SCRUM BOOK

SUBMITTED BY

NIHILA M J

(NCE22MCA-2032)

SUBMITTED TO

ASHISH L

(ASSISTANT PROFESSOR)

PROJECT TOPICS

SUBMITTED ON

29 JAN 2024

TOPIC 1:

E-SCRAP WEBSITE

The e-scrap website is your go-to hub for all things related to electronic waste management. Offering a wealth of resources and services, we cater to individuals and businesses alike seeking responsible solutions for e-waste disposal. Dive into informative articles, guides, and convenient recycling options tailored to your needs. Join us in our commitment to environmental sustainability and contribute to a cleaner, greener future by effectively managing electronic waste.

TOPIC 2:

VINTAGE VEHICLE RENTAL WEBSITE

A vintage vehicle rental website is an online platform that offers services related to renting classic and antique automobiles for various purposes, such as weddings, events, photo shoots, or personal enjoyment. These websites typically showcase a collection of meticulously maintained vintage cars, ranging from iconic models like vintage Cadillacs, Rolls Royces, or Mustangs to charming VW buses or classic European sports cars. Customers can browse through the available fleet, check availability, and make reservations directly through the website. Additionally, these websites may provide information about rental rates, rental terms, and contact details for inquiries or bookings.

TOPIC 3:

SLEEP, HEALTH AND LIFESTYLE PREDICTOR WEBSITE

A sleep, health, and lifestyle predictor website is an online platform that utilizes data analysis, algorithms, and user input to provide personalized insights and predictions related to sleep quality, overall health, and lifestyle habits. Users typically input information such as sleep patterns, diet, exercise routines, stress levels, and other relevant factors. The website then processes this data to generate predictions or recommendations regarding various aspects of the user's health and lifestyle, such as sleep quality, risk of certain health conditions, optimal exercise routines, dietary recommendations, and strategies for stress management. These predictions and recommendations are often based on scientific research, statistical analysis, and machine learning techniques. The goal of such websites is to empower users to make informed decisions about their health and lifestyle habits, ultimately leading to improved well-being and quality of life.

ABSTRACT

ON

SLEEPWELL

SLEEP, HEALTH AND LIFESTYLE PREDICTOR WEBSITE

SUBMITTED ON

9 FEB 2024

RESEARCH AREA

This project involves the development of a website that predicts users' sleep health and lifestyle based on a dataset using machine learning techniques.

PROBLEM STATEMENT

Presently, a significant number of individuals encounter challenges in achieving restful sleep. Consequently, many suffer from sleep disorders and other health-related complications. It is essential to recognize that quality sleep is intricately linked to one's lifestyle choices.

EXISTING SYSTEM

Numerous sleep tracking devices and applications are accessible on the internet. However, most of them focus on monitoring sleep duration and providing an overall assessment of sleep quality. Many sleep trackers primarily focus on total sleep time without delving into the different sleep disorders and sleep quality.

PROPOSED SYSTEM

This comprehensive project encompasses the meticulous analysis of a diverse sleep, health and lifestyle dataset. Leveraging machine learning techniques, we have developed an innovative application capable of predicting stress levels. Beyond stress prediction, the application tailors personalized sleep schedules based on user-provided information, thereby promoting overall well-being, and fostering healthier sleep habits

OBJECTIVES

The main objective is to create a website which can perform the following features

- To develop a website
- Sleep health analysis
- Lifestyle factor analysis
- Cardiovascular health analysis
- Sleep disorder analysis
- To provide personalized sleep schedules based on all the information

REQUIREMENTS

HARDWARE REQUIREMENTS:

PC or Laptop

- Processor i5 or more
- 4GB RAM

SOFTWARE REQUIREMENTS:

OS: Windows 10 or 11

IDE: VS code

DATABASE: PostgreSQL

FRAMEWORK: Django Framework

LANGUAGES:

- Python for Backend
- HTML, CSS, JS & Jinja for frontend

MODULES

ADMIN

This module encompasses the administration functions and systems within the overall system, including data handling and user management.

USER

This module encompasses user registration, login, and all other user-related activities.

HEALTH RECORD

This module mainly encompasses the Dataset and the user data.

MACHINE LEARNING

This module is related to the codes of Machine Learning and other functions related to it.

