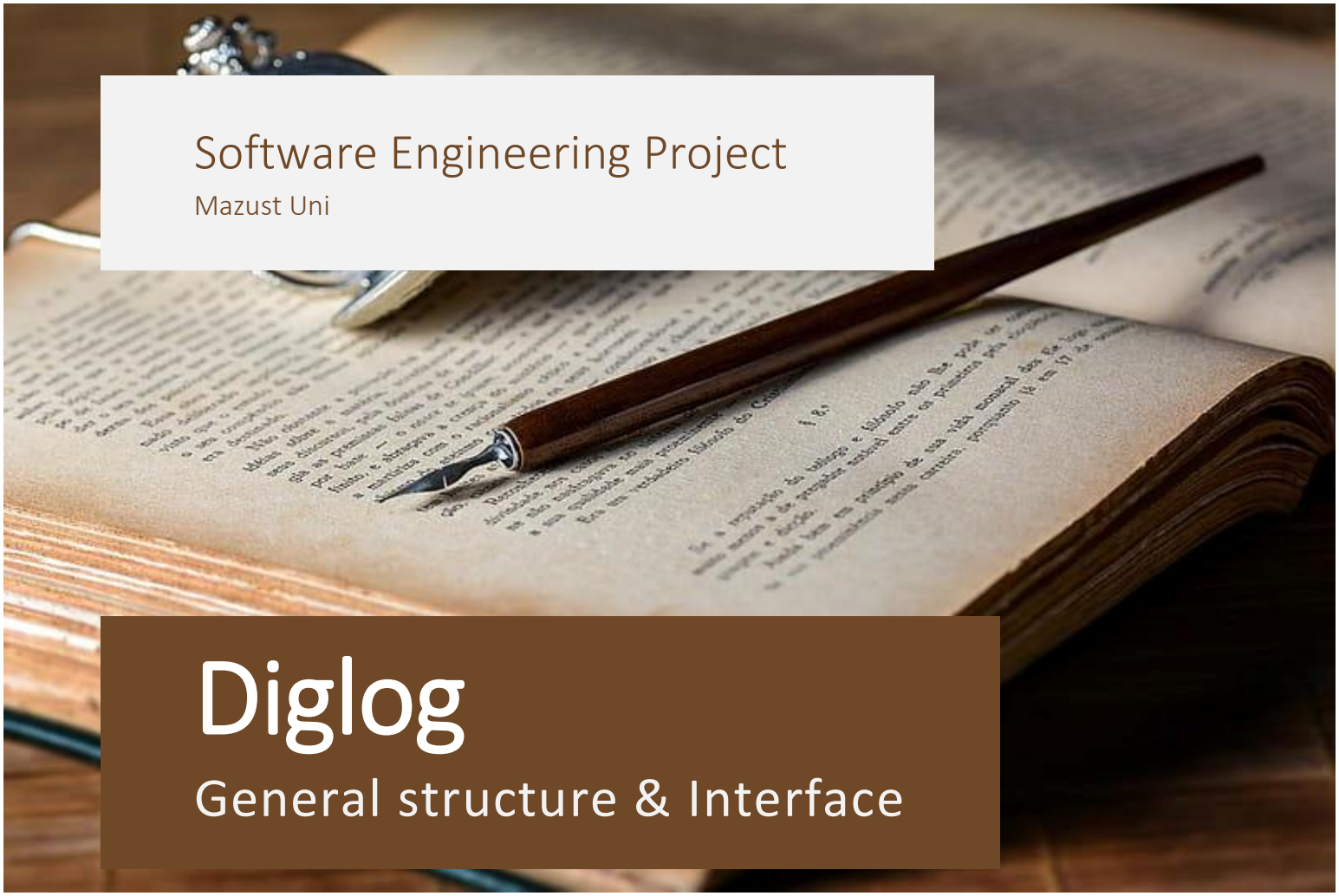

Software Engineering Project

Mazust Uni

Diglog

General structure & Interface

01 توضیحات کلی و ساختار پروژه

02 بخش های مختلف پروژه و جزئیات آن

پروژه مهندسی نرم افزار

وبلاگ (با زبان پایتون – فریموورک جنگو)

استاد: دکتر ریاحی

دانشجو: محمد موسی پور

دانشگاه علم و فناوری مازندران

توضیحات کلی پروژه Diglog

این پروژه یک وبلاگ کامل بر پایه وب می باشد که بک اند آن با فریمورک جنگو (Django) که یک فریمورک به زبان پایتون، و فرانت آن با HTML، CSS، Bootstrap و JavaScript است. کدنویسی بک اند این پروژه، ساختار شی گرا دارد و با استفاده از Class Base Views و Generic Views توسعه داده شده است، همچنین پروژه از MVT پشتیبانی میکند که ساختار استاندارد یک اپلیکیشن برپایه وب است؛ به طور خلاصه به این شکل کار میکند که درخواست ارسالی از طرف کاربر به توابع اپ ارسال میشود و پس از پردازش، به دیتابیس ارجاع داده میشود تا اطلاعات مورد نظر کوئری کشی شود و سپس اطلاعات به تمپلیت جهت اجرا فرستاده میشود.

