# داکیومنت پروژه سیستم پایان نامه

1. **عنوان و اطلاعات کلی پروژه**
   * **نام پروژه:** سیستم مدیریت پایان‌نامه (Thesis Management System)
   * **نام و نام خانوادگی: فاطمه پورحسین**
   * **شماره دانشجویی: 403103210**
   * **آدرس ریپوزیتوری: https://github.com/FatemehPourhossein/thesis\_management.git**
2. **خلاصه پروژه (Project Overview)**

**هدف پروژه:**

هدف اصلی این پروژه مدیریت تمام مراحل یک پایان‌نامه از ثبت درخواست تا ارزیابی نهایی است، به نحوی که دانشجو و استاد بتوانند فرآیند را به راحتی مشاهده و مدیریت کنند.

**عملکرد کلی برنامه**

* **دانشجو:**
  + ثبت‌نام و ورود با کد دانشجویی و رمز عبور
  + مشاهده لیست دروس قابل اخذ
  + ثبت درخواست پایان‌نامه و مشاهده وضعیت آن
  + ارسال مجدد درخواست رد شده
  + ثبت درخواست دفاع پس از تایید پایان‌نامه
* **استاد:**
  + ورود با کد استادی و رمز عبور
  + بررسی درخواست‌های دانشجویان و تایید یا رد آن‌ها
  + مدیریت دفاع: تعیین تاریخ و داوران جلسه دفاع
  + ثبت نمره نهایی و مختومه کردن پایان‌نامه

**آپشن‌های اضافی پیاده‌سازی شده**

* هش کردن رمزهای دانشجو و استاد برای امنیت بیشتر
* ثبت تاریخ‌ها به صورت خودکار در JSON (تاریخ ثبت درخواست، تایید، دفاع)
* امکان ارسال مجدد درخواست ردشده توسط دانشجو

**امکانات در نظر گرفته شده ولی پیاده نشده**

* داشبورد دانشجو و استاد
* سیستم پیام و نوتیفیکیشن
* API برای عملکردهای اصلی
* پردازش هوش مصنوعی روی متون و فایل‌ها
* تولید صورت جلسه نهایی خودکار

## نیازمندی‌ها (Requirements)

* **نسخه Python: 3.10 یا بالاتر**
* **کتابخانه‌ها / پکیج‌ها:**
* json برای ذخیره و خواندن داده‌ها
* os برای بررسی مسیر فایل‌ها
* hashlib برای هش کردن رمز عبور
* datetime برای مدیریت تاریخ‌ها

1. ساختار پروژه (Project Structure)

 **data/**: شامل فایل‌های JSON است که داده‌ها در آن‌ها ذخیره می‌شوند.

 **files/**: پوشه‌ای برای ذخیره تصاویر و فایل‌های PDF مربوط به پایان‌نامه‌ها.

 **main.py**: نقطه ورود برنامه و مدیریت منوها.

 **student.py و teacher.py**: کلاس‌ها و متدهای مربوط به دانشجو و استاد.

 **storage.py**: توابع کمکی برای خواندن و ذخیره داده‌ها.

### توضیح کلاس‌ها و توابع (Classes & Functions)

### storage.py

* load\_json(path): اگر فایل وجود نداشته باشد، لیست خالی برمی‌گرداند؛ در غیر این صورت داده‌های JSON را برمی‌گرداند.
* save\_json(path, data): داده‌ها را با فرمت JSON در مسیر مشخص ذخیره می‌کند.

### student.py

**کلاس Student**

* **Attributes:**
  + **code: کد دانشجویی**
  + **name: نام کامل**
  + **password: رمز هش شده**
  + **email: ایمیل اختیاری**
* **Methods:**
  + **register(): ثبت‌نام دانشجو با جلوگیری از تکرار کد**
  + **login(): ورود دانشجو با بررسی رمز هش‌شده**
  + **list\_courses(): نمایش لیست دروس با ظرفیت‌ها**
  + **request\_thesis(): ثبت درخواست پایان‌نامه با بررسی ظرفیت و تکرار**
  + **view\_requests(): مشاهده وضعیت درخواست‌ها**
  + **resubmit\_request(): ارسال مجدد درخواست ردشده**
  + **request\_defense(): ثبت درخواست دفاع بعد از تایید پایان‌نامه**

