Cambridge Academy for Science and Technology

AQA Computer Science

Practical Computing Project

CRYPTOGRAPHY ONLINE

Author
J.P. Jacob Powell

Supervisor
B.C. Barry Cooper

February 1, 2018

Contents

Chapter 1

Analysis

1.1 What is Cryptology

Chapter 2

Documented Design

2.1 High-Level Overview

2.2 Description of Algorithms

2.3 Description of Data Structures

2.4 Design of Database

2.5 Design of user interface

2.6 Hardware/Software Selection

2.7 System Security and Integrity of Data

Chapter 3

Technical Solution

3.1 Testing Code Input

Here is some code.

```
/*
 * File: crypto_online_home.cc
 * Created: 19/12/2017 13:32
 * Finished:
 *
 * Description:
 *
 * Author: Jacob Powell
 */

#include "crypto_online_navigation_grid.h"

#include <Wt/WPushButton.h>

crypto_online_navigation_grid::crypto_online_navigation_grid() {
    populate_navigation_grid();
}


void crypto_online_navigation_grid::populate_navigation_grid() {
    this->setup_basic_contents();
    this->setup_symmetric_contents();
    this->setup_asymmetric_contents();
    this->setup_applications_contents();
    this->clear_grid();
}

void crypto_online_navigation_grid::setup_basic_contents() {
    this->title_cryptography_basics = Wt::cpp14::make_unique<Wt::WText>("Cryptography Basics");
    this->title_cryptography_basics->setStyleClass("navigation_grid_title");
    this->addWidget(std::move(this->title_cryptography_basics),0,1);

    this->cryptography_basics_concepts = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
            Wt::LinkType::InternalPath, "/basics/concepts"));
```

```
35      auto button_cryptography_basics_concepts = Wt::cpp14::make_unique<Wt::WPushButton>("Basic Concepts");
36      button_cryptography_basics_concepts->setStyleClass("navigation_grid_item");
37      this->cryptography_basics_concepts->addWidget(std::move(button_cryptography_basics_concepts));
38      this->addWidget(std::move(cryptography_basics_concepts),1,0, Wt::AlignmentFlag::Center);
39
40      this->cryptography_basics_real_world_applications = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
41              Wt::LinkType::InternalPath, "/basics/real-world-applications"));
42      auto button_cryptography_basics_real_world_applications =
43              Wt::cpp14::make_unique<Wt::WPushButton>("Real World Applications");
44      button_cryptography_basics_real_world_applications->setStyleClass("navigation_grid_item");
45      this->cryptography_basics_real_world_applications->addWidget(
46              std::move(button_cryptography_basics_real_world_applications));
47      this->addWidget(std::move(this->cryptography_basics_real_world_applications),1,1, Wt::AlignmentFlag::Center);
48
49      this->cryptography_basics_concepts_history = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
50              Wt::LinkType::InternalPath, "/basics/history"));
51      auto button_cryptography_basics_history = Wt::cpp14::make_unique<Wt::WPushButton>("History");
52      button_cryptography_basics_history->setStyleClass("navigation_grid_item");
53      this->cryptography_basics_concepts_history->addWidget(
54              std::move(button_cryptography_basics_history));
55      this->addWidget(std::move(this->cryptography_basics_concepts_history),1,2, Wt::AlignmentFlag::Center);
56  }
57
58  void crypto_online_navigation_grid::setup_symmetric_contents() {
59      this->title_symmetric = Wt::cpp14::make_unique<Wt::WText>("Symmetric Cryptography");
60      this->title_symmetric->setStyleClass("navigation_grid_title");
61      this->addWidget(std::move(this->title_symmetric),2,1);
62
63      this->subtitle_introduction_to_symmetric_cryptography = Wt::cpp14::make_unique<Wt::WText>("Introduction to Symmetric Cryptography");
64      this->subtitle_introduction_to_symmetric_cryptography->setStyleClass("navigation_grid_subtitle");
