|  |  |
| --- | --- |
|  | Recall python data types |
| *Task 1* | Implement searching algorithm in python.  ***Note****: Don't use an existing library.* |
| *Task 2* | Implement sorting algorithm in python.  ***Note****: Don't use an existing library.* |
| *Task 3* | Implement searching and sorting algorithms on database data.  (use psycopg2 library to connect database, raw sql, python) |
|  | Move to the development phase.   * Learn about the virtual environment, package installation through pip inside the virtual environment. * Study the basic flow of the django app *(Learn how to create models, relationships in Django Models, how to write logics in views and render the context in template)* * Learn about django ORM, querysets in Django. Learn about the Django admin site and admin customization. |
| *Task 4* | Create a Django app. Simply create the models, migrate the data and directly add data to the django admin site. Learn about the usage of ***django-extensions*** and ***ipython*** and start practising the django queryset in terminal.  (mainly focus on : filter(), get(), create(), all(), values(), values\_list(), select\_related(), prefetch\_related(), order\_by(), exclude(), aggregate(), annotate(), field lookups) |

***Note:*** *Please build the habit of reading Django documentation.*

|  |  |
| --- | --- |
|  | Learn about how the folder (static and template) is managed in the project. Learn about template tags. Learn how to implement template tags in html templates. You can simply refer to Django Documentation during the learning process.  *https://docs.djangoproject.com/en/3.2/ref/templates/builtins/* |
| *Task 5* | * Create a Django CRUD app with ***function-based views***.   (App should be able to create, update, list and delete ***the*** ***book***)  Refer this link:  *https://www.geeksforgeeks.org/django-crud-create-retrieve-update-delete-function-based-views/*  ***Note:*** *Please learn the overall flow of how the app is working.* |
| *Task 6* | * Create a Django CRUD app with ***class-based views***.   (App should be able to create, update, list and delete the book) |
| *Task 7* | * Create a signup form. Check email validation. Passwords should include capital letters, special characters, and alphabets. * Create a login form and raise an error if the user is not authenticated. * Send email for password-reset. |
| *Task 8* | Create a simple blog-like application.  Draw a schema diagram with entity: ***user, userprofile, post, category***.  Features to be included:   * Search post based on title. * Add Pagination.   Provide basic functionality like:   * Post (Create, Update, Delete, List) * User Profile (Update)   ***Note:***   1. *Only the authenticated user should be able to enter the page.* 2. *Blogs posted by others should have just read-only access.*   *Beautify your app with html, css and bootstrap.* |
| Challenging task | * Export a list of users who have posted the blog in the google sheet file. * Create python package |

***Note: Please build the habit of reading Django documentation***