Technical Domain Tasks

Recruitments2K23

General Guidelines 👩‍🔧

- Applicants are required to accomplish at least one task within their chosen field of application.
- You are required to place each task in a GitHub repository that is accessible to the public.
- The projects must be well-structured and should contain a ReadME file that explains the project.
- Web Dev projects should be hosted on a platform of their choice, ex: Netlify, Vercel.

Deadline: 12th September, 2023

AI/ML 🤖

1st Years:

You have been provided with a dataset containing student performance details. Perform Exploratory Data Analysis (EDA) by creating a Jupyter notebook. Analyse univariate (individual variables) bivariate (relationships between variables), and multivariate (multiple variables) aspects.

Share your findings and inferences about trends, patterns, or insights you discovered through EDA.

Upload your completed notebook to a GitHub repository and share the link for evaluation.

Dataset - student_ml_dataset.csv

2nd Years:

Analyse a student performance dataset by performing Exploratory Data Analysis (EDA) in a Jupyter notebook. Explore trends, correlations, and insights using univariate, bivariate, and multivariate techniques. Choose a machine learning model (e.g., linear regression) to predict a target variable based on features from the dataset.

Train and evaluate the model, then upload your notebook to GitHub and share the link.

Dataset - student_extended_ml_dataset2.csv

(or)

Analyse a student-teacher interaction dataset using NLP techniques. Preprocess the text, explore patterns and sentiment, and propose an NLP-based insight to enhance interactions.

Share your findings in a Jupyter Notebook on GitHub and provide the repository link.

Dataset - nlp.csv

3rd years:

Develop a machine learning model using the "Singapore Maritime Dataset" to classify vessel types (e.g., cargo ships, fishing boats) in videos. Download the dataset from Singapore Maritime Dataset | Kaggle and create a Jupyter notebook. Preprocess data, partition it and use advanced techniques like Convolutional Neural Networks (CNNs). Evaluate model accuracy on test data, and discuss performance, challenges, and real-world applications.

Upload your notebook on GitHub and provide the repository link for evaluation.

Resources

- **EDA -** https://towardsdatascience.com/exploratory-data-analysis-eda-python-87178e35b14, https://www.youtube.com/watch?v=-o3AxdVcUtQ&ab_channel=edureka%21, https://towardsdatascience.com/exploratory-data-analysis-in-python-a-step-by-step-process-d0dfa6bf94ee
- **Model -** https://www.youtube.com/watch?v=E5RjzSK0fvY&ab_channel=edureka%21
- **Image classification -** https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d
- **CNNs -** https://www.youtube.com/watch?v=FTr3n7uBIuE&ab_channel=SirajRaval
- **NLP -** https://medium.com/geekculture/basics-of-natural-language-processing-for-beginners-d86351df9d09

WEB DEV 🌐

1st years:

Create a portfolio site of yourself. Let us know about you, your skills, projects, certifications and anything else you would like to share.

2nd Years:

Frontend:

Develop an e-commerce store using a framework of your choice. The project entails creating a product listing page to display various products with details such as title, price, description, and images.

Implement dynamic routing to showcase individual product pages. Build a cart system that enables users to add, update, and remove items from their cart, along with a dedicated cart page displaying the added products, quantities, and total prices.

Optionally design a checkout process where users can review their cart, enter shipping and payment information, and submit orders.

Apply responsive styling and deploy the e-commerce store to a hosting platform.

Recommended tech stack: ReactJS, NextJS, Tailwind

Backend:

Create a web-based Library Catalog Management System that allows librarians to manage their library's collection of books, magazines, and other materials. The system will provide a RESTful API for performing CRUD operations on library catalogue entries.

Make sure to design the API with clean and intuitive endpoints that are easy to use, follow standard RESTful practices and comprehensive documentation for all your endpoints.

All entries must be saved on any database (Firestore, MongoDB, DynamoDB etc.)

3rd years:

Event Planner App

Description:

Create a full-stack web application for event planning, enabling users to organise and share details about their events. You can use any Web Frameworks you like ;)

We are looking for some Key Features like:🔐

- Event Creation: Users can create events, specifying details such as event name, date, location, and a brief description.
- User Authentication: Implement user registration and login for users to manage their events.
- Event Editor: Provide an easy-to-use event editor, allowing users to customise event details and layout.
- Guest List: Users can manage guest lists for their events, including sending invitations and tracking RSVPs.
- Responsive Design: Ensure the application is responsive to work on various devices and screen sizes.
- User Profiles: Users can create profiles with their upcoming events and a list of attended events.
- Comments and Ratings: Allow users to leave comments and ratings on event pages for social interaction.

*Projects should be hosted on a platform of your choice
ex: Netlify, Vercel.

Resources 📑📑

Frontend

- React - https://react.dev/
- Next - https://nextjs.org/
- TailwindCSS - https://tailwindcss.com/

Backend

- Node.js - https://nodejs.org/en/docs/

- Express - https://expressjs.com/
- Django- https://www.djangoproject.com/
- Flask- https://flask.palletsprojects.com/
- FastAPI- https://fastapi.tiangolo.com/
- MySQL- https://dev.mysql.com/doc/refman/8.0/en/
- MongoDB- https://www.mongodb.com/docs/atlas/
- DynamoDB-https://www.dynamodbguide.com/what-is-dynamo-db
- Firebase - https://firebase.google.com/

APP DEV

1st years:

PORTFOLIO APP

You are required to make a BASIC Portfolio app consisting of only a single screen displaying details about you such as:

- Full Name
- Profile Picture
- Contact Details, etc.

2nd years:

WEATHER APP

Create a mobile application that allows users to view the weather conditions for their current location. Your app must have:

- Home screen displaying the weather conditions of your current location.
- Another screen to search the city name and display its weather report.
- You can use the Openweather API or any other API of your choice.

3rd years:

CHAT APP

Create a mobile application that allows users to chat and text with multiple other users using the same app (Similar to WhatsApp, Messenger, etc.).
You must use:

- Firebase Authorization in the app for Sign in and Sign Up.
- Different screens for different functionalities of the app.
- Finally linking the Firebase with the Flutter Chat App.

*Make sure your app features an intuitive UI and attractive colour scheme. Do whatever you can to make it custom to your own self.