65      this->addWidget(std::move(this->subtitle_introduction_to_symmetric_cryptography),3,1);
66
67      this->symmetric_modular_arithmetic = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
68              Wt::LinkType::InternalPath, "/symmetric/modular-arithmetic"));
69      auto button_cryptography_symmetric_modular_arithmetic = Wt::cpp14::make_unique<Wt::WPushButton>("Modular Arithmetic");
70      button_cryptography_symmetric_modular_arithmetic->setStyleClass("
```

```
        navigation_grid_item");
71      this->symmetric_modular_arithmetic->addWidget(
72              std::move(button_cryptography_symmetric_modular_arithmetic));
73      this->addWidget(std::move(this->symmetric_modular_arithmetic),4,0, Wt↩
            ::AlignmentFlag::Center);
74
75      this->symmetric_cipher_types = Wt::cpp14::make_unique<Wt::WAnchor>(Wt↩
            ::WLink(
76              Wt::LinkType::InternalPath, "/symmetric/cipher-types"));
77      auto button_cryptography_symmetric_cipher_types = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Cipher Types");
78      button_cryptography_symmetric_cipher_types->setStyleClass("↩
            navigation_grid_item");
79      this->symmetric_cipher_types->addWidget(std::move(↩
            button_cryptography_symmetric_cipher_types));
80      this->addWidget(std::move(this->symmetric_cipher_types),4,1, Wt::↩
            AlignmentFlag::Center);
81
82      this->symmetric_historical_ciphers = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
83              Wt::LinkType::InternalPath, "/symmetric/historical-ciphers"))↩
                ;
84      auto button_cryptography_symmetric_historical_ciphers = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Historical Ciphers");
85      button_cryptography_symmetric_historical_ciphers->setStyleClass("↩
            navigation_grid_item");
86      this->symmetric_historical_ciphers->addWidget(std::move(↩
            button_cryptography_symmetric_historical_ciphers));
87      this->addWidget(std::move(this->symmetric_historical_ciphers),4,2, Wt↩
            ::AlignmentFlag::Center);
88
89      this->symmetric_randomness = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
            WLink(
90              Wt::LinkType::InternalPath, "/symmetric/random-number-↩
                generators"));
91      auto button_cryptography_symmetric_randomness = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Random Number Generators");
92      button_cryptography_symmetric_randomness->setStyleClass("↩
            navigation_grid_item");
93      this->symmetric_randomness->addWidget(std::move(↩
            button_cryptography_symmetric_randomness));
94      this->addWidget(std::move(this->symmetric_randomness),5,0, Wt::↩
            AlignmentFlag::Center);
95
96      this->symmetric_stream_ciphers = Wt::cpp14::make_unique<Wt::WAnchor>(↩
            Wt::WLink(
97              Wt::LinkType::InternalPath, "/symmetric/stream-ciphers"));
98      auto button_cryptography_symmetric_stream_ciphers = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Stream Ciphers");
99      button_cryptography_symmetric_stream_ciphers->setStyleClass("↩
            navigation_grid_item");
100     this->symmetric_stream_ciphers->addWidget(std::move(↩
            button_cryptography_symmetric_stream_ciphers));
101     this->addWidget(std::move(this->symmetric_stream_ciphers),5,1, Wt::↩
            AlignmentFlag::Center);
102
103     this->symmetric_linear_feedback_shift_registers = Wt::cpp14::↩
            make_unique<Wt::WAnchor>(Wt::WLink(
```

```
104             Wt::LinkType::InternalPath, "/symmetric/stream-ciphers"));
105     auto button_cryptography_symmetric_lfsr = Wt::cpp14::make_unique<Wt::↩
            WPushButton>("Linear Feedback Shift Registers");
106     button_cryptography_symmetric_lfsr->setStyleClass("↩
            navigation_grid_item");
107     this->symmetric_linear_feedback_shift_registers->addWidget(std::move(↩
