AI ASSISTED CODING LAB TEST-3

ENROLLMENT NO:2503A51L14

BATCH NO: 19(E4)

NAME: ROHITH GOPAGANI

Q1:

Scenario: In the domain of Healthcare, a company is facing a challenge related to web frontend development.

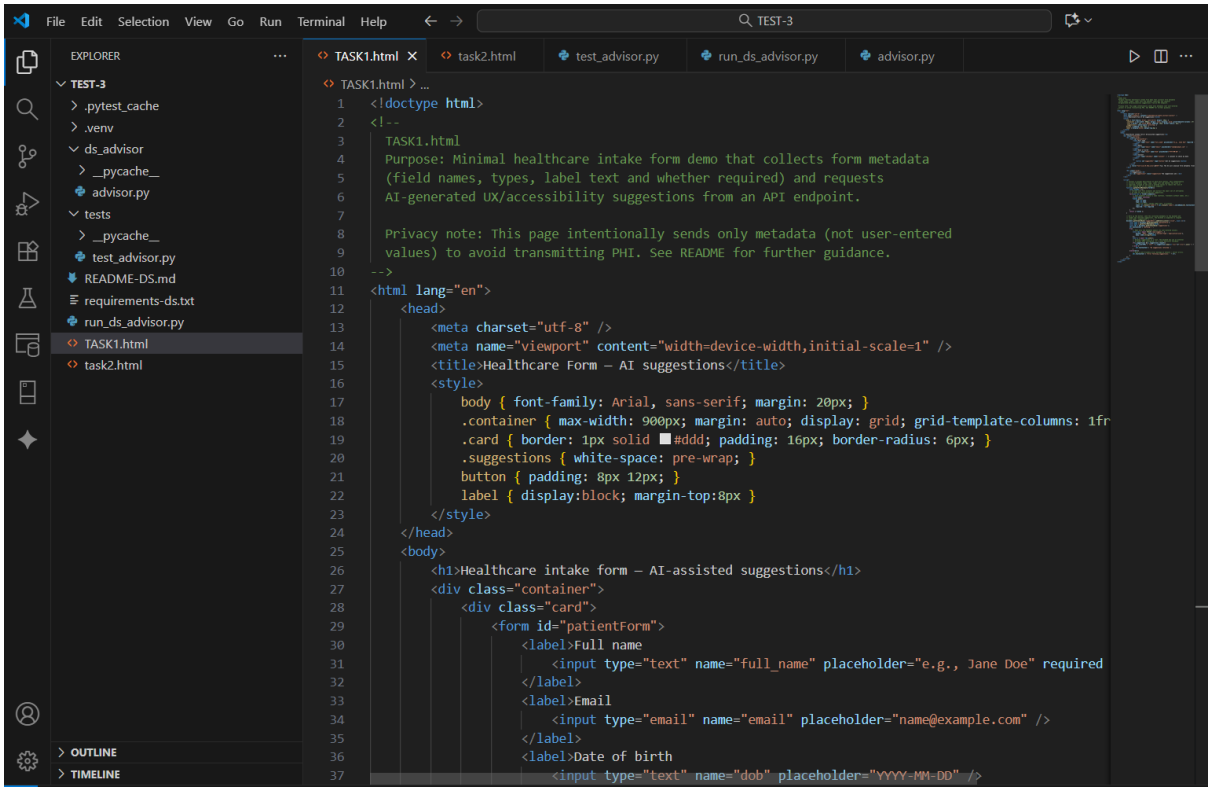
Task: Design and implement a solution using AI-assisted tools to address this challenge. Include code, explanation of AI integration, and test results.

Deliverables: Source code, explanation, and output screenshots.

