# TechArena 2025 - Phase 1 Submission Instructions

# Overview

Welcome to TechArena 2025! This document provides comprehensive instructions for submitting your Phase 1 solution. Please read all sections carefully to ensure your submission is properly validated and evaluated.

# **&** Submission Requirements

### Required Deliverables

Your submission must include **all** of the following components:

- 1. **Main Script**: main.py (in project root directory)
- 2. Input Data Processing: Code to read and process the provided Excel input file
- 3. Output Files: Three result files in CSV or Excel format
- 4. **Dependencies**: requirements.txt file (if using external packages)
- 5. **Documentation**: README.md file explaining your approach

### **Output Files Required**

Your main.py script must generate exactly **three output files** with the following names (CSV or Excel format):

- 1. TechArena\_Phase1\_Configuration.csv (or .xlsx)
  - Optimal BESS configuration parameters
  - Required columns: C-rate, number of cycles, yearly profits [kEUR/MW], levelized ROI
     [%]
- 2. TechArena\_Phase1\_Investment.csv (or .xlsx)
  - Investment analysis and ROI calculations
  - Must include: WACC, inflation rate, discount rate, yearly profits, year-by-year analysis, levelized
     ROI
- 3. TechArena\_Phase1\_Operation.csv (or .xlsx)
  - Required columns: Timestamp, Stored energy [MWh], SoC [-], Charge [MWh], Discharge [MWh], Day-ahead buy [MWh], Day-ahead sell [MWh], FCR Capacity [MW], aFRR Capacity POS [MW], aFRR Capacity NEG [MW]

## Project Structure

Your submission should follow this **exact** directory structure:

### Critical Requirements

- main.py MUST be in the root directory (not in subdirectories)
- Input file MUST be in input/ subdirectory
- Output files will be generated in the output directory
- No absolute file paths use relative paths only

# Technical Specifications

### Python Requirements

• Python Version: 3.8 or higher

Output Format: CSV (.csv) or Excel (.xlsx) files

### Input Data Structure

The input file contains 4 sheets:

- Day-ahead prices: Historical electricity prices for 2024
- FCR prices: Frequency Containment Reserve prices
- aFRR capacity prices: Automatic Frequency Restoration Reserve prices
- Data description: Market conditions and BESS parameters

#### **Data Validation Constraints**

### **Configuration File:**

 Must include required parameters: yearly profits [kEUR/MW],levelized ROI [%] for all C-rate and number of cycles combinations

#### **Investment File:**

- Must include required parameters: WACC, inflation rate, discount rate
- Must include Levelized ROI calculation
- Below is the sample table for reference

WACC	value
Inflation Rate	value
Discount rate	value
Yearly profits (2024)	value

Year Initial Investment [kEUR/MWh] Yearly profits [kEUR/MWh]

Year	Initial Investment [kEUR/MWh]	Yearly profits [kEUR/MWh]
2023		
2024		
2025		
2026		
2027		
2028		
2029		
2030		
2031		
2032		
2033		

| Levelized ROI | Value |

### **Operation File:**

Must include required parameters: Timestamp, Stored energy [MWh], SoC [-], Charge [MWh], Discharge [MWh], Day-ahead buy [MWh], Day-ahead sell [MWh], FCR Capacity [MW], aFRR Capacity POS [MW], aFRR Capacity NEG [MW]

# % Implementation Guidelines

### Example main.py Structure

```
import os
import pandas as pd

def main():
    """
    Main execution function for TechArena Phase 1 solution
    """
    # Read input data
    input_file = os.path.join("input", "TechArena2025_ElectricityPriceData.xlsx")

# Process data and run optimization
    # ... your implementation here ...

# Generate output files
    config_df.to_csv("TechArena_Phase1_Configuration.csv", index=False)
    investment_df.to_csv("TechArena_Phase1_Investment.csv", index=False)
    operation_df.to_csv("TechArena_Phase1_Operation.csv", index=False)
```

```
print(" All output files generated successfully!")

if __name__ == "__main__":
    main()
```