بخش های مختلف پروژه و جزئیات آن

یک وبسایت چندین بخش دارد، فرانت اند، بک اند و دیتابیس، بخش فرانت مربوط به طراحی و اینترفیس سایت میباشد و با کاربر (کلاینت) سر کار دارد. بخش بک اند برای مدیریت دیتای ورودی و خروجی است و همچنین توابع عملکردی اصلی در این بخش پیاده سازی میشوند؛ فریموورک جنگو ساختار Oriented Object دارد و تماما با شی گرایی کار میکند؛ بخش بعدی هم دیتابیس وبسایت است که برای ذخیره دیتا به کار میرود. *آنالیز این مدل به عهده خودم بوده و نقشه و طرح کلی را به طور کامل خودم انجام داده ام و نیاز یک کاربر را پیش بینی و پیاده سازی کرده ام.*

بخش ظاهری وبسایت

لیست همه پست ها (مرتب سازی بر اساس ابتدا جدید ترین)

نوار سایت

- گزینه هایی مثل ساخت پست و لیست دسته بندی ها، تنظیمات پروفایل و ...

قابلیت سرچ

صفحه بندی

بخش دسترسی کاربر به های وبسایت

ثبت نام کاربر

- ساخت اکانت
- ورود کاربر
- نام کاربری، پسورد، ایمیل و ...

ساخت پست

- عنوان
- نام نویسنده
- تصویر هدر
- تاریخ انتشار (روز و ساعت انتشار)

- توضیحات کوتاه
- متن اصلی پست (Django CKEditor)
  - متن با امکاناتی مثل قابلیت خط کشیدن زیر کلمات، بلد کردن و ...
  - قابلیت پرینت جداگانه
  - نوشتن متن به صورت سورس
  - قالب و تمپلیت بندی فرم متن پست
  - ثبت عکس در متن (حتی با دادن url)
  - اضافه کردن نمودار، نماد، اموجی، نقل قول و ...
- ثبت دسته بندی پست
  - اضافه کردن دسته بندی
- تاریخ آخرین آپدیت

قابلیت ادیت کردن پست

قابلیت حذف کردن پست

- پیام هشدار حذف پست

قابلیت لایک کردن پست برای کاربران وارد شده

دیدن ویو یک پست (قابلیت مشاهده کاربری که پست را دیده از طرف ادمین و پنل ادمین (تاریخ و ساعت دید + یوزر نیم و بقیه مشخصات))

قابلیت ثبت کامنت برای کاربران وارد شده

مشخصات و پروفایل نویسنده، زمان انتشار پست و ...

پروفایل کاربر

نام، نام خانوادگی، نام کاربری، بیوگرافی، لینک های مفید (وبسایت، اینستاگرام و ...)

- این مشخصات میتوانند فقط به دست خود کاربر آپدیدت شوند
- قابلیت آپدیت کردن پسورد و تغیر آن

تاریخ آخرین ورود به سایت، تاریخ آخرین پیوستن به سایت

پنل دسترسی ادمین

دسترسی به تمام پست ها

دسترسی به تمام کاربران و دسترسی های آنان

دسترسی به اطلاعات (نام کاربری و ساعت و تاریخ) ویو یک پست و مشاهده کسی که پستی را دیده است

فیلتر و مرتب سازی اطلاعات

ورود جداگانه ادمین

دیتابیس

ساختار دیتابیس به صورت چند جدول است که ارتبطات مختلفی با هم دارند. فیلد های دیتابیس این وبسایت بر اساس نیاز کاربر و طراحی برنامه ساخته و پیاده سازی شده اند. فیلد های این جداول به صورت زیر است:

Post Table
Title = String (max_length=255 char)
Author = Foreign Key (to **User** table, CASCADE)
Header Image = Image Field (null able)
Short Description = String (max_length=350)
Body Text = Rich Text Field (Base on django-ckeditor)
Category = String (max_length=255 characters)
Published Date = Date Time Field (Auto generate)
Slug = Slug Field (Auto create – Based on Title Field)
Likes = Many to Many (to **User** table)
Update Date = Date Time Field (Changes by updating the post)

Profile Table
User = One to One Field (to **User** table)
Biography = Text Field (max_length=500 characters)
Profile Picture = Image Field (null able -> There is a default profile picture)
Web Link = String (max_length=255, null able)
Instagram Link = String (max_length=255, null able)
Twitter Link = String (max_length=255, null able)
Slug = Slug Field (Auto create – Based on Title Field)

Comment Table
Post = Foreign Key (to **Post** table, CASCADE)
Name = String (Auto Field based on first name and last_name of authenticated user)
Body = Text Field
Date = Date Time Field (Auto generate)

Category Table
Category Name = String (max_length=255 characters)
Related to posts by python list in 'category_adder.py'.

جدول User از جدل های پیشفرض در django.contrib.auth.models می باشد و نیازی به پیاده سازی دوباره آن نیست:

User Table
Username = String
First Name = String
Last Name = String

Is Staff = یک متغیر بولین به معنی اینکه آیا به دسترسی های ادمین دسترسی دارد یا نه

Is Active =

یک متغیر بول که پیشفرض فعال است و برای فعال کردن اکانت است و به طور مثال میتوان آن را در آینده بسط داد (برای ارسال ایمیل فعالسازی)

Is Super User = این هم یک متغیر بولین است و فقط ادمین و مالک اصلی این قابلیت را دارد، همچنین این کاربر میتواند دسترسی بقیه ادمین ها را تغییر دهد و آن ها را ویرایش کند

Last Login = آخرین ورود به اکانت کاربر – فیلد تاریخ و ساعت

Join Date = تاریخ پیوستن به سایت که از نوع فیلد تاریخ است

تمام چیزی که از دیتابیس این پروژه نیاز بود، گفته شد، این دیتابیس بر پایه SQLite است که به احتمال بالا در آینده به My SQL تغییر پیدا کند.

ساختار کلی دیتابیس به این فرم است: