To design an online quiz system, you'll need multiple components and files to manage user data, questions, answers, quiz sessions, results, and more. Here's an overview of typical files and their functions in such a system:

1. Frontend Files

HTML Files: Define the structure of the quiz system pages (e.g., index.html, quiz.html, results.html, etc.).

CSS Files: Style the layout, colors, and fonts for the quiz system's interface (styles.css).

JavaScript Files: Handle interactivity (e.g., managing timer, fetching questions, validating answers, etc.) (quiz.js, timer.js).

2. Backend Files

Server File (main server script): Responsible for handling requests, managing sessions, storing results, and serving pages. This file could be in Python (app.py with Flask), Node.js (server.js), or PHP (index.php).

Database Configuration: Configures database connection settings (config.py, .env for credentials).

API Endpoints: Specific routes for accessing quiz data, submitting answers, and retrieving results (routes.py).

3. Database Files

Database Schema: SQL or ORM files to define tables for users, quizzes, questions, answers, and scores (schema.sql, models.py).

Seed Data: Sample questions and users to populate the database for testing (seed.sql).

4. Other Essential Files

User Authentication Files: Scripts to manage user login, signup, and access control (auth.py or auth.js).

Quiz Management Files: Admin interface for adding, editing, and deleting quiz questions (admin_panel.py, quiz_management.js).

Result Processing: Functions to calculate scores and generate reports or feedback (result_processor.py).

Configuration File: Sets global variables and configurations (config.py, settings.json).

Readme/Documentation: Explains how to set up and use the system (README.md).

Example Project Structure

Quiz-system/

```
├── frontend/
│   ├── index.html
│   ├── quiz.html
│   ├── results.html
│   ├── styles.css
│   └── quiz.js
```

```
├── backend/
│   ├── app.py              # Main server file
│   ├── config.py           # Database and server configurations
│   ├── auth.py             # User authentication
│   ├── routes.py           # API endpoints
│   ├── result_processor.py    # Processes quiz results
│   └── quiz_management.py     # Manages questions and quizzes
├── database/
│   ├── schema.sql          # Database schema
│   ├── seed.sql            # Seed data for testing
│   └── models.py           # ORM models if using a framework
└── README.md               # Project documentation
```

This structure can vary based on the programming language and framework you choose but should provide a solid foundation for developing an online quiz system. Let me know if you want more details on any specific file or functionality!