1.Introduction:

This web application combines various features, including an e-commerce website, project registration form, personal portfolio, image polling, and CAD project selection. The server, built with Express.js, handles form submissions from the project registration page.

2. HTML Files

2.1 index.html

This file represents the landing page where users can register for a project. It includes a form that collects information such as name, number, email, and more.

2.2 e-commerce.html

The e-commerce page displays products with images, descriptions, and prices. Users can add items to their cart, simulating an e-commerce experience.

Key Features:

Product listings with add-to-cart functionality.
Dynamic shopping cart updates.

e-commerce.html:

```
function addToCart(product, price) {
    cartItems.push({ product, price });
    total += price;
    updateCart();
```

2.3 personal_portfolio.html

This page serves as a personal portfolio, showcasing the user's educational background, work experience, certifications, and skills.

Key Features:

Sections for About our sir, Experience, Education, Certifications, and Skills.

Responsive design with CSS styling.

personal_portfolio.html :

```
<img src="images\\sir.jpg" alt="Fawaz Abdulkader" width="300px",
height="300px">
```

2.4 polling.html

The polling page implements an image polling system. Users can vote for their preferred image (in this case, either a dog or a cat) and provide demographic information.

Key Features:

Voting for images (Dog or Cat) with demographic information collection.

polling.html :

```
<form id="demographicForm" style="display: none;">
    <!-- ... Demographic form fields ... -->
    <button onclick="submitDemographicForm()">Submit</button>
```

2.5 project_selection.html

This page acts as a hub, providing links to various CAD projects, including personal portfolio, e-commerce, polling, and more.

Key Features:

Client-side form validation.
Submission of form data to the server.

project_selection.html :

```
function resetForm() {
    document.getElementById("projectForm").reset();
}
```

3. Node Server.js (server.js)

3.1 Setup and Dependencies
The server script is set up using Express for route handling and bodyParser for parsing request bodies.

3.2 Static File Serving
Express serves static files (HTML, CSS, images) from the root directory, allowing clients to access these resources.

3.3 Default Route
The default route (/) sends the index.html file to users when they access the root of the server.

3.4 Form Data Handling
Form data submitted from index.html is captured and saved as a JSON file (formData.json) in the /data directory.

3.5 Server Start
The server starts listening on port 3000, making the project accessible at http://localhost:3000.

3.6 Things to remember:

- Ensure Node.js is installed.
- Open a terminal and navigate to the project directory (cad-projects).
- Run npm install to install dependencies.
- Execute node server.js to start the server.
- Open a web browser and visit http://localhost:3000.
- This documentation provides a comprehensive overview of the project structure, HTML files, and the Node.js server. Each component is explained in detail to facilitate understanding and further development.

Key Features:

Express server setup with routes.
Handling form submissions and writing data to a JSON file.

Server.js

```
app.listen(PORT, () => {
    console.log(`Server is running at http://localhost:${PORT}`);
```

```
});
```