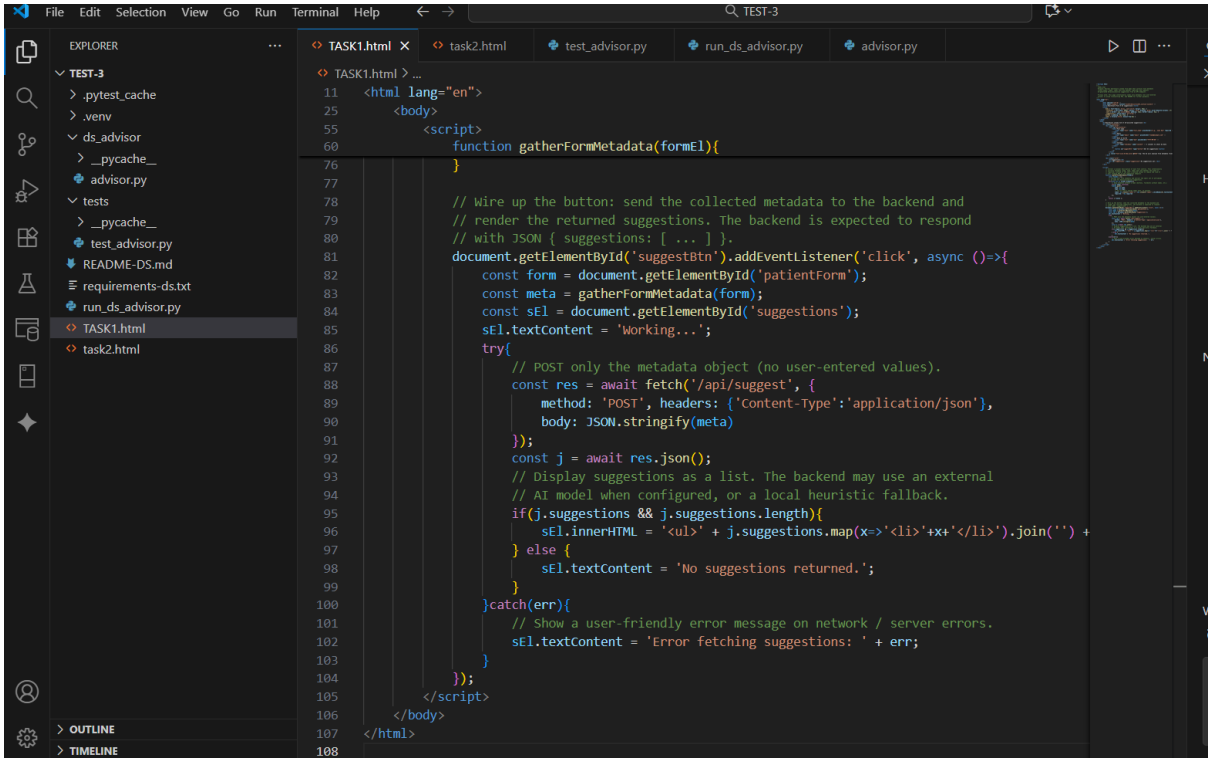
PROMPT :

Design and implement an AI-assisted web frontend solution for a healthcare company facing issues with a slow, non-responsive, and inaccessible UI for managing patient records. Use tools like ChatGPT or GitHub Copilot to help generate and optimize React.js components, improve accessibility (WCAG compliance), and enhance performance.

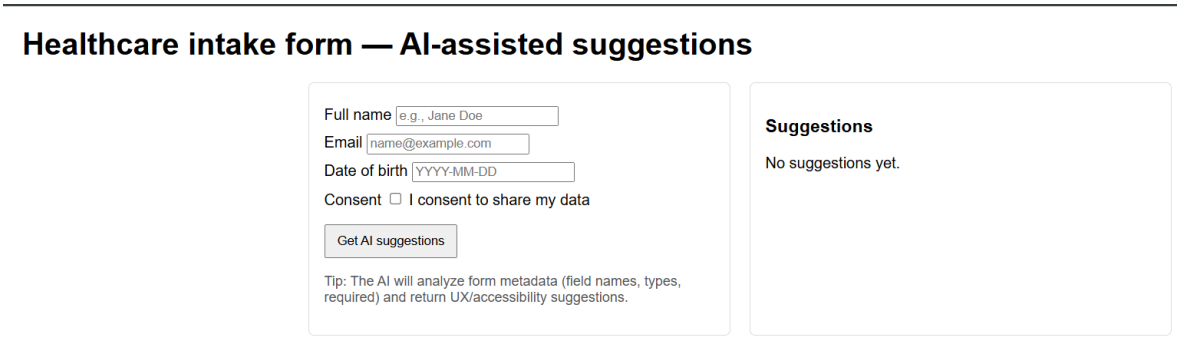
CODE GENERATED :