SCRUM REVIEW

SLEEPWELL

SUBMITTED ON

27 FEB 2024

MODULES DESCRIPTION

ADMIN

This module encompasses all administrative functions, superuser activities, and admin sites, along with the associated database tables essential for the seamless operation of a website. Django automatically generates this module during the project creation process. Developers can establish a superuser by executing the following command in the project's main directory:

```
py manage.py create superuser
```

Subsequently, developers have the capability to specify the username, password, and email address. These details can be modified later by accessing the admin site.

Additionally, the module incorporates numerous built-in functions provided by Django to facilitate rapid, straightforward, and efficient development and utilization of the website. The principal database tables associated with this module include:

1. auth_group
2. auth_group_permissions

3. auth_permissions
4. auth_user
5. auth_user_groups
6. auth_user_user_permissions

SLEEPWELL

This is the core module which contains basic and overall functions. Below mentioned files are present in this module

1. settings.py:

Within the confines of your Django project, settings.py serves as a pivotal file housing essential configuration. It encompasses database setup, static file directives, middleware configurations, and other project-specific settings. This file offers a platform for tailoring various facets of your Django application to meet specific requirements.

2. urls.py:

Playing a pivotal role in your Django project, urls.py is dedicated to delineating URL patterns. By establishing mappings between URL patterns and views, this file enables Django to discern the appropriate view to handle distinct HTTP requests. Routing logic is articulated within this file, outlining how URLs are associated with specific views and controllers.

3. wsgi.py:

An acronym for Web Server Gateway Interface, wsgi.py facilitates the exposure of your Django application to a WSGI server. WSGI serves as the standard interface bridging web servers and Python web applications. It establishes a conduit for external web servers, such as Apache or Nginx, to interact with your Django application seamlessly.

4. asgi.py:

Abbreviated from Asynchronous Server Gateway Interface, asgi.py serves a parallel purpose to wsgi.py. This file is instrumental in exposing your Django application to an ASGI server, which supports asynchronous communication. ASGI proves particularly adept at managing prolonged connections and efficiently handling concurrent requests.

These files stand as integral constituents of a Django project, typically residing at the topmost tier of the project directory. They play indispensable roles in configuring settings, delineating URL patterns, and establishing connections between your Django application and web servers via WSGI or ASGI protocols.

MAINAPP

This module encompasses the functionality pertaining to the Homepage, About page, and a base HTML template that serves as an extension for other HTML files within the website.

It includes the organization of static and templates directories, housing CSS and JavaScript files, as well as About, Home, and base HTML files, respectively.

Within the module, views.py and urls.py contain the code responsible for rendering the Homepage (including the index page) and the About page. At present, this module does not incorporate any models or database tables.

HEALTHPROFILEAPP

The module primarily manages user profiles and health records, encompassing functionalities related to sign-in, sign-up, sign-out, and dashboard interactions.

Within the project structure, the `urls.py` and `views.py` files handle routing and processing of requests pertaining to authentication and user interface navigation.

HTML templates corresponding to the aforementioned functionalities are stored within the `templates` directory.

Form definitions, including`OccupationForm`, `HealthProfileSignUpForm`, and `HealthProfileSignInForm`, are defined in `forms.py`, with Django facilitating the automatic rendering of HTML webpages based on these forms in response to relevant requests.

The data model layer, defined in `models.py`, encapsulates the structure and relationships of entities such as `Occupations`, `HealthProfile`, and `HealthRecord`.

Administrative functions for managing these entities are housed within `admins.py`, facilitating interaction with these models via the Django admin site. This module governs the administration of three distinct database tables: `occupations`, `Health profile`, and `health record`.

DATABASE TABLES

- auth_group
- auth_group_permissions
- auth_permissions
- auth_user
- auth_user_groups

- auth_user_user_permissions
- Django_admin_log
- Django_migrations
- Django_content_type
- Django_sessions
- Healthprofileapp_occupation
- Healthprofileapp_healthprofile
- Healthprofileapp_healthrecord

