Tally Code Brewers 2024
References:

- https://www.researchgate.net/publication/253121932_XGet_a_highly_scalable_and_efficient_file_transfer_tool_for_clusters
  In this research paper, we got an overall understanding of the different network file transfer techniques and also understand which approach would be helpful for us to implement.
- **Python Socket Programming Tutorial**
- **How to create a simple file transfer program using python**
  We understood how basic socket can be made for simple file transfers and connect multiple clients.
- **✔️ Windows 11 - How to Share Files, Folders & Drives Between Computers Over …**
- **WHATSAPP System Design: Chat Messaging Systems for Interviews**
- https://pypi.org/project/multicast_expert/
  We reviewed for using library packages to incorporate multicasting for file sharing
- https://www.linkedin.com/pulse/how-connect-two-computers-wirelessly-step-inderjeet-singh/
- https://github.com/shah-anurag/Peer-to-Peer-Chat-Application
- https://github.com/zerotier/toss

Gpt References for basic idea:

Example: Basic Server with File Upload (Python/Flask)

python Copy code

```python
from flask import Flask, request, jsonify
import os

app = Flask(__name__)
UPLOAD_FOLDER = '/path/to/upload'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

@app.route('/upload', methods=['POST'])
def upload_file():
    if 'file' not in request.files:
        return jsonify({"error": "No file part"}), 400
    file = request.files['file']
    if file.filename == '':
        return jsonify({"error": "No selected file"}), 400
    if file:
        file.save(os.path.join(app.config['UPLOAD_FOLDER'], file.filename))
        return jsonify({"message": "File uploaded successfully"}), 200

if __name__ == '__main__':
    app.run(debug=True)
```