو → ورود با:

ID: 40110001 | Password: 1234 *

ID: 40110002 | Password: 1234 یا *

* استاد → ورود با:

ID: 9001 | Password: admin1 *

ID: 9002 | Password: admin2 یا *

نمونه سناریوهای اجرا:

1. دانشجو درخواست پایان‌نامه ثبت می‌کند

* ورود به منوی دانشجو → گزینه «Request Thesis».

* وارد کردن Advisor ID (مثلاً 9001)، «Keywords، Title» و «با کاما» و «Abstract».

* در صورت موفقیت، شناسه‌ی پایان‌نامه‌ی ایجادشده (مثل T1) چاپ می‌شود.

2. استاد درخواست‌ها را بررسی و تأیید می‌کند

* استاد راهنما وارد می‌شود → «Review Requests» → لیست دانشجویان در وضعیت requested نمایش داده می‌شود.

* با انتخاب Student ID و پاسخ y، دانشجو به approved تغییر می‌کند و به لیست راهنمایی افزوده می‌شود.

3. زمان‌بندی دفاع و تعیین داورها

* استاد راهنما → «Assign Judges» → وارد کردن Thesis ID, Internal Judge ID (مثلاً 9002), نام داور خارجی و تاریخ دفاع.

4. ثبت نتیجه دفاع

* استاد راهنما → «Record Grade» → وارد کردن Thesis ID, Grade و Result.

5. جست‌وجو در بانک پایان‌نامه‌ها

* از منوی اصلی یا منوی نقش‌ها، گزینه «Search Theses» → وارد کردن کلیدواژه (مثلاً AI 2024 Smith).

مدیریت خطا و پیام‌ها:

* خطاهای ظرفیتی و مجوز با پیام‌های قابل فهم چاپ می‌شوند (ValueError, PermissionError).

* ورودی‌های اجباری با prompt_nonempty کنترل می‌شود.

* عملیات فایل‌ها در برابر فایل خراب با default ایمن‌سازی شده‌است.