Function List and Documentation

1. `validateForm(formData)`

- **Purpose**: Validates that all required user form fields are filled out and correctly formatted.
- **Arguments**: `formData` (object) – contains `name`, `email`, and `issue`.
- **Returns**: `Boolean` – `true` if valid, `false` otherwise.
- **Code**:

javascript
CopyEdit

```
function validateForm(formData) {
  if (!formData.name || !formData.email || !formData.issue) {
    alert("All fields are required.");
    return false;
  }

  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (!emailRegex.test(formData.email)) {
    alert("Please enter a valid email address.");
    return false;
  }

  return true;
}
```

---

2. `generateTicket(formData)`

- **Purpose**: Generates a support ticket object with a unique ID and timestamp.
- **Arguments**: `formData` (object)
- **Returns**: `ticket` (object)

- **Code**:

javascript
CopyEdit

```
function generateTicket(formData) {
  const ticketId = "TICKET-" + Date.now();

  return {
    id: ticketId,
    name: formData.name,
    email: formData.email,
    issue: formData.issue,
    submittedAt: new Date().toLocaleString()
  };
}
```

---

3. `submitTicket(ticket)`

- **Purpose**: Simulates submitting the ticket to the server.
- **Arguments**: `ticket` (object)
- **Returns**: `Boolean` – simulated success.
- **Code**:

javascript
CopyEdit

```
function submitTicket(ticket) {
  // Here, the function calls the server to store the ticket
  console.log("Submitting ticket to server:", ticket);
  return true;
}
```

---

4. `displayTicketSummary(ticket)`

- **Purpose**: Displays a printable ticket summary in the browser.
- **Arguments**: `ticket` (object)
- **Returns**: `void`
- **Code**:

javascript
CopyEdit

```
function displayTicketSummary(ticket) {
  const summary = `
    <h3>Support Ticket Summary</h3>
    <p><strong>ID:</strong> ${ticket.id}</p>
    <p><strong>Name:</strong> ${ticket.name}</p>
    <p><strong>Email:</strong> ${ticket.email}</p>
    <p><strong>Issue:</strong> ${ticket.issue}</p>
    <p><strong>Submitted At:</strong> ${ticket.submittedAt}</p>
  `;
  document.getElementById("ticketSummary").innerHTML = summary;
}
```

---

5. `searchTickets(query)`

- **Purpose**: Searches existing tickets by name or ticket ID.
- **Arguments**: `query` (string)
- **Returns**: `Array` of matching ticket objects.
- **Code**:

javascript
CopyEdit

```
function searchTickets(query) {
  // Simulated search call to server
  const dummyTickets = [
    { id: "TICKET-123", name: "Alice", issue: "Printer not working" },
```

```
    { id: "TICKET-124", name: "Bob", issue: "Email issues" }
  ];

  return dummyTickets.filter(ticket =>
    ticket.name.toLowerCase().includes(query.toLowerCase()) ||
    ticket.id.toLowerCase() === query.toLowerCase()
  );
}
```

---

6. `displaySearchResults(results)`

- **Purpose**: Renders a list of search results.
- **Arguments**: `results` (array)
- **Returns**: `void`
- **Code**:

javascript
CopyEdit

```
function displaySearchResults(results) {
  if (results.length === 0) {
    document.getElementById("searchResults").innerHTML = "<p>No
tickets found.</p>";
    return;
  }

  let output = "<h3>Search Results:</h3><ul>";
  results.forEach(ticket => {
    output += `<li><strong>${ticket.id}</strong>: ${ticket.name} -
${ticket.issue}</li>`;
  });
  output += "</ul>";

  document.getElementById("searchResults").innerHTML = output;
```