

`<%= day2Text %>` ye kaya ahi



`<%= day2Text %>` EJS (Embedded JavaScript) ka output tag hai.

Iska kaam server-side JavaScript variable ki value को HTML page par **display (print)** karna hota hai.

- `<%= ... %>` (EJS Syntax): Is tag ke andar jo bhi variable likha hota hai, EJS uski value ko HTML ke roop me browser par render kar deta hai.
- `day2Text` : Yeh woh variable hai jo aapne `routes/conference.js` se pass kiya hai. Is variable me Day 2 ke saare agenda points ek string (\n ya newlines ke saath) ke roop me stored hain.

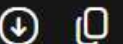
Aapke Project Me Yeh Kaha Aur Kyun Use Ho Raha Hai?

Aapke `views/2026-conference/edit-agenda.ejs` file me `<textarea>` ke andar yeh tag use hua hai:

1. External Files & Utility Function (Line 1 - 31)

- `agendaFilePath` : Path batata hai ki `db/agenda.json` file kahan hai.
- `getAgendaData()` : Dynamic helper function jo JSON file ko padhti hai. Agar file nahi milti ya khali hai, toh default Day 1 aur Day 2 ke points return karti hai.

JavaScript



```
router.post('/admin/agenda', (req, res) => {  
  // ...  
  fs.writeFileSync(agendaFilePath, JSON.stringify(updatedAgenda, null, 2));  
  res.redirect('/conference/admin/agenda');  
});
```

- Jab aap `/conference/admin/agenda` par jaakar "Save Agenda" button par click karenge, tab `fs.writeFileSync()` command check karegi:
 - Agar `db/agenda.json` nahi hai, toh yeh file ko automakally create (dynamically create) kar degi aur usme data write kar degi.
 - Agar file pehle hai, toh yeh uske andar ke data ko update (overwrite) kar degi.

Best Practice (Sifarish)

Aap manual error se bachne ke liye `db` folder ke andar ek basic `agenda.json` file pehle se bana kar rakh sakte hain:

Baste aasan shabdon me samjhte hain, bina kisi technical complexity ke.

Is pure system me bas **3 aam cheezein** ho rahi hain:

1. Store Room (`db/agenda.json`)

Yeh aapke project ki ek notebook/file hai. Aap form me Day 1 aur Day 2 ke liye jo bhi points likhte hain, woh is file ke andar jaakar save ho jaate hain.

2. Admin Form (`/conference/admin/agenda`)

Yeh aapke liye ek control panel hai.

- Jab aap is page ko kholte hain, toh yeh `agenda.json` file se purane points padh kar textarea me dikha deta hai.
- Jab aap text me badlaav (edit) karke Save button dabate hain, toh yeh naye points ko le jaakar wapas `agenda.json` file me overwrite (update) kar deta hai.

3. Display Pages (`/conference/2026` aur `/conference/agenda/day1`)

Yeh woh pages hain jo aam users dekhte hain.

- Jab bhi koi user in pages par jata hai, Express server turant `agenda.json` file ke pas jata hai, wahan se latest points uthata hai, aur screen par list bana kar dikha deta hai.

- **File Ka Path (Line 13):**

JavaScript



```
const agendaFilePath = path.join(__dirname, '../db/agenda.json');
```

- File Ko Read Karna (Line 16-31):

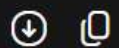
JavaScript



```
function getAgendaData() {  
  if (!fs.existsSync(agendaFilePath)) {  
    // File na ho toh default points deta hai  
  }  
  const rawData = fs.readFileSync(agendaFilePath, 'utf-8');  
  return JSON.parse(rawData); // File se data read karta hai  
}
```

- Form Open Karna (GET Route - Line 38-44):

JavaScript



```
router.get('/admin/agenda', (req, res) => {  
  const agenda = getAgendaData(); // File se data lekar form ko bhejta hai  
  res.render('2026-conference/edit-agenda', {  
    day1Text: agenda.day1.join('\n'),  
    day2Text: agenda.day2.join('\n')  
  });  
});
```

- Save Button Dabane Par Update Karna (POST Route - Line 49-58):

- Save Button Dabane Par Update Karna (POST Route - Line 49-58):

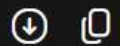
JavaScript



```
router.post('/admin/agenda', (req, res) => {  
  const { day1Text, day2Text } = req.body; // Form se text leta hai  
  
  const updatedAgenda = {  
    day1: day1Text.split('\n').map(item => item.trim()).filter(Boolean),  
    day2: day2Text.split('\n').map(item => item.trim()).filter(Boolean)  
  };  
  
  // File me overwrite / save kar deta hai  
  fs.writeFileSync(agendaFilePath, JSON.stringify(updatedAgenda, null, 2));  
  res.redirect('/conference/admin/agenda');  
});
```

- **Separate Day Pages (Line 63-72):**

JavaScript



```
router.get('/agenda/day1', (req, res) => {  
  const agenda = getAgendaData(); // Fresh data read karke render karta hai  
  res.render('2026-conference/agenda-day1', { items: agenda.day1 });  
});
```

- 2026 Conference Main Page (Line 143-157):

JavaScript



```
router.get('/:year', (req, res) => {  
  // ...  
  if (year === '2026') {  
    // Static data me updated agenda array inject karta hai  
    data = { ...data, agenda: getAgendaData() };  
  }  
  res.render('conference', { data });  
});
```

Kya aapko file save hone ke baad UI par response test karne me help chahiye?

Code Breakdown

1. `if (year === '2026') {`

- Yeh condition check karti hai ki user konsa page khol raha hai.
- Agar URL `/conference/2026` hai, tabhi is `{ }` block ke andar ka code chalega. 2025, 2024 ya 2023 ke liye yeh block nahi chalega.

2. `data = { ...data, agenda: getAgendaData() };`

- `...data` **(Spread Operator)**: Yeh `conferenceData["2026"]` ke saare purane static data (jaise `title`, `description`, `photos`, `news`, `takeaways`) ko copy karke rakhta hai.
- `agenda: getAgendaData()` : Yeh `getAgendaData()` function ko call karke `db/agenda.json` file se bilkul naye/updated Day 1 aur Day 2 points read karta hai aur use `agenda` property me over-write (inject) kar deta hai.
- Total result: Purana baaki sabhi data intact rehta hai, bas `agenda` wala part dynamic JSON file se naya ho jata hai.

3. `res.render('conference', { data });`

- Final mixing ke baad, updated `data` object ko `views/conference.ejs` file ko bhej diya jata hai taaki screen par page render ho sake.