Use case to be developed in AIG

As discussed, please learn how to do the below activities

1. Build a UI application
   1. React as frontend
   2. Django Python as API
2. Build a Machine learning model that should predict the defect and defect description from the rally
   1. Use rally data from the PCGT project to train the model
   2. Select either a classification or regression model based on the feature of the data
   3. Train the model
   4. Test the model to reach an accuracy of above 80%
3. Call model from UI using API
4. When the user enters a new defect, UI should suggest if this defect already exists and predict if there need to be new defects, if yes help raising the defect with the given information

Training needs to go through before Use case starts

1. Application development

Building an application with a React frontend, Python Django backend, and PostgreSQL database involves several steps. Please see high-level overview of the process:

1. **Set up your development environment**:
   * Install Node.js and npm for React.
   * Install Python and Django.
   * Install PostgreSQL and set up your database.
2. **Create your React frontend**:
   * Use create-react-app to bootstrap your project.
   * Develop your components and views.
   * Use state management libraries like Redux if needed.
   * Style your application with CSS or a framework like Bootstrap.
3. **Develop your Django backend**:
   * Set up a Django project and app.
   * Define your models that will interact with PostgreSQL.
   * Create views to handle the business logic.
   * Set up URLs for routing.
4. **Connect React with Django**:
   * Use Django REST framework to create APIs.
   * Fetch data from the backend using React’s fetch or axios.
5. **Configure PostgreSQL database**:
   * Define your database schema.
   * Run migrations to create database tables.
   * Connect Django to PostgreSQL using the DATABASES setting in settings.py.
6. **Test your application**:
   * Write tests for both frontend and backend.
   * Use tools like Jest for React and Django’s test framework.
7. **Deploy your application**:
   * Choose a hosting service for deployment.
   * Configure static files and database for production.
   * Use tools like Docker for containerization if necessary.
8. **Maintain and update your application**:
   * Regularly update dependencies and frameworks.
   * Add new features and fix bugs.

1.[React JS + Python Django + SQLite | full-stack app tutorial (youtube.com)](https://www.youtube.com/watch?v=WsBYK5Nv2V8)

Here instead of SQL lite, use PostgreSQL

2.[Python Django + PostgreSQL | REST API Tutorial (youtube.com)](https://www.youtube.com/watch?v=Pwwz4_AvHDU)

3.[Django & React Web App Tutorial - Authentication, Databases, Deployment & More... (youtube.com)](https://www.youtube.com/watch?v=c-QsfbznSXI)

1. Regression model creation and integration with UI
2. Create Machine Learning model: Python libraries like scikit-learn, TensorFlow, or PyTorch to create your machine learning model. This involves data preprocessing, model training, and model evaluation steps.
3. Expose model through a Python API: Once model is ready, you can create a Python API using a web framework like Flask or Django. This API should have endpoints that accept input data, run this data through your model, and return the model's predictions.
4. Create React UI: create a React application that interacts with your Python API. This application should have a form or some other way for users to input or upload data, and it should display the model's predictions once they're returned by the API.
5. Connect React UI to Python API: This involves making HTTP requests from React application to Python API. Use the fetch API or a library like axios to make these requests.

[( [Build your first machine learning model in Python (youtube.com)](https://www.youtube.com/watch?v=29ZQ3TDGgRQ)](https://www.youtube.com/watch?v=c-QsfbznSXI)

[Python Machine Learning Tutorial (Data Science) (youtube.com)](https://www.youtube.com/watch?v=7eh4d6sabA0)

[How To Deploy Machine Learning Model On React JS (youtube.com)](https://www.youtube.com/watch?v=-XirZSq6Wcs)