

Scenario: You've inherited this snippet. It's meant to fetch, sort, and return posts—but sometimes it returns duplicates or hangs.

```
// postsController.js async

function getSortedPosts(req, res)

{

  const posts = await Posts.find(); // Mongoose model

  posts.sort((a, b) => b.created - a.created);

  res.json(posts); }

// router.js

router.get('/posts', async (req, res) => {

  await getSortedPosts(req, res);

  console.log('Done.');
```

});

1. Identify at least two problems (bugs or performance issues) here.

Answer:

1. No error handling can cause hanging
2. Client side sorting instead of database which cause performance issue
3. No limit or pagination which cause memory issue or duplicate
4. Date comparison might be fail if created is not Date datatype
5. No validation or sanitization

2. Propose corrections (code snippets or descriptions).

Implement pagination and sorting on server side

```
async function getSortedPosts(req, res) {
  try {
    const page = parseInt(req.query.page) || 1;
    const limit = parseInt(req.query.limit) || 20;
    const skip = (page - 1) * limit;

    const posts = await Posts.find()
      .sort({ created: -1 })
      .skip(skip)
```

```

        .limit(limit)
        .lean();

    res.json({
        posts,
        page,
        limit,
        hasMore: posts.length === limit
    });

} catch (error) {
    res.status(500).json({
        error: 'Failed to fetch posts',
        message: 'Internal server error'
    });
}
}
}

```

3. Explain in plain English why each fix matters.

If database query fail the function will throw an unhandled exception causing the request to hang,

Post.sort happen on client side when all the post fetch from database

With no limit can fetch million of record causing memory exhausting

Assume if Created is date object but it might be string then it will not perform proper comparison