```python
# From endpoints.py
from flask import Flask, Blueprint, request, jsonify;
from app.db import get_db
from app.helper import *
import json

# From api/__init__.py
from .endpoints import *

# From helper/__init__.py
from .helper_functions import *

# from helper_functions.py
import os
import platform
import pprint
import json

# from app/__init__.py
# generic code used from flask documentation for application factory functionality
# source link: https://flask.palletsprojects.com/en/1.1.x/tutorial/factory/
import os

from flask import Flask
from . import db
import app.api as api
from flask_cors import CORS

def create_app(test_config=None):
    # create and configure the app
    app = Flask(__name__, instance_relative_config=True)
    app.config.from_mapping(
        SECRET_KEY='dev',
        DATABASE=os.path.join(app.instance_path, 'game_db.sqlite'),
    )

    if test_config is None:
        # load the instance config, if it exists, when not testing
        app.config.from_pyfile('config.py', silent=True)
    else:
        # load the test config if passed in
        app.config.from_mapping(test_config)

    # ensure the instance folder exists
    try:
        os.makedirs(app.instance_path)
    except OSError:
        pass
```

```
49
50      db.init_app(app)    # registers database with app
51
52      app.register_blueprint(api.endpoints.bp)
53      CORS(app)
54      return app
55
56  # From db.py
57  # generic code used from flask documentation to set up a sqlite database
      instance
58  # source link: https://flask.palletsprojects.com/en/1.1.x/tutorial/database/
59  import sqlite3
60  import click
61  from flask import current_app, g
62  from flask.cli import with_appcontext
63  import app.data as data
64
65  def get_db():
66      if 'db' not in g:
67          g.db = sqlite3.connect(
68              current_app.config['DATABASE'],
69              detect_types=sqlite3.PARSE_DECLTYPES
70          )
71          g.db.row_factory = sqlite3.Row
72
73      return g.db
74
75
76  def close_db(e=None):
77      db = g.pop('db', None)
78
79      if db is not None:
80          db.close()
81
82  def init_db():
83      db = get_db()
84
85      with current_app.open_resource('schema.sql') as f:
86          db.executescript(f.read().decode('utf8'))
87
88  @click.command('init-db')
89  @with_appcontext
90  def init_db_command():
91      """Clear the existing data and create new tables."""
92      init_db()
93      click.echo('Initialized the database.')
94
95  def init_app(app):
96      app.teardown_appcontext(close_db)
```

```
    app.cli.add_command(init_db_command)

# from data/__init__.py
from .database_functions import *

# from database_functions.py
import csv

# requirements.txt
click==7.1.2
Flask==1.1.2
Flask-Cors==3.0.10
itsdangerous==1.1.0
Jinja2==2.11.3
MarkupSafe==1.1.1
six==1.15.0
Werkzeug==1.0.1
```