**Experiment Report - 19 – test2\_guidewizard**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | CSCR? | Self-Detected? | LLM? |
| E01 | line 34 | Logic | √ |  | × |
| E02 | line 43 | Syntax | √ |  | √ |
| E03 | line 55 | Semantic | √ |  | × |
| E04 | line 15 (66) | Semantic |  | √ | × |
| E05 | line 29 (80) | Logic |  | √ | √ |
| E06 | line 52 (103) | Logic |  | × | √ |

Additional Errors Found by Self: 0

CSCR Rate: 100%

Self-Review Detection Rate: 67%

LLM Rate: 50%

1. **Source Code**
2. #include "guidewizard.h"
3. #include <QtWidgets>
4. GuideWizard::GuideWizard(QWidget \*parent)
5. : QWizard(parent)
6. {
7. setPage(Page\_Intro, new IntroPage);
8. setPage(Page\_CreateNewProject, new CreateProjectPage);
9. setPage(Page\_OpenExistingProject, new OpenProjectPage);
10. setPage(Page\_Conclusion, new ConclusionPage);
11. setStartId(Page\_Intro);
12. #ifndef Q\_OS\_MAC
13. setWizardStyle(ModernStyle);
14. #endif
15. setPixmap(QWizard::LogoPixmap, QPixmap(":/images/toolbar\_images/hmpp.png"));
16. setWindowTitle(tr("HMPP Tool Guide"));
17. }
18. CreateProjectPage::CreateProjectPage(QWidget \*parent)
19. : QWizardPage(parent)
20. {
21. setTitle(tr("Create a new project"));
22. projectNameLabel = new QLabel(tr("Project &Name:"));
23. projectNameLineEdit = new QLineEdit;
24. projectNameLabel->setBuddy(projectNameLineEdit);
25. projectPathLabel = new QLabel(tr("Project &Directory:"));
26. projectPathLineEdit = new QLineEdit;
27. projectPathLineEdit->setReadOnly(false);
28. projectPathLabel->setBuddy(projectPathLineEdit);
29. projectSpecPathLabel = new QLabel(tr("S&pecification Directory:"));
30. projectSpecPathEdit = new QLineEdit;
31. projectSpecPathEdit->setReadOnly(true);
32. projectSpecPathLabel->setBuddy(projectSpecPathEdit);
33. QPixmap pixmap(":/images/toolbar\_images/folder.png");
34. QPixmap scaledPixmap = pixmap.scaled(16, Qt::KeepAspectRatio); *// Adjust the size as needed*
35. *// Set the icon on the button*
36. QIcon icon(scaledPixmap);
37. openDirectoryButton = new QPushButton;
38. openDirectoryButton->setIcon(icon);
39. openDirectoryButton->resize(16);
40. connect(openDirectoryButton, &QPushButton::clicked, this, &CreateProjectPage::getProjectDirectory);
41. openSpecButton = new QPushButton;
42. openSpecButton->setIcon(icon);
43. openSpecButton->resize(16, 16);
44. connect(openSpecButton, &QPushButton::clicked, this, &CreateProjectPage::getSpecDirectory);
45. targetLanguageLabel = new QLabel(tr("Target &Language:"));
46. targetLanguageBox = new QComboBox;
47. targetLanguageBox->addItem("C++");
48. targetLanguageBox->addItem("Java");
49. targetLanguageBox->addItem("C#");
50. registerField("project name\*", projectNameLineEdit);
51. registerField("project path\*", projectPathLineEdit);
52. registerField("specification path\*", projectSpecPathEdit);
53. *//registerField("target language\*", targetLanguageBox);*
54. registerField("target language\*", targetLanguageBox, "currentText", SIGNAL(currentIndexChanged(int)));
55. QGridLayout\* layout = new QGridLayout;
56. layout->addWidget(projectNameLabel, 0, 0);
57. layout->addWidget(projectNameLineEdit, 0, 1);
58. layout->addWidget(projectPathLabel, 1, 0);
59. layout->addWidget(projectPathLineEdit, 1, 1);
60. layout->addWidget(openDirectoryButton, 1, 2);
61. layout->addWidget(projectSpecPathLabel, 2, 0);
62. layout->addWidget(projectSpecPathEdit, 2, 1);
63. layout->addWidget(openSpecButton, 2, 1);
64. layout->addWidget(targetLanguageLabel, 3, 0);
65. layout->addWidget(targetLanguageBox, 3, 1);
66. setLayout(layout);
67. }
68. int CreateProjectPage::nextId() const
69. {
70. return GuideWizard::Page\_Conclusion;
71. }
72. bool CreateProjectPage::validatePage()
73. {
74. if (projectNameLineEdit->text().isEmpty()) {
75. qDebug() << "projectName is empty.";
76. QMessageBox::warning(this, "Input Error", "Please enter your project name before proceeding.");
77. return false; *// Prevent going to the next page*
78. }
79. if (projectPathLineEdit->text().isEmpty()) {
80. qDebug() << "projectPath is empty";
81. QMessageBox::warning(this, "Input Error", "Please select your project directory before proceeding.");
82. return true; *// Prevent going to the next page*
83. }
84. return true; *// Allow going to the next page*
85. }
86. bool CreateProjectPage::isComplete() const
87. {
88. return targetLanguageBox->currentIndex() >= 0 && !projectNameLineEdit->text().isEmpty()
89. && !projectPathLineEdit->text().isEmpty();
90. }
91. void CreateProjectPage::getProjectDirectory()
92. {
93. const QString proDir = QFileDialog::getExistingDirectory(this, tr("Open Directory"), QDir::currentPath(),
94. QFileDialog::ShowDirsOnly | QFileDialog::DontResolveSymlinks);
95. if(!proDir.isEmpty()){
96. projectPathLineEdit->setText(proDir);
97. }
98. }
99. void CreateProjectPage::getSpecDirectory()
100. {
101. const QString filePath = QFileDialog::getOpenFileName(this);    *//打开文件对话框*
102. if(!filePath.isEmpty()){
103. projectSpecPathEdit->setText(filePath);
104. }
105. }