






BlueEdge Quick Start Guide

Get up and running with BlueEdge mobile edge data cleaning in under 5 minutes!

What You'll Build

By the end of this guide, you'll have:

-  **BlueEdge app** running on your device
 -  **Sample dataset** loaded and processed
 -  **Real-time duplicate detection** working
 -  **Firebase integration** (optional) configured
 -  **Performance metrics** displayed
-

Prerequisites

System Requirements

- **Mobile:** Android 6.0+ or iOS 12.0+
- **RAM:** 3GB minimum (6GB recommended)
- **Storage:** 500MB free space
- **Network:** WiFi or 4G/5G connection

Development Requirements (Optional)

- **Python:** 3.8 or higher
- **Git:** For cloning the repository
- **Code Editor:** VS Code, PyCharm, or similar

Quick Installation

Method 1: Direct APK Installation (Fastest)

```
bash

# Download the latest APK
wget https://github.com/YourOrg/BlueEdge/releases/latest/download/BlueEdge.apk

# Install on Android device
adb install BlueEdge.apk

# Or install manually:
# 1. Enable "Unknown Sources" in Android Settings
# 2. Download APK to device
# 3. Tap APK file to install
```

Method 2: From Source (Recommended for Developers)

```
bash
```

```
# 1. Clone the repository
git clone https://github.com/YourOrg/BlueEdge.git
cd BlueEdge

# 2. Create virtual environment
python -m venv blueedge_env

# 3. Activate virtual environment
# Windows:
blueedge_env\Scripts\activate
# macOS/Linux:
source blueedge_env/bin/activate

# 4. Install dependencies
pip install -r requirements.txt

# 5. Run the application
python main.py
```

First Run: 5-Minute Demo

Step 1: Launch BlueEdge

```
bash

# If installed from source:
cd BlueEdge
python main.py

# If using APK:
# Just tap the BlueEdge icon on your device
```

Step 2: Load Sample Data

```
python

# The app comes with built-in sample data
# Click "Load Sample Dataset" button
# Or manually load data:

sample_data = [
    {
        "name": "Mohammed Ahmed Hassan",
        "email": "mohammed.ahmed@example.com",
        "id": "12345"
    },
    {
        "name": "Mohammad Ahmad Hasan", # Similar spelling
        "email": "mohammed.ahmed@example.com",
        "id": "12346"
    },
    {
        "name": "Dr. Ahmed Hassan Omar",
        "email": "ahmed.hassan@example.com",
        "id": "12347"
    },
    {
        "name": "Ahmed Hassan Omar", # Without honorific
        "email": "ahmed.hassan@example.com",
        "id": "12348"
    }
]
```

Step 3: Run Duplicate Detection

1. Click **"Process Data"** button
2. **Watch real-time processing** (should take ~1 second)
3. **View results** in the results panel

Expected Output:

✅ Processing Complete!

📊 Total Records: 4

🔍 Duplicates Found: 2 pairs

⚡ Processing Time: 0.8 seconds

💾 Memory Used: 4.2 KB

🎯 Accuracy: 95.2%

Duplicate Pairs:

1. "Mohammed Ahmed Hassan" ↔ "Mohammad Ahmad Hasan" (78.4% similarity)
2. "Dr. Ahmed Hassan Omar" ↔ "Ahmed Hassan Omar" (95.2% similarity)

Step 4: 📈 View Performance Metrics

Performance Dashboard:

Metric	Value
Records/Second	5,000
Memory Usage	5KB
CPU Usage	18%
Battery Impact	<1%
Network Usage	Minimal

Configuration

Basic Configuration

```
python

# config/settings.py
BLUEEDGE_CONFIG = {
    # Algorithm Settings
    "similarity_threshold": 0.25,
    "enable_phonetic_matching": True,
    "case_sensitive": False,

    # Performance Settings
    "batch_size": 1000,
    "max_records_per_session": 10000,
    "enable_caching": True,

    # Error Types to Detect
    "error_types": [
        "spelling_variations",
        "misspellings",
        "name_abbreviations",
        "honorific_prefixes",
        "common_nicknames",
        "split_names"
    ]
}
```

Firestore Integration (Optional)

```
javascript
```

```
// config/firebase.json
{
  "apiKey": "your-api-key-here",
  "authDomain": "blueedge-demo.firebaseio.com",
  "databaseURL": "https://blueedge-demo.firebaseio.com",
  "projectId": "blueedge-demo",
  "storageBucket": "blueedge-demo.appspot.com",
  "messagingSenderId": "123456789"
}
```

Real-World Example

University Student Records


python

```


# Load university dataset (included in the app)
from blueedge import DataProcessor

processor = DataProcessor()






# Load the university dataset
university_data = processor.load_dataset("data/university_records.csv")


print(f" Loaded {len(university_data)} student records")

# Process for duplicates
results = processor.detect_duplicates(university_data)

# Results
print(f"""
 BlueEdge Processing Results:

=====

 Total Records: {results.total_records;}
 Duplicates Found: {results.duplicates_found;}
 Processing Time: {results.processing_time:.2f}s
 Memory Used: {results.memory_used}KB
 Overall Accuracy: {results.accuracy:.1f}%

 Performance by Error Type:

=====

- Different Spellings: {results.spelling_accuracy:.1f}%
- Misspellings: {results.misspelling_accuracy:.1f}%
- Name Abbreviations: {results.abbreviation_accuracy:.1f}%
- Honorific Prefixes: {results.honorific_accuracy:.1f}%
- Common Nicknames: {results.nickname_accuracy:.1f}%
- Split Names: {results.split_name_accuracy:.1f}%
""")


```


Expected Output:

BlueEdge Processing Results:

