

ProjPack Project Structure

ProjPack Project Structure

```
projpack/  
├─ projpack/  
│   ├─ __init__.py  
│   ├─ core.py           # Main aggregation logic  
│   ├─ ignore_parser.py  # .ppignore file parsing  
│   ├─ file_handler.py   # File reading and filtering  
│   └─ cli.py            # Command-line interface  
├─ tests/  
│   ├─ __init__.py  
│   ├─ test_core.py  
│   ├─ test_ignore_parser.py  
│   ├─ test_file_handler.py  
│   └─ fixtures/        # Test project structures  
│       ├─ sample_project/  
│       └─ .ppignore_samples/  
├─ examples/  
│   ├─ sample_output.txt # Example of ProjPack output  
│   └─ .ppignore.example # Example .ppignore file  
├─ setup.py  
├─ pyproject.toml  
├─ requirements.txt  
├─ README.md  
├─ LICENSE  
├─ .gitignore  
└─ .ppignore            # ProjPack's own ignore file
```

Module Responsibilities

core.py

- Main `ProjectPacker` class
- Directory traversal logic
- File aggregation orchestration
- Output generation

ignore_parser.py

- Parse .ppignore files
- Pattern matching (glob-style patterns)
- Directory/file exclusion logic

file_handler.py

- File reading utilities
- Text/binary file detection
- File size validation
- Encoding handling

cli.py

- Command-line argument parsing
- User interface
- Error handling and user feedback
- Integration with core functionality

Key Features to Implement

1. Recursive Directory Traversal

- Walk through all subdirectories
- Respect ignore patterns
- Handle symbolic links appropriately

2. Smart File Filtering

- Binary file detection and exclusion
- File size limits
- Common ignore patterns (if no .ppignore exists)

3. Structured Output Format

- Clear file demarcation
- Relative path headers
- Consistent formatting
- Metadata inclusion (file count, total size)

4. CLI Interface

- Source directory specification
- Output file designation
- Verbose/quiet modes
- Custom ignore file specification