

🏠 Legacy Keeper AI - Complete Project Structure

📁 Project Folder Structure

...

```
legacy-keeper-ai/
|
├── main.py                # Flask application (main entry point)
├── requirements.txt       # Python dependencies
├── .gitignore             # Git ignore file
├── README.md              # Project documentation
|
├── templates/             # HTML templates folder
|   ├── index.html         # Homepage
|   ├── login.html         # Login page
|   ├── register.html      # Registration page
|   ├── dashboard.html     # User dashboard
|   ├── chat.html          # Harmony AI chat interface
|   ├── family.html        # Family hub page
|   └── memorials.html     # Memorial pages
|
└── static/                # Static files folder (optional)
    ├── css/               # CSS files (if separated)
    ├── js/                # JavaScript files (if separated)
    └── images/             # Images folder
```

...

📥 Download Instructions

Since I cannot create actual downloadable files directly, here are **3 easy methods** to get all your files:

Method 1: Manual Download (Easiest)

Step 1: Create Project Folder

On your computer, create a folder called `legacy-keeper-ai`

Step 2: Create Files

Root Files:

File: main.py

- Copy from artifact: "main.py - Flask App with Harmony AI"
- Save as: `main.py`

File: requirements.txt

```
```txt
Flask==3.0.0
gunicorn==21.2.0
Flask-SQLAlchemy==3.1.1
psycpg2-binary==2.9.9
openai==1.3.0
Werkzeug==3.0.1
python-dotenv==1.0.0
```
```

- Save as: `requirements.txt`

File: .gitignore

```
```
.env
.env.local
__pycache__/
*.py[cod]
*.db
*.sqlite
venv/
ENV/
.DS_Store
logs/
```
```

- Save as: `.gitignore`

File: README.md

```
```markdown
```

# 🏠 Legacy Keeper AI

Preserve your family's legacy with Harmony - your AI companion for keeping memories alive across generations.

## ## Features

- 💬 Chat with Harmony AI
- 💖 Memory preservation
- 🕊️ Memorial pages
- 👨‍👩‍👧 Family hub

## ## Quick Start

### 1. Install dependencies:

```
```bash
pip install -r requirements.txt
```
```

### 2. Set environment variables:

```
```bash
export DATABASE_URL="your-postgres-url"
export OPENAI_API_KEY="your-openai-key"
export FLASK_SECRET_KEY="your-secret-key"
```
```

### 3. Run the app:

```
```bash
python main.py
```
```

## ## Deployment

Configured for Render.com deployment with:

- PostgreSQL (Neon.tech)
- OpenAI API
- Gunicorn WSGI server

## ## Tech Stack

- **Backend:** Flask, SQLAlchemy
- **Database:** PostgreSQL
- **AI:** OpenAI GPT-3.5/4
- **Frontend:** HTML, CSS, JavaScript
- **Hosting:** Render.com

## ## Environment Variables

Required variables:

- `DATABASE\_URL` - PostgreSQL connection string
- `OPENAI\_API\_KEY` - OpenAI API key
- `OPENAI\_BASE\_URL` - OpenAI API endpoint
- `FLASK\_SECRET\_KEY` - Flask secret key
- `FLASK\_ENV` - Environment (production/development)
- `PORT` - Port number (default: 5000)

## License

Built with ❤️ for families who value their heritage.

---

- Save as: `README.md`

#### **\*\*Templates Folder:\*\***

Create a folder called `templates/` inside your project folder, then save these files inside it:

**\*\*File: templates/index.html\*\***

- Copy from artifact: "Enhanced index.html - Beautiful Homepage with Menu"

**\*\*File: templates/login.html\*\***

- Copy from artifact: "templates/login.html & register.html" (first part - login section)

**\*\*File: templates/register.html\*\***

- Copy from artifact: "templates/login.html & register.html" (second part - register section)

**\*\*File: templates/dashboard.html\*\***

- Copy from artifact: "templates/dashboard.html - User Dashboard"

**\*\*File: templates/chat.html\*\***

- Copy from artifact: "templates/chat.html - Harmony AI Chat Interface"

**\*\*File: templates/family.html\*\***

- Copy from artifact: "templates/family.html & memorials.html" (first part - family section)

**\*\*File: templates/memorials.html\*\***

- Copy from artifact: "templates/family.html & memorials.html" (second part - memorials section)

---

## ## \*\*Method 2: Create on GitHub Directly\*\*

1. Go to: <https://github.com/adamsjeremy/AiHarmony>
2. Click "Add file" → "Create new file"
3. For templates, type: `templates/index.html` (GitHub creates folder automatically)
4. Paste code and commit
5. Repeat for all files

Then download the entire repo:

- Click "Code" button → "Download ZIP"

---

## ## \*\*Method 3: Use Git Commands\*\* (If you have Git installed)