**ویژگی‌ها و محدودیت‌ها:**

* **تمام تغییرات در thesis.json ذخیره می‌شوند.**
* **وضعیت پایان‌نامه‌ها به صورت متنی ذخیره می‌شود (pending, approved, rejected, defense\_pending, defense, done).**
* **امکان ثبت دفاع تنها بعد از گذشت حداقل ۹۰ روز از تایید پایان‌نامه وجود دارد، با امکان رد درخواست توسط دانشجو قبل از ادامه.**

teacher.py

**کلاس Teacher**

* **Attributes:**
  + **code, name, password, email, cap\_sup, cap\_judge**
* **Methods:**
  + **login(): ورود استاد**
  + **check\_requests(): بررسی و تایید یا رد درخواست‌های دانشجو**
  + **manage\_defense(): تعیین تاریخ و داوران جلسه دفاع**
  + **judge\_score(): ثبت نمره نهایی و انتقال داده‌ها به defended\_thesis.json**

**ویژگی‌ها و محدودیت‌ها:**

* **ظرفیت راهنمایی و داوری اساتید به صورت پیش‌فرض ۵ و ۱۰ است.**
* **داورها پس از ثبت نمره، ظرفیت آن‌ها آزاد می‌شود.**

main.py

* نمایش منوی اصلی و منوهای دانشجو و استاد
* فراخوانی متدهای مربوطه بر اساس انتخاب کاربر

1. جزئیات پیاده‌سازی (Implementation Details)

* **هش رمز عبور: با SHA-256 و کتابخانه hashlib**
* **ذخیره داده‌ها: در JSON برای سادگی و قابلیت ویرایش دستی**
* **مدیریت ظرفیت: بررسی ظرفیت درس و پایان‌نامه قبل از ثبت درخواست**
* **تاریخ‌ها: ثبت خودکار تاریخ درخواست و دفاع**
* **وضعیت پایان‌نامه‌ها: چند مرحله‌ای برای نمایش کامل فرآیند**

1. نحوه اجرا (How to Run)
2. نصب Python 3.10 یا بالاتر
3. کلون کردن ریپو
4. اجرای main.py
5. انتخاب نقش دانشجو یا استاد و استفاده از منوها

### نمونه خروجی (Sample Output)

### ثبت‌نام دانشجو

مثلا:

کد دانشجویی جدید: 401102832

نام کامل: علی رضایی

رمز عبور: 1234  
ایمیل(اختیاری): ali@email.com  
ثبت نام انجام شد.

### ورود استاد و تایید درخواست

مثلا:

کد استادی: T101

رمز عبور: \*\*\*\*

خوش آمدید استاد کریمی

دانشجو: 401 درس: CSE101 وضعیت: pending

کد درس برای تایید/رد: CSE101

کد دانشجو: 401

تایید یا رد؟ (a/r): a

درخواست تایید شد

### ثبت نمره دفاع

مثلا:

دانشجو: 401 درس: CSE101 تاریخ: 1404-03-10

کد دانشجو: 401

کد درس: CSE101

نمره بده (0 تا 20): 18

نمره ثبت شد و پایان نامه مختومه شد.

1. مشکلات و پیشنهادات (Known Issues & Improvements)

* امکان تغییر استاد تا قبل از دفاع هنوز پیاده نشده
* سیستم پیام و نوتیفیکیشن بین دانشجو و استاد طراحی نشده
* داشبورد دانشجو و استاد می‌تواند اضافه شود
* API و پردازش هوش مصنوعی می‌تواند برای بررسی خودکار فایل‌ها و متون استفاده شود
* صورت‌جلسه نهایی داور داخلی می‌تواند تولید شود و در فولدر thesis ذخیره شود

**نکته:** این پروژه اولین پروژه جدی من بود و تلاش کردم تمام بخش‌های پایه را با رعایت شی‌گرایی و ماژولاریتی پیاده کنم...گرچه احتمالا نقص زیاد خواهد داشت و پروژه کامل و تمیز نشد... . همچنین شرایط طوری پیش نرفت که بارگزاری پایان نامه و عکس را به درستی و کامل اجرا کنم