LANGUAGES

FRONTEND

- HTML
- CSS
- JS
- JINJA

BACKEND

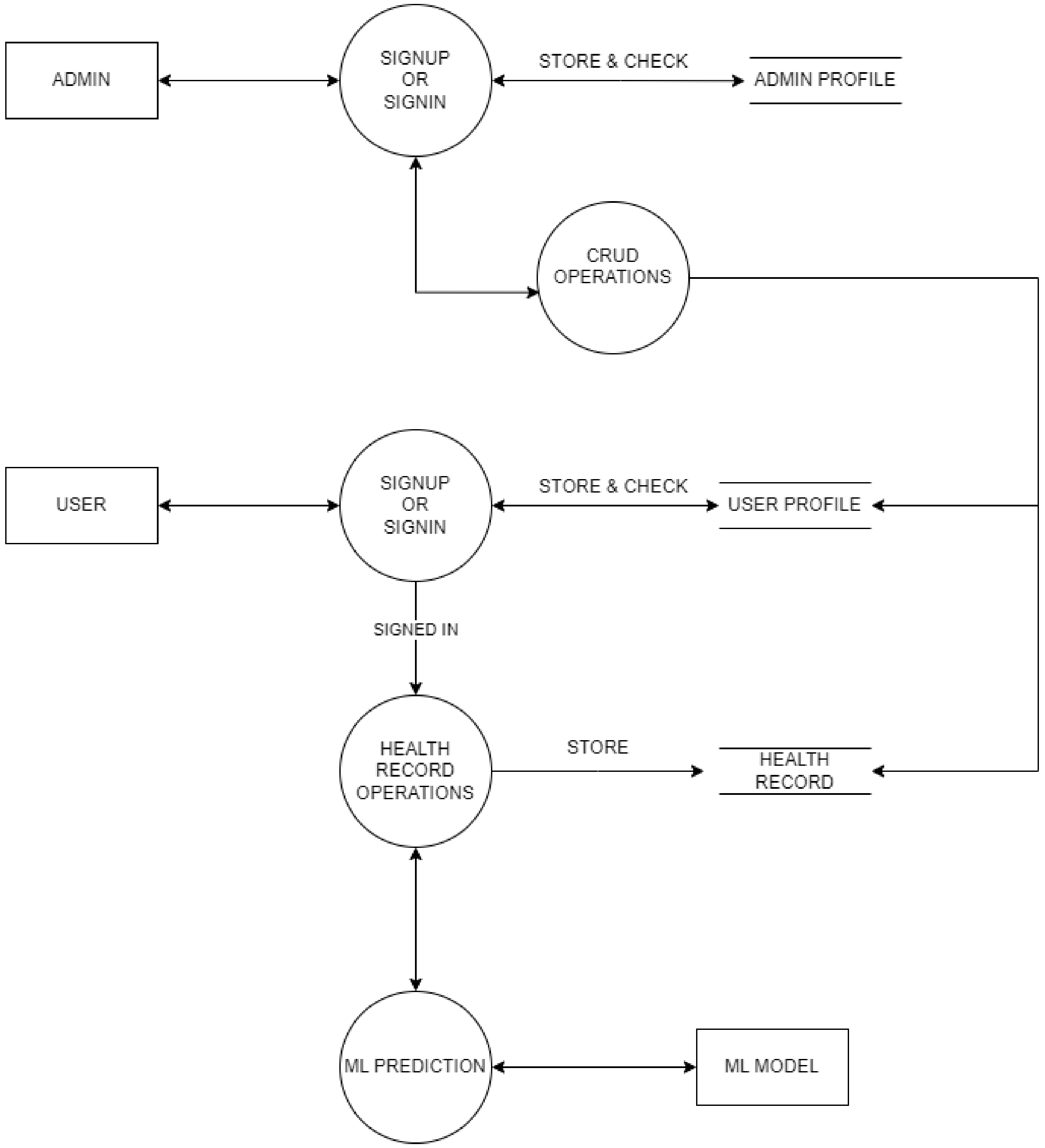
- PYTHON

FRAMEWORK

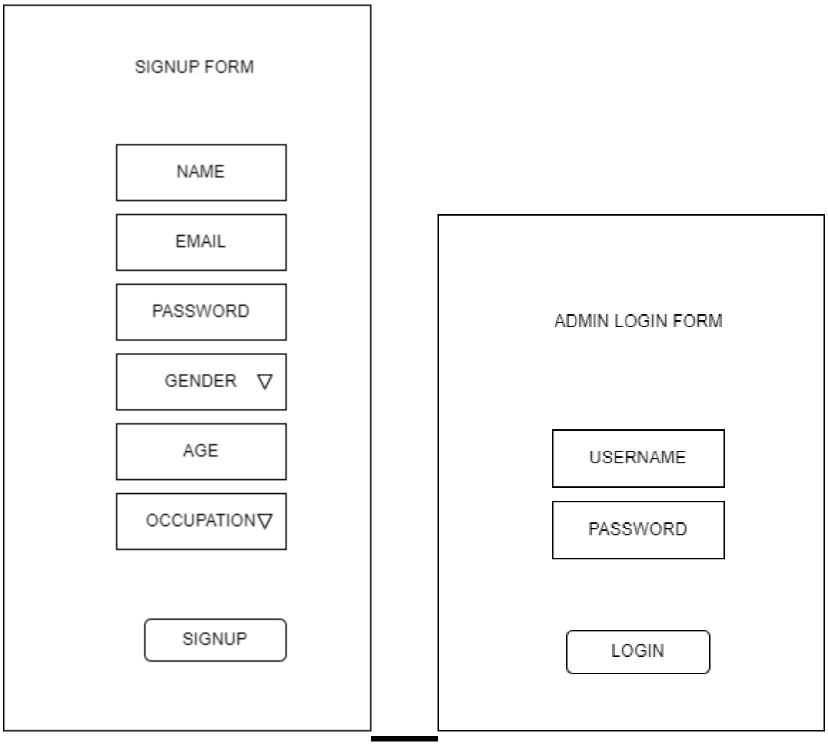
DJANGO

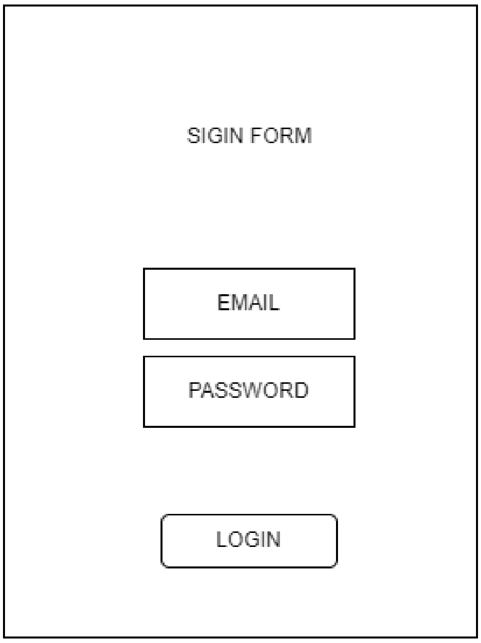
Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It provides a robust set of tools and libraries for building web applications, simplifying common web development tasks such as URL routing, database interaction, and form handling. Django follows the Model-View-Controller (MVC) architectural pattern, with models representing the data structure, views handling user interface logic, and templates managing the presentation layer. Its "batteries-included" philosophy means that many common features, such