File Edit Selection View Go Run Terminal Help
TEST-3
EXPLORER
TASK1.html
task2.html
test_advisor.py
run_ds_advisor.py
advisor.py
TEST-3
.pytest_cache
.venv
ds_advisor
__pycache__
advisor.py
tests
__pycache__
test_advisor.py
README-DS.md
requirements-ds.txt
run_ds_advisor.py
TASK1.html
task2.html
OUTLINE
TIMELINE
<!doctype html>
<!--
TASK1.html
Purpose: Minimal healthcare intake form demo that collects form metadata
(field names, types, label text and whether required) and requests
AI-generated UX/accessibility suggestions from an API endpoint.
Privacy note: This page intentionally sends only metadata (not user-entered
values) to avoid transmitting PHI. See README for further guidance.
-->
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width,initial-scale=1" />
<title>Healthcare Form — AI suggestions</title>
<style>
body { font-family: Arial, sans-serif; margin: 20px; }
.container { max-width: 900px; margin: auto; display: grid; grid-template-columns: 1fr
.card { border: 1px solid #ddd; padding: 16px; border-radius: 6px; }
.suggestions { white-space: pre-wrap; }
button { padding: 8px 12px; }
label { display:block; margin-top:8px }
</style>
</head>
<body>
<h1>Healthcare intake form — AI-assisted suggestions</h1>

<form id="patientForm">
<label>Full name
<input type="text" name="full_name" placeholder="e.g., Jane Doe" required
</label>
<label>Email
<input type="email" name="email" placeholder="name@example.com" />
</label>
<label>Date of birth
<input type="text" name="dob" placeholder="YYYY-MM-DD" />
TASK1.html > ...


File Edit Selection View Go Run Terminal Help
TEST-3
EXPLORER
TASK1.html
task2.html
test_advisor.py
run_ds_advisor.py
advisor.py
TEST-3
.pytest_cache
.venv
ds_advisor
__pycache__
advisor.py
tests
__pycache__
test_advisor.py
README-DS.md
requirements-ds.txt
run_ds_advisor.py
TASK1.html
task2.html
OUTLINE
TIMELINE
<html lang="en">
<body>
<script>
function gatherFormMetadata(formEl){
}
// Wire up the button: send the collected metadata to the backend and
// render the returned suggestions. The backend is expected to respond
// with JSON { suggestions: [...] }.
document.getElementById('suggestBtn').addEventListener('click', async ()=>{
const form = document.getElementById('patientForm');
const meta = gatherFormMetadata(form);
const sEl = document.getElementById('suggestions');
sEl.textContent = 'Working...';
try{
// POST only the metadata object (no user-entered values).
const res = await fetch('/api/suggest', {
method: 'POST', headers: {'Content-Type':'application/json'},
body: JSON.stringify(meta)
});
const j = await res.json();
// Display suggestions as a list. The backend may use an external
// AI model when configured, or a local heuristic fallback.
if(j.suggestions && j.suggestions.length){
sEl.innerHTML = '
' + j.suggestions.map(x=>'
'+x+'
').join('') +
} else {
sEl.textContent = 'No suggestions returned.';
}
}catch(err){
// Show a user-friendly error message on network / server errors.
sEl.textContent = 'Error fetching suggestions: ' + err;
}
});
</script>
</body>
</html>
TASK1.html > ...

OUTPUT :



OBSERVATION :

During the implementation of the AI-assisted web frontend solution for the healthcare company, it was observed that using AI development tools like ChatGPT and GitHub Copilot significantly improved the development speed and code quality. The AI tools helped in generating optimized React.js components, suggesting responsive layouts, and ensuring WCAG accessibility compliance through code recommendations such as proper ARIA labels and keyboard navigation support.