            button_cryptography_symmetric_lfsr));
108     this->addWidget(std::move(this->↩
            symmetric_linear_feedback_shift_registers),5,2, Wt::AlignmentFlag↩
            ::Center);
109
110     this->subtitle_des = Wt::cpp14::make_unique<Wt::WText>("DES: Data ↩
            Encryption Standard");
111     this->subtitle_des->setStyleClass("navigation_grid_subtitle");
112     this->addWidget(std::move(subtitle_des),6,1);
113
114     this->symmetric_des_overview = Wt::cpp14::make_unique<Wt::WAnchor>(Wt↩
            ::WLink(
115             Wt::LinkType::InternalPath, "/symmetric/des-overview"));
116     auto button_cryptography_symmetric_des_overview = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("DES Overview");
117     button_cryptography_symmetric_des_overview->setStyleClass("↩
            navigation_grid_item");
118     this->symmetric_des_overview->addWidget(std::move(↩
            button_cryptography_symmetric_des_overview));
119     this->addWidget(std::move(this->symmetric_des_overview),7,0, Wt::↩
            AlignmentFlag::Center);
120
121     this->symmetric_des_feistal_networks = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
122             Wt::LinkType::InternalPath, "/symmetric/feistal-networks"));
123     auto button_cryptography_symmetric_des_feistal_networks = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Feistal Networks");
124     button_cryptography_symmetric_des_feistal_networks->setStyleClass("↩
            navigation_grid_item");
125     this->symmetric_des_feistal_networks->addWidget(std::move(↩
            button_cryptography_symmetric_des_feistal_networks));
126     this->addWidget(std::move(this->symmetric_des_feistal_networks),7,1, ↩
            Wt::AlignmentFlag::Center);
127
128     this->symmetric_des_internals = Wt::cpp14::make_unique<Wt::WAnchor>(↩
            Wt::WLink(
129             Wt::LinkType::InternalPath, "/symmetric/des-internals"));
130     auto button_cryptography_symmetric_des_internals = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("DES Internals");
131     button_cryptography_symmetric_des_internals->setStyleClass("↩
            navigation_grid_item");
132     this->symmetric_des_internals->addWidget(std::move(↩
            button_cryptography_symmetric_des_internals));
133     this->addWidget(std::move(this->symmetric_des_internals),7,2, Wt::↩
            AlignmentFlag::Center);
134
135     this->symmetric_des_decryption = Wt::cpp14::make_unique<Wt::WAnchor>(↩
            Wt::WLink(
136             Wt::LinkType::InternalPath, "/symmetric/des-decryption"));
137     auto button_cryptography_symmetric_des_decryption = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("DES Decryption");
138     button_cryptography_symmetric_des_decryption->setStyleClass("↩
```

```
        navigation_grid_item");
139     this->symmetric_des_decryption->addWidget(std::move(↩
            button_cryptography_symmetric_des_decryption));
140     this->addWidget(std::move(this->symmetric_des_decryption),8,0, Wt::↩
            AlignmentFlag::Center);
141
142     this->symmetric_des_security = Wt::cpp14::make_unique<Wt::WAnchor>(Wt↩
            ::WLink(
143             Wt::LinkType::InternalPath, "/symmetric/des-security"));
144     auto button_cryptography_symmetric_des_security = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("DES Security");
145     button_cryptography_symmetric_des_security->setStyleClass("↩
            navigation_grid_item");
146     this->symmetric_des_security->addWidget(std::move(↩
            button_cryptography_symmetric_des_security));
147     this->addWidget(std::move(this->symmetric_des_security),8,1, Wt::↩
            AlignmentFlag::Center);
148
149     this->symmetric_des_alternatives = Wt::cpp14::make_unique<Wt::WAnchor↩
            >(Wt::WLink(
150             Wt::LinkType::InternalPath, "/symmetric/des-alternatives"));
