Interview task

ABC is an AI company and they are daily working with object detection algorithms and AI engineers in that company approached you regarding the creation of one application that will help them to reduce time in their daily work. So, you have to create that application in the following manner.

Frameworks to be used:

Backend: Flask or Django
Database: Any SQL or NoSQL DB
Frontend: You can create HTML pages with Flask or Django templates or you can use a frontend framework like React JS. (Bonus points if you can create a frontend in React JS). You don't have to worry about the styling of the frontend but it should be functional and perfectly integrated with the backend.

Task-1:

Create a page in the frontend where the user will upload an image file and its corresponding XML. After uploading the image and XML file, the output image with bounding boxes drawn over it should be displayed on that same page in UI.

Here XML file will be used for extracting information of objects present in that image,it will contain the object name and it's coordinates (xmin,ymin,xmax,ymax) in the <object> tag.

E.g., refer to P00X000-2019092701422.jpg and P00X000-2019092701422.xml from the **sample_input_dataset** folder which is shared with you. After uploading the image and XML file, the user should see the output image like the one which is inside the **sample_output_image** folder.

Create a REST API that will receive the image and XML file uploaded from the frontend, parse the XML and draw the bounding boxes over an image and return back the output image to the frontend.
Also, you have to store the following information of the uploaded image in the database.

- image_name
- object name and bounding box coordinates.
- timestamp

Note that one image can contain more than one object.

You can choose any SQL or NoSQL database according to your choice but give a proper justification for the same. Please mention your justification in the Readme file.

Task-2:

After completing the above task you will have all data in the database. Now create a report extraction page where the user will select the start date and end date from the UI and should get a report in the .csv format of the images and bounding box details of that specific period.

Create a REST API that will accept start_date and end_date in the parameter and it will generate a report of the following fields and return a .csv file.

Report columns:

- image_name
- object_name
- x_min
- y_min
- x_max
- y_max
- timestamp

If there are multiple objects in one image then create multiple rows for that image in the report.

Deployment (Bonus points if you complete this step)

Dockerize your app and create docker-compose files for building docker images and running services.

Submission Steps:

Share the link of the GitHub repository and also create a Readme file mentioning steps for running this application, API documentation, and any other information if needed.