



FIGURE 5: PROTOTYPE SNAPSHOT 2

This snapshot shows the project directory that is generated after the coding model executes the task tree. The directory includes all scaffolded files and folders, configurations, source code, and testing modules, providing a complete structure for the project.

```
# Generated Python Project

## Project Details
- **Language**: python
- **Framework**: fastapi
- **Generated from**: Create a REST API for Fibonacci sequence with FastAPI

## Project Structure
...
{
  "./app\main.py": {
    "name": "setup_main_app",
    "description": "Create main FastAPI application instance with configuration",
    "subtasks": [],
    "function_name": "app",
    "parameters": {},
    "return_type": "FastAPI",
    "file_path": "./app\main.py",
    "implementation_details": {
      "TYPE": "application",
      "expected_loc": 20,
      "to_be_coded": true,
      "logic": "1. Import FastAPI and CORS middleware\n2. Create FastAPI instance\n3. Add CORS middleware to allow all origins\n4. Include fibonacci router\n5. Add health check endpoint at /health",
      "dependencies": [
        "fastapi",
        "fastapi.middleware.cors"
      ],
      "framework_specifics": "Use FastAPI() instance creation and router inclusion",
      "example_usage": "app = FastAPI()\napp.include_router(fibonacci.router)"
    },
    "language": "python",
    "framework": "fastapi"
  },
  "./app\schemas.py": {
    "name": "fibonacci_schemas",
    "description": "Define Pydantic models for request and response",
    "subtasks": [],
    "function_name": "FibonacciRequest, FibonacciResponse",
    "parameters": {},
    "return_type": "BaseModel",
    "file_path": "./app\schemas.py",
  }
}
```

FIGURE 6: PROTOTYPE SNAPSHOT 3