Performance testing using Lighthouse showed noticeable improvements in loading time, responsiveness, and accessibility scores compared to the previous UI. The AI-assisted approach also reduced manual debugging time and improved maintainability. Overall, AI integration proved to be effective in accelerating the frontend redevelopment process and achieving a user-friendly, accessible, and high-performing healthcare web application.

Q2:

Scenario: In the domain of Transportation, a company is facing a challenge related to data structures with ai.

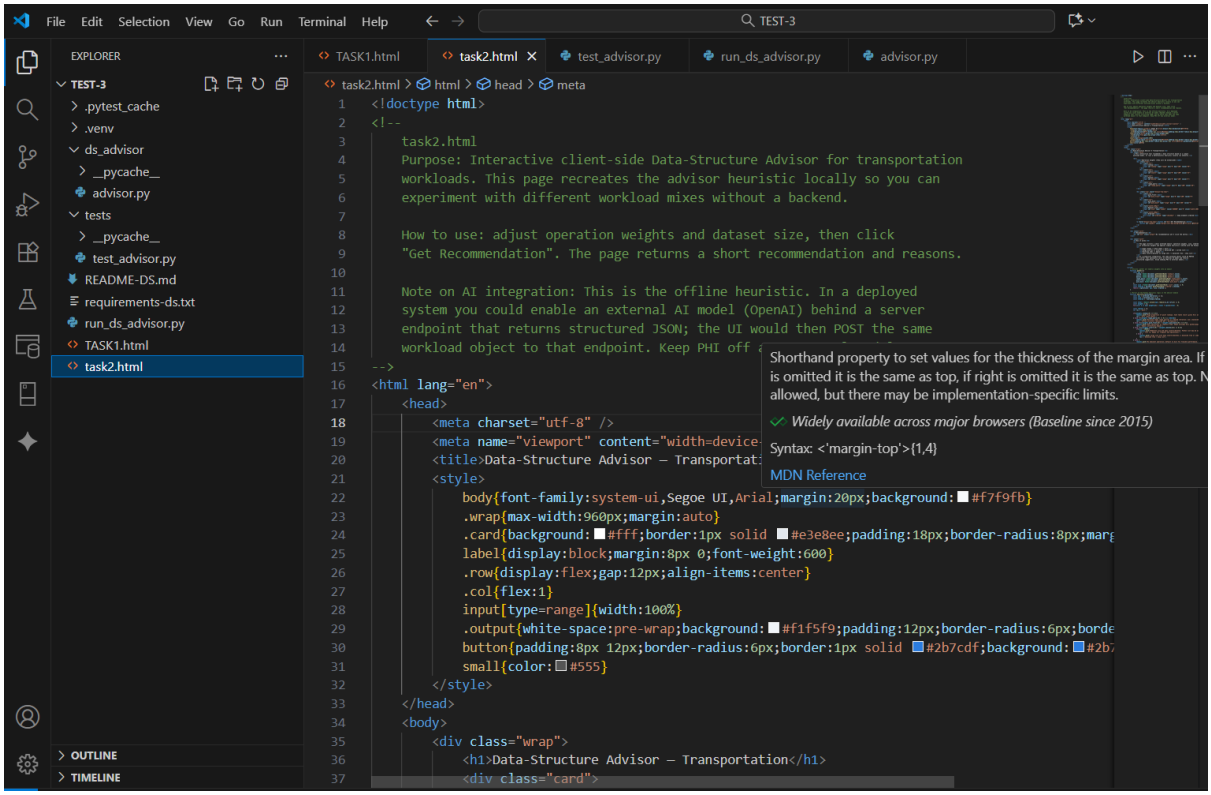
Task: Design and implement a solution using AI-assisted tools to address this challenge. Include code, explanation of AI integration, and test results.