as authentication, admin interfaces, and session management, are built-in and easily configurable. Django emphasizes reusability, modularity, and the principle of "Don't Repeat Yourself" (DRY), making it a popular choice for developers seeking to create scalable and maintainable web applications.

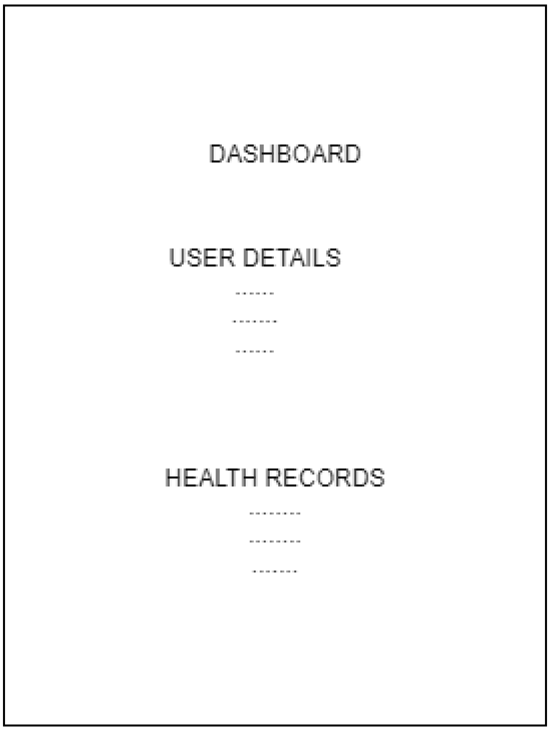
DFD


ADMIN
SIGNUP OR SIGNIN
STORE & CHECK
ADMIN PROFILE
CRUD OPERATIONS
USER
SIGNUP OR SIGNIN
STORE & CHECK
USER PROFILE
SIGNED IN
HEALTH RECORD OPERATIONS
STORE
HEALTH RECORD
ML PREDICTION
ML MODEL

USER INTERFACE







DASHBOARD

USER DETAILS
......
....
......

HEALTH RECORDS
....
....
....

GIT

All codes and documents are committed and upload to below GitHub repository.