151     auto button_cryptography_symmetric_des_alternatives = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("DES Alternatives");
152     button_cryptography_symmetric_des_alternatives->setStyleClass("↩
            navigation_grid_item");
153     this->symmetric_des_alternatives->addWidget(std::move(↩
            button_cryptography_symmetric_des_alternatives));
154     this->addWidget(std::move(this->symmetric_des_alternatives),8,2, Wt::↩
            AlignmentFlag::Center);
155
156     this->subtitle_aes = Wt::cpp14::make_unique<Wt::WText>("AES: Advanced↩
             Encryption Standard");
157     this->subtitle_aes->setStyleClass("navigation_grid_subtitle");
158     this->addWidget(std::move(subtitle_aes),9,1);
159
160     this->symmetric_aes_overview = Wt::cpp14::make_unique<Wt::WAnchor>(Wt↩
            ::WLink(
161             Wt::LinkType::InternalPath, "/symmetric/aes-overview"));
162     auto button_cryptography_symmetric_aes_overview = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("AES Overview");
163     button_cryptography_symmetric_aes_overview->setStyleClass("↩
            navigation_grid_item");
164     this->symmetric_aes_overview->addWidget(std::move(↩
            button_cryptography_symmetric_aes_overview));
165     this->addWidget(std::move(this->symmetric_aes_overview),10,0, Wt::↩
            AlignmentFlag::Center);
166
167     this->symmetric_aes_galois_fields = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
168             Wt::LinkType::InternalPath, "/symmetric/aes-galois-fields"));
169     auto button_cryptography_symmetric_aes_galois_fields = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("AES Galois Fields");
170     button_cryptography_symmetric_aes_galois_fields->setStyleClass("↩
            navigation_grid_item");
171     this->symmetric_aes_galois_fields->addWidget(std::move(↩
            button_cryptography_symmetric_aes_galois_fields));
172     this->addWidget(std::move(this->symmetric_aes_galois_fields),10,1, Wt↩
            ::AlignmentFlag::Center);
```

```
173
174     this->symmetric_aes_internals = Wt::cpp14::make_unique<Wt::WAnchor>(↩
            Wt::WLink(
175             Wt::LinkType::InternalPath, "/symmetric/aes-internals"));
176     auto button_cryptography_symmetric_aes_internals = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("AES Internals");
177     button_cryptography_symmetric_aes_internals->setStyleClass("↩
            navigation-grid-item");
178     this->symmetric_aes_internals->addWidget(std::move(↩
            button_cryptography_symmetric_aes_internals));
179     this->addWidget(std::move(this->symmetric_aes_internals),10,2, Wt::↩
            AlignmentFlag::Center);
180
181     this->symmetric_aes_decryption = Wt::cpp14::make_unique<Wt::WAnchor>(↩
            Wt::WLink(
182             Wt::LinkType::InternalPath, "/symmetric/aes-decryption"));
183     auto button_cryptography_symmetric_aes_decryption = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("AES Decryption");
184     button_cryptography_symmetric_aes_decryption->setStyleClass("↩
            navigation-grid-item");
185     this->symmetric_aes_decryption->addWidget(std::move(↩
            button_cryptography_symmetric_aes_decryption));
186     this->addWidget(std::move(this->symmetric_aes_decryption),11,0, Wt::↩
            AlignmentFlag::Center);
187
188     this->symmetric_aes_implementations_hardware_software = Wt::cpp14::↩
            make_unique<Wt::WAnchor>(Wt::WLink(
189             Wt::LinkType::InternalPath, "/symmetric/aes-implementations-↩
                hardware-software"));
190     auto button_cryptography_symmetric_aes_implementations =
191             Wt::cpp14::make_unique<Wt::WPushButton>("AES Implementations ↩
                in Hardware and Software");
192     button_cryptography_symmetric_aes_implementations->setStyleClass("↩
            navigation-grid-item");