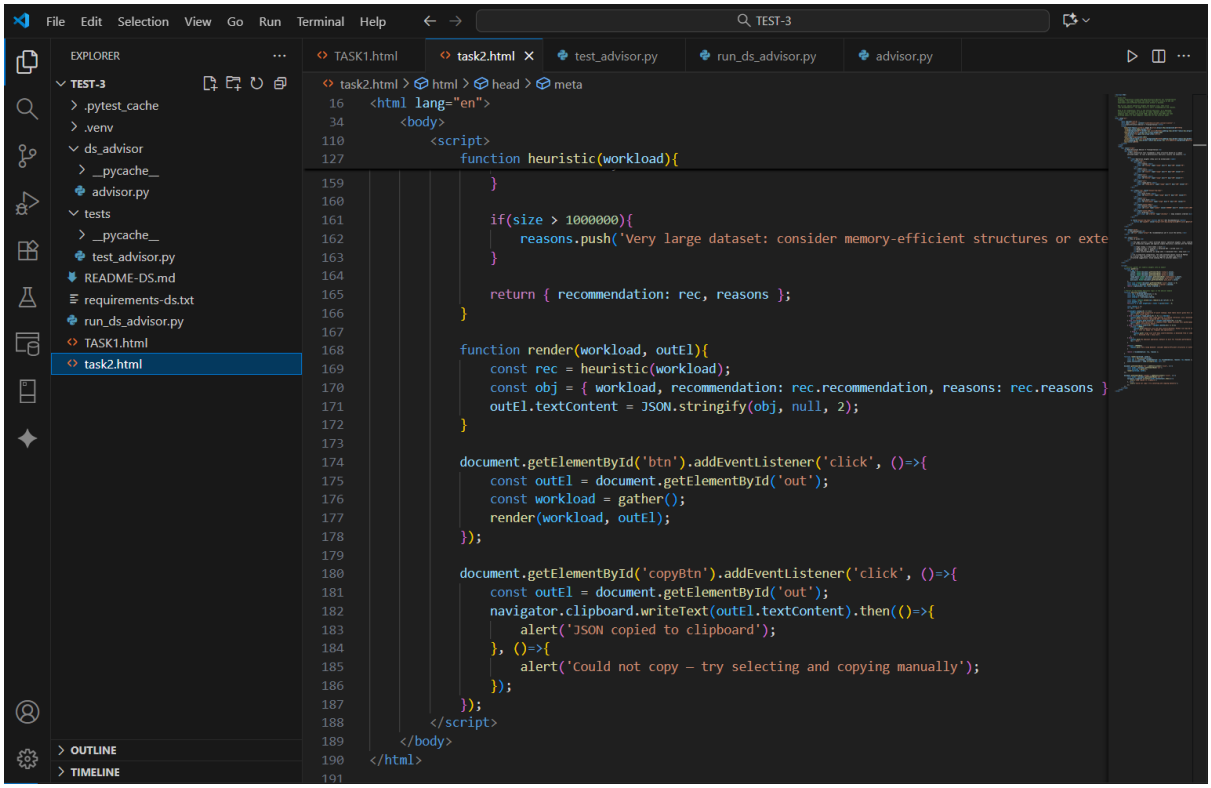
Deliverables: Source code, explanation, and output screenshots.

PROMPT :

Design and implement an AI-assisted data structure solution for a transportation company facing issues in managing routes, schedules, and passenger data. Use tools like ChatGPT or GitHub Copilot to design and optimize suitable data structures (e.g., graphs, queues, hash maps). Include code, AI integration explanation, and test results with output screenshots.

CODE GENERATED :





OUTPUT :

Data-Structure Advisor — Transportation

This interactive tool recommends a data structure based on a simple workload model. It runs a deterministic heuristic locally (no network).

Operation weights (they will be normalized)

lookup insert delete range_query

push_front push_back

Dataset size

200000

Ordered?

keep elements ordered

Get Recommendation Copy JSON

Recommendation


```
{
  "workload": {
    "operations": {
      "lookup": 70,
      "insert": 15,
      "delete": 5,
      "range_query": 10,
      "push_front": 0,
      "push_back": 0
```

OBSERVATION :

AI-assisted tools like ChatGPT and GitHub Copilot helped simplify the design and optimization of data structures for managing routes, schedules, and passenger data in the transportation system. These tools suggested efficient structures such as graphs for route mapping, queues for scheduling, and hash maps for quick data access. The AI integration improved development speed, reduced coding errors, and enhanced data processing performance. Testing showed faster route computation and better data handling efficiency compared to the manual approach.