 Total Records: 2,971

 Duplicates Found: 146

 Processing Time: 2.97s

 Memory Used: 15KB

 Overall Accuracy: 82.2%

Performance by Error Type:

- Different Spellings: 78.4%
- Misspellings: 72.0%
- Name Abbreviations: 90.5%
- Honorific Prefixes: 95.2%
- Common Nicknames: 76.2%
- Split Names: 85.7%

Advanced Features

Custom Algorithm Tuning

```
python
```

```
# Advanced configuration
from blueedge import AdvancedConfig

config = AdvancedConfig(
    # Levenshtein distance threshold
    similarity_threshold=0.20,

    # Enable fuzzy matching
    fuzzy_matching=True,

    # Phonetic algorithms
    enable_soundex=True,
    enable_metaphone=True,

    # Performance optimization
    parallel_processing=True,
    cache_results=True,

    # Specific to your domain
    custom_dictionaries={
        "honorifics": ["Dr", "Prof", "Mr", "Mrs", "Ms"],
        "common_nicknames": {
            "Mohammed": ["Hamada", "Hammouda"],
            "Abdullah": ["Abdu", "Abdo"],
            "Ibrahim": ["Bebo"],
            # Add your custom mappings
        }
    }
)

processor = DataProcessor(config=config)
```

Real-Time Processing

```
python

# Set up real-time monitoring
from blueedge import RealTimeMonitor

monitor = RealTimeMonitor()

@monitor.on_new_record
def handle_new_record(record):
    """Process each new record as it arrives"""
    result = processor.check_duplicate(record)

    if result.is_duplicate:
        print(f"🚨 Duplicate detected: {record.name}")
        print(f"  Matches: {result.matching_record.name}")
        print(f"  Similarity: {result.similarity:.1f}%")
    else:
        print(f"✅ New record added: {record.name}")

# Start monitoring
monitor.start()
```

Mobile-Specific Features

Offline Mode

```
python
```

```
# Enable offline processing
processor.enable_offline_mode()

# Process data without internet
results = processor.process_offline(data)

# Sync when connection restored
processor.sync_when_online()
```

Battery Optimization

```
python

# Configure for battery efficiency
processor.configure_battery_mode(
    low_power_mode=True,
    reduce_cpu_usage=True,
    batch_processing=True
)
```

Background Processing

```
python
```

```
# Android background service
from blueedge.mobile import BackgroundService

service = BackgroundService()
service.start_background_processing()

# iOS background task
from blueedge.mobile import BackgroundTask

task = BackgroundTask()
task.process_in_background(data, callback=handle_results)
```

Troubleshooting

Common Issues

Issue	Cause	Solution
App crashes on startup	Insufficient RAM	Close other apps, restart device
Slow processing	Large dataset	Enable batch processing
No internet sync	Firebase config missing	Check firebase.json configuration
Permission denied	Android permissions	Grant storage/network permissions
High battery usage	Continuous processing	Enable power saving mode

Performance Optimization

```
python
```

```
# Optimize for your device
if device.ram < 4: # GB
    config.batch_size = 500
    config.enable_caching = False
elif device.ram < 6:
    config.batch_size = 1000
    config.enable_caching = True
else:
    config.batch_size = 2000
    config.enable_caching = True
    config.parallel_processing = True
```

Performance Benchmarking

Quick Benchmark Test

```
python
```

```
# Run built-in benchmark
```

```
from blueedge import Benchmark
```

```
benchmark = Benchmark()
```

```
results = benchmark.run_quick_test()
```

```
print(f"""
```

```
⚡ BlueEdge Benchmark Results:
```

```
=====
```

```
Device: {benchmark.device_info}
```

```
RAM: {benchmark.ram_gb}GB
```

```
CPU: {benchmark.cpu_cores} cores
```

```
📊 Performance Metrics:
```

```
- Records/Second: {results.records_per_second:,}
```

```
- Memory Efficiency: {results.memory_efficiency:.1f}%
```

```
- CPU Utilization: {results.cpu_usage:.1f}%
```

```
- Battery Impact: {results.battery_impact:.1f}%
```

```
🏆 Performance Rating: {results.rating}/10
```

```
""")
```

Next Steps

1. Explore Advanced Features

- **Custom algorithms:** Tune for your specific data
- **Enterprise integration:** Connect to your databases
- **Batch processing:** Handle large datasets efficiently

2. Integration Options

- **REST API:** Integrate with existing systems
- **Database connectors:** Direct database integration
- **Cloud services:** Enhanced cloud synchronization

3. Customization




- **Custom UI:** Modify the interface for your needs
- **Domain-specific rules:** Add industry-specific patterns
- **Multi-language support:** Process non-English names

4. Production Deployment




- **Security hardening:** Implement additional security measures
 - **Performance monitoring:** Set up continuous monitoring
 - **Backup strategies:** Implement data backup procedures
-

Additional Resources

Documentation

-  [Full Documentation](#)
-  [Technical Specifications](#)
-  [Architecture Guide](#)
-  [Security Guide](#)

Community

-  [Discord Community](#)
-  [Twitter Updates](#)
-  [YouTube Tutorials](#)

-  [Blog Articles](#)






Support

-  [Report Issues](#)
 -  [Email Support](#)
 -  [Enterprise Support](#)
-


Congratulations!

You've successfully set up BlueEdge and processed your first dataset!

What you've accomplished:

-  Installed and configured BlueEdge
-  Processed real data with 82.2% accuracy
-  Achieved 4-30x performance improvement over commercial tools
-  Reduced memory usage by 2-12x
-  Enabled privacy-preserving local processing

Ready for production? Check out our [Enterprise Guide](#) for deployment best practices.

 *Happy data cleaning with BlueEdge!*