#### **Best Practices**

#### **☑** DO:

- Use relative paths for all file operations
- Check if input files exist before processing
- Handle exceptions gracefully with try-catch blocks
- Validate your output before submission
- Test your script in a clean environment
- Use meaningful variable names and add comments
- Include all required dependencies in requirements.txt

### X DON'T:

- Use absolute file paths (e.g., C:\Users\YourName\...)
- Hardcode team-specific paths or credentials
- Rely on files not provided in the input
- Create infinite loops or excessive processing
- Leave debugging print statements that flood output
- Submit incomplete or placeholder files

#### Sample requirements.txt

```
pandas>=1.5.0
numpy>=1.21.0
openpyxl>=3.0.9
matplotlib>=3.5.0
scipy>=1.8.0
```

## Submission Process

### Step 1: Prepare Your Submission

- 1. Complete your implementation in main.py
- 2. Test thoroughly with the provided input data
- 3. Verify all output files are generated correctly
- 4. Create/update requirements.txt with all dependencies
- 5. Write README.md explaining your approach and any special instructions

#### Step 2: Test Your Submission

Before submitting, test your solution:

```
# Create a clean test environment
cd your_project_directory

# Install dependencies (if using virtual environment)
pip install -r requirements.txt

# Run your main script
python main.py

# Verify output files are created
ls -la *.csv # or *.xlsx
```

### Step 3: Create Submission Package

- 1. Create a ZIP file containing your entire project
- 2. Name the file: TeamName\_TechArena2025\_Phase1.zip
- 3. **Verify contents** match the required structure
- 4. Test the ZIP file by extracting and running in a new location

### Step 4: Submit

- 1. **Upload your ZIP file** to the designated submission platform
- 2. Confirm successful submission before the deadline

### Automated Validation

Your submission will undergo automated validation that checks:

### ✓ Structure Validation

- Correct file and directory structure
- Presence of required files (main.py, output files)
- Proper file naming conventions

## ✓ Code Quality

- Script executability and syntax
- Import statement validation
- Function and documentation presence
- Excel/CSV handling capabilities

## ✓ Execution Testing

- Virtual environment setup and dependency installation
- Error handling and output generation

## Output Validation

Correct number and format of output files

- Schema compliance (column names and types)
- File size and content reasonableness

### Common Issues and Solutions

Issue: "main.py not found in project root"

**Solution**: Ensure main.py is directly in the ZIP file root, not in a subdirectory.

Issue: "Input file not found"

**Solution**: Place input file in input/ subdirectory and use relative path:

```
input_file = os.path.join("input", "TechArena2025_ElectricityPriceData_v2.xlsx")
```

Issue: "Requirements installation failed"

#### Solution:

- Test your requirements.txt in a clean environment
- Use specific version numbers for critical packages
- Include openpyx1 for Excel file handling

Issue: "Output file validation failed"

#### Solution:

- Check column names match exactly (case-sensitive)
- Verify data types and ranges
- Ensure no missing values in required fields

Issue: "Script execution timeout"

#### Solution:

- Optimize your algorithms for efficiency
- Avoid unnecessary data loading or processing
- Consider using vectorized operations with pandas/numpy

# Support and Contact

**Technical Support** 

• Email: amandine@bemyapp.com

Only For Code Submission Support

• **Email**: mangesh.mankar@huawei.com

**FAQ** and Updates

- Event Website: https://techarena-nuremberg.hackathon.com/
- **Discussion Forum**: Available on the event platform
- Live Q&A Sessions: Scheduled times will be announced

# Submission Checklist

Before submitting, verify:

- main.py is in the root directory
- Input file is in input/ subdirectory
- Script runs successfully and generates all 3 output files
- Prequirements.txt includes all dependencies
- README.md explains your approach
- ZIP file is named correctly: TeamName\_TechArena2025\_Phase1.zip
- No absolute file paths or hardcoded paths
- All output files have correct column names
- Tested in a clean environment

### Good luck, and may the best solution win! &

TechArena 2025 Organizing Committee

Last Updated: 26-08-2025