193     this->symmetric_aes_implementations_hardware_software->addWidget(std↩
            ::move(button_cryptography_symmetric_aes_implementations));
194     this->addWidget(std::move(this->↩
            symmetric_aes_implementations_hardware_software),11,1, Wt::↩
            AlignmentFlag::Center);
195
196     this->symmetric_aes_example = Wt::cpp14::make_unique<Wt::WAnchor>(Wt↩
            ::WLink(
197             Wt::LinkType::InternalPath, "/symmetric/aes-example"));
198     auto button_cryptography_symmetric_aes_example = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("AES Example");
199     button_cryptography_symmetric_aes_example->setStyleClass("↩
            navigation-grid-item");
200     this->symmetric_aes_example->addWidget(std::move(↩
            button_cryptography_symmetric_aes_example));
201     this->addWidget(std::move(this->symmetric_aes_example),11,2, Wt::↩
            AlignmentFlag::Center);
202
203     this->subtitle_more_about_block_ciphers = Wt::cpp14::make_unique<Wt::↩
            WText>("More about Block Ciphers");
204     this->subtitle_more_about_block_ciphers->setStyleClass("↩
            navigation-grid-subtitle");
205     this->addWidget(std::move(this->subtitle_more_about_block_ciphers)↩
            ,12,1);
```

```
206
207     this->symmetric_more_about_block_ciphers_modes_of_operation = Wt::↩
            cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
208             Wt::LinkType::InternalPath, "/symmetric/modes-of-operation"))↩
                ;
209     auto ↩
            button_cryptography_symmetric_more_about_block_ciphers_modes_of_operation↩
             =
210             Wt::cpp14::make_unique<Wt::WPushButton>("Modes of Operation")↩
                ;
211     button_cryptography_symmetric_more_about_block_ciphers_modes_of_operation↩
            ->setStyleClass("navigation-grid-item");
212     this->symmetric_more_about_block_ciphers_modes_of_operation->↩
            addWidget(
213             std::move(↩
                    button_cryptography_symmetric_more_about_block_ciphers_modes_of_operatio
                    ));
214     this->addWidget(std::move(this->↩
            symmetric_more_about_block_ciphers_modes_of_operation),13,0, Wt::↩
            AlignmentFlag::Center);
215
216     this->symmetric_more_about_block_ciphers_increasing_security = Wt::↩
            cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
217             Wt::LinkType::InternalPath, "/symmetric/increasing-block-↩
                cipher-security"));
218     auto ↩
            button_cryptography_symmetric_more_about_block_ciphers_increasing_security↩
             =
219             Wt::cpp14::make_unique<Wt::WPushButton>("Increasing security ↩
                of Block Ciphers");
220     button_cryptography_symmetric_more_about_block_ciphers_increasing_security↩
             ->setStyleClass("navigation-grid-item");
221     this->symmetric_more_about_block_ciphers_increasing_security->↩
            addWidget(
222             std::move(↩
                    button_cryptography_symmetric_more_about_block_ciphers_increasing_securi
                    ));
223     this->addWidget(std::move(this->↩
            symmetric_more_about_block_ciphers_increasing_security),13,1, Wt::↩
            AlignmentFlag::Center);
224
225     this->↩
            symmetric_more_about_block_ciphers_revisit_of_exhaustive_key_search↩
             = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
226             Wt::LinkType::InternalPath, "/symmetric/revisit-of-exhaustive↩
                -key-search"));
227     auto ↩
            button_cryptography_symmetric_more_about_block_ciphers_revisit_of_exhaustive_key
             =
228             Wt::cpp14::make_unique<Wt::WPushButton>("Revisit of ↩
                Exhaustive Key Search");
229     button_cryptography_symmetric_more_about_block_ciphers_revisit_of_exhaustive_key_sea
