Code review techniques in Software Engineering

Unit-5-part b

code review in software quality

It contributes to software quality by:

- Detecting bugs early in the development process
- Promoting coding standards and best practices
- Enhancing collaboration among team members
- Providing opportunities for knowledge sharing and skill development

Types of Software Reviews

- **1. Peer Review**
- **2. Walkthrough**
- **3. Technical Review**
- **4. Inspection**
- **5. Code Review**

Peer Review

- Informal review by colleagues
- Focus: Logic, clarity, and standards
- Often used during early development stages

Walkthrough

- Author-led review session
- Participants: Developers, testers, and stakeholders
- Goal: Understand and gather feedback on the artifact

Technical Review

- Formal, documented process
- Focus: Technical quality and adherence to standards
- Led by a trained moderator

Inspection

- Most rigorous review technique
- Steps: Planning → Overview → Preparation → Inspection → Rework → Follow-up
- Detects defects systematically and quantitatively

code review process typically involves several key stages:

1. **Preparation**: The author prepares the code for review, often including documentation or context about the changes.
2. **Review**: Reviewers analyze the code, looking for issues such as bugs, adherence to coding standards, and overall code quality.
3. **Feedback**: Reviewers provide feedback, which can include suggestions for improvements or identifying areas that need fixing.
4. **Revisions**: The author makes changes based on the feedback received.
5. **Approval**: Once the changes are made, the code is re-reviewed and approved for merging.

Types of code review techniques

1. Formal code review

- This technique involves a structured process, often documented with defined roles and responsibilities. It typically includes:
- A review team consisting of developers, testers, and other stakeholders.
- A predefined checklist to evaluate code against coding standards.
- Detailed documentation of findings and actions taken.

Types of code review techniques

2. Peer code review

- Peer code review, also known as informal review, involves colleagues examining each other's code.
- This technique encourages collaboration and knowledge sharing.
- **example** : A developer submits a pull request in GitHub, and a peer reviews it for logic errors and compliance with team coding standards.
- This back-and-forth discussion can lead to valuable insights and improvements in code quality.

