

```

class Scenario(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.CASCADE)
    scenario_name = models.CharField(max_length=100,unique=True, null=False)
    description = models.CharField(max_length=200)

    def __str__(self):
        return f"{self.pk}.{self.user}"

class ScenarioSolution(models.Model):
    SOLUTION_TYPE_CHOISES = (
        ('supervised', 'Supervised'),
        ('unsupervised', 'Unsupervised')
    )
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
on_delete=models.CASCADE)
    scenario = models.ForeignKey(Scenario, on_delete=models.CASCADE)
    solution_name = models.CharField(max_length=200,unique=True,null=False)
    description = models.TextField(blank=True, null=True)
    solution_type = models.CharField(max_length=20,
choices=SOLUTION_TYPE_CHOISES,null=False)
    training_file = models.FileField(upload_to='files', blank=True, null=True)
    test_file = models.FileField(upload_to='files', blank=True, null=True)
    protected_features = models.CharField(max_length=200, blank=True,
null=True)
    protected_values = models.CharField(max_length=200, blank=True, null=True)
    target_column = models.CharField(max_length=200, null=True,
default='target')
    outlier_data_file = models.FileField(upload_to='files', blank=True,
null=True)
    favourable_outcome = models.CharField(max_length=200, blank=True,
default='1')
    factsheet_file = models.FileField(upload_to='files', blank=True,
null=True)
    model_file = models.FileField(upload_to='files', blank=True, null=True)
    metrics_mappings_file=models.FileField(upload_to='files', blank=True,
null=True)

```

A ScenarioSolution belongs to a Scenario and is linked to a User

A ScenarioSolution name is unique for a given Scenario for a given User

A Scenario name is unique for a given User

For supervised and unsupervised ScenarioSolution there are 4 PillarScores: Accoutnability, Fairness, Robustness, Explainability.

#### A) Pillar API Endpoints

**a. Supervised ScenarioSolution**

- i. GET Fairness, returns  
`get_fairness_score_supervised(ScenarioSolution.solution_name)`
- ii. GET Robustness, returns  
`get_robustness_score_supervised(ScenarioSolution.solution_name)`
- iii. GET Accountability, returns  
`get_accountability_score_supervised(ScenarioSolution.solution_name)`
- iv. GET Explainability, returns  
`get_explainability_score_supervised(ScenarioSolution.solution_name)`
- v.

**b. Unsupervised ScenarioSolution**

- i. GET Fairness, returns `analyse(ScenarioSolution.solution_name)` from `unsupervised/Functions/Fairness/Fairness.py`
- ii. GET Robustness, returns  
`get_robustness_score_supervised(ScenarioSolution.solution_name)`, returns  
`analyse()` from `unsupervised/Functions/Robustness/Robustness.py`
- iii. GET Accountability, returns `analyse(ScenarioSolution.solution_name)` from  
`unsupervised/Functions/Accountability/Accountability.py`
- iv. GET Explainability, returns `analyse(ScenarioSolution.solution_name)` from  
`unsupervised/Functions/Explainability/Explainability.py`

Currently implemented are each of the above api endpoints as POST but they take several parameters and not just a `ScenarioSolution.solution_name`, this change is needed.

**B) MetricScore API Endpoints**

Again here for each metric a POST API Endpoint is implemented but it takes several parameters but only should take the `ScenarioSolution.solution_name`, the change has to be done and it should now be a GET API Endpoint

**C) TrustScore API Endpoints**

Given a `ScenarioSolution.name` there should be 1 Getendpoint that returns the TrustScore given by `trusting_AI_scores_supervised`, `trusting_AI_scores_unsupervised`, there is a bug and the current implementation of both endpoints as POST and with parameters is not working

**D) General remarks API Endpoints**

Please check that all Endpoints work (also the already implemented GET, POST, PUT, DELETE for scenario, scenariosolution and get users, get user, put user, delete user for admin. Write tests

**E) Fix bugs**

Fix bugs in the code, currently often creating a Scenario isn't working, and when uploading a unsupervised Solution on the clientside on the dashboard nothing is displayed instead of the scores and Analyze, Compare won't work either

Deploy webapp and ensure that one can create a Scenario, it's saved correctly in database (unique scenariename for a given user), can create a ScenarioSolution and it's saved correctly in database (unique scenariosolution.name for given user for a given scenario). Ensure that after uploadsolution on dashboard the scores of that solution are displayed, that analyze, compare work. Add graphs to display analyze, compare results

F) Dashboard

- a) Add a section where each metric, pillar for supervised, unsupervised is shortly described (see code, else I will provide description if not there)
- b) Add a section where the logged in users scenarios are listed as `scenario.names` and a click leads to a page that display the scenario (name, description)
- c) Add a section where the logged in users solutions are listed as `solutions.names` and a click leads to a page that display the solution (name description, `scenario.name`, `scenario.description`, files e.g, `training_dataset`, `test_dataset` etc.)

G) General Frontend

- a. Remove Userpage, Userprofile
- b. Remove unnecessary code
- c. Same colors everywhere

H) General App

- a. Clean code
- b. Document code
- c. Add design, architecture (uml diagrams) description