            ->setStyleClass("navigation-grid-item");
230     this->↩
            symmetric_more_about_block_ciphers_revisit_of_exhaustive_key_search↩
            ->addWidget(
231             std::move(↩
                    button_cryptography_symmetric_more_about_block_ciphers_revisit_of_exhaus
```

```
                ));
232     this->addWidget(std::move(this->↩
            symmetric_more_about_block_ciphers_revisit_of_exhaustive_key_search↩
            ),13,2, Wt::AlignmentFlag::Center);
233 }
234
235 void crypto_online_navigation_grid::setup_asymmetric_contents() {
236     this->title_asymmetric = Wt::cpp14::make_unique<Wt::WText>("↩
            Asymmetric Cryptography");
237     this->title_asymmetric->setStyleClass("navigation-grid-title");
238     this->addWidget(std::move(this->title_asymmetric),14,1);
239
240     this->subtitle_introduction_to_asymmetric_cryptography = Wt::cpp14::↩
            make_unique<Wt::WText>("Introduction to Asymmetric Cryptography");
241     this->subtitle_introduction_to_asymmetric_cryptography->setStyleClass↩
            ("navigation-grid-subtitle");
242     this->addWidget(std::move(this->↩
            subtitle_introduction_to_asymmetric_cryptography),15,1);
243
244     this->asymmetric_symmetric_vs_asymmetric = Wt::cpp14::make_unique<Wt↩
            ::WAnchor>(Wt::WLink(
245             Wt::LinkType::InternalPath, "/asymmetric/symmetric-vs-↩
                asymmetric"));
246     auto button_cryptography_asymmetric_vs_symmetric = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Symmetric vs Asymmetric");
247     button_cryptography_asymmetric_vs_symmetric->setStyleClass("↩
            navigation-grid-item");
248     this->asymmetric_symmetric_vs_asymmetric->addWidget(std::move(↩
            button_cryptography_asymmetric_vs_symmetric));
249     this->addWidget(std::move(asymmetric_symmetric_vs_asymmetric),16,0, ↩
            Wt::AlignmentFlag::Center);
250
251     this->asymmetric_authenticity_of_public_keys = Wt::cpp14::make_unique↩
            <Wt::WAnchor>(Wt::WLink(
252             Wt::LinkType::InternalPath, "/asymmetric/authenticity-of-↩
                public-keys"));
253     auto button_cryptography_asymmetric_authenticity_of_public_keys = Wt↩
            ::cpp14::make_unique<Wt::WPushButton>("Authenticity of Public Keys↩
            ");
254     button_cryptography_asymmetric_authenticity_of_public_keys->↩
            setStyleClass("navigation-grid-item");
255     this->asymmetric_authenticity_of_public_keys->addWidget(std::move(↩
            button_cryptography_asymmetric_authenticity_of_public_keys));
256     this->addWidget(std::move(asymmetric_authenticity_of_public_keys)↩
            ,16,1, Wt::AlignmentFlag::Center);
257
258     this->asymmetric_key_lengths_and_security_levels = Wt::cpp14::↩
            make_unique<Wt::WAnchor>(Wt::WLink(
259             Wt::LinkType::InternalPath, "/asymmetric/key-lengths-and-↩
                security-levels"));
260     auto button_cryptography_asymmetric_key_lengths_security_levels = Wt↩
            ::cpp14::make_unique<Wt::WPushButton>("Key Lengths and Security ↩
            Lengths");
261     button_cryptography_asymmetric_key_lengths_security_levels->↩
            setStyleClass("navigation-grid-item");
262     this->asymmetric_key_lengths_and_security_levels->addWidget(std::move↩
            (button_cryptography_asymmetric_key_lengths_security_levels));
263     this->addWidget(std::move(asymmetric_key_lengths_and_security_levels)↩
```

```
        ,16,2, Wt::AlignmentFlag::Center);
264
265     this->subtitle_basic_number_theory_for_pk = Wt::cpp14::make_unique<Wt↩
            ::WText>("Basic Number Theory for Public-Key Algorithms");
266     this->subtitle_basic_number_theory_for_pk->setStyleClass("↩
            navigation-grid-subtitle");
267     this->addWidget(std::move(this->subtitle_basic_number_theory_for_pk)↩
            ,17,1);
268