Types of code review techniques

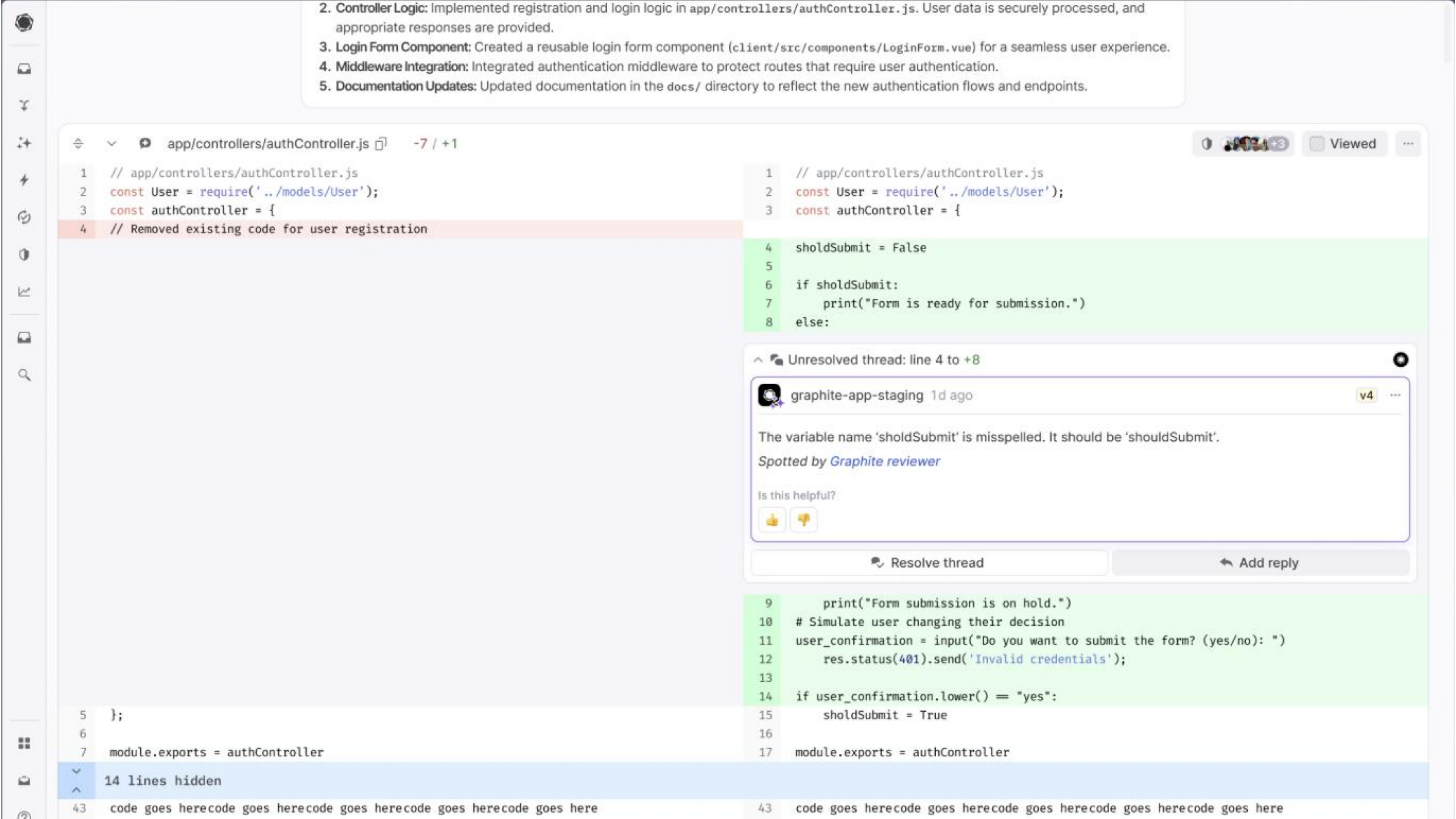
3. Pair programming

- Pair programming is a software development technique where two developers work together at a single workstation, with one writing the code (the "driver") while the other reviews the code as it's written (the "observer" or "navigator").
- This method encourages real-time feedback and knowledge sharing, helping to catch issues early and improve the quality of the code.

Types of code review techniques

4. Tool-assisted review

- Using code review tools can automate parts of the code review process and help identify issues related to code quality, maintainability, and adherence to coding standards.
- For example, a tool automatically detects bugs and errors before human reviewers even start, allowing your team to spend less time reviewing code and more time building.



2. **Controller Logic:** Implemented registration and login logic in `app/controllers/authController.js`. User data is securely processed, and appropriate responses are provided.
3. **Login Form Component:** Created a reusable login form component (`client/src/components/LoginForm.vue`) for a seamless user experience.
4. **Middleware Integration:** Integrated authentication middleware to protect routes that require user authentication.
5. **Documentation Updates:** Updated documentation in the `docs/` directory to reflect the new authentication flows and endpoints.

app/controllers/authController.js -7 / +1

```
1   // app/controllers/authController.js
2   const User = require('../models/User');
3   const authController = {
4   // Removed existing code for user registration
5   };
6
7   module.exports = authController
```

```
1   // app/controllers/authController.js
2   const User = require('../models/User');
3   const authController = {
4   sholdSubmit = False
5
6   if sholdSubmit:
7       print("Form is ready for submission.")
8   else:
```

Unresolved thread: line 4 to +8

graphite-app-staging 1d ago

The variable name 'sholdSubmit' is misspelled. It should be 'shouldSubmit'.

Spotted by Graphite reviewer

```
9       print("Form submission is on hold.")
10  # Simulate user changing their decision
11  user_confirmation = input("Do you want to submit the form? (yes/no): ")
12      res.status(401).send('Invalid credentials');
13
14  if user_confirmation.lower() == "yes":
15      sholdSubmit = True
16
17  module.exports = authController
```

14 lines hidden

```
43  code goes herecode goes herecode goes herecode goes herecode goes here
```

```
43  code goes herecode goes herecode goes herecode goes herecode goes here
```

code review techniques : A briefing

- Follow a checklist
- Use automated tools
- Code style guide enforcement: Ensure the code follows team or organization-wide style guides
- Peer reviews
- Focus on logic and structure
- *Encourage constructive feedback:Provide feedback that is specific, actionable, and focused on improving the code.
  - Avoid personal attacks and frame suggestions as improvements rather than criticisms.