```
``bash
Clone your repo
git clone https://github.com/adamsjeremy/AiHarmony.git
cd AiHarmony

Create templates folder
mkdir templates

Create main.py
cat > main.py << 'EOF'
[paste main.py code here]
EOF

Create requirements.txt
cat > requirements.txt << 'EOF'
Flask==3.0.0
gunicorn==21.2.0
Flask-SQLAlchemy==3.1.1
psycpg2-binary==2.9.9
openai==1.3.0
Werkzeug==3.0.1
python-dotenv==1.0.0
EOF

Create .gitignore
```

```
cat > .gitignore << 'EOF'
```

```
.env
```

```
__pycache__/
```

```
*.pyc
```

```
venv/
```

```
EOF
```

```
Create templates (repeat for each template file)
```

```
Copy code from artifacts above
```

```
Commit and push
```

```
git add .
```

```
git commit -m "Complete Legacy Keeper AI application"
```

```
git push origin main
```

```
...
```

```

```

```
📦 **File-by-File Checklist**
```

Mark when completed:

**\*\*Root Directory:\*\***

- ☐ main.py (Flask app)
- ☐ requirements.txt (Dependencies)
- ☐ .gitignore (Git ignore)
- ☐ README.md (Documentation)

**\*\*templates/ Directory:\*\***

- ☐ index.html (Homepage)
- ☐ login.html (Login)
- ☐ register.html (Register)
- ☐ dashboard.html (Dashboard)
- ☐ chat.html (Harmony chat)
- ☐ family.html (Family hub)
- ☐ memorials.html (Memorials)

```

```

```
🚀 **After Creating All Files:**
```

### ### **\*\*Option A: Upload to GitHub\*\***

1. Create repo on GitHub (or use existing AiHarmony repo)
2. Upload all files maintaining folder structure
3. Render will auto-deploy

### ### **\*\*Option B: Test Locally First\*\***

```
``bash
Install dependencies
pip install -r requirements.txt

Set environment variables
export DATABASE_URL="your-db-url"
export OPENAI_API_KEY="your-api-key"
export FLASK_SECRET_KEY="random-secret"

Run app
python main.py
``
```

Visit: <http://localhost:5000>

---

### ## 📁 **\*\*Alternative: I'll Create Individual Download Links\*\***

Since I can't create a ZIP directly, here's what I recommend:

**\*\*Copy each artifact code block above, then:\*\***

1. Use an online tool like **\*\*CodePen\*\*** or **\*\*JSFiddle\*\*** to save
2. Use **\*\*Pastebin\*\*** for code sharing
3. Copy directly to GitHub
4. Use a text editor and save locally

---

### ## 🎯 **\*\*Fastest Method Right Now:\*\***

**\*\*For your phone:\*\***

1. Open each artifact I created above
2. Long-press the code → Select All → Copy
3. Go to GitHub on your phone

4. Create new file with correct name
5. Paste code
6. Commit

**\*\*This takes about 15 minutes for all files!\*\***

---

**## ⚡ \*\*Need Help Organizing?\*\***

Tell me which method you prefer:

- **\*\*A)\*\*** Manual copy/paste from artifacts
- **\*\*B)\*\*** Upload directly to GitHub
- **\*\*C)\*\*** Use Git commands
- **\*\*D)\*\*** Something else

I'll guide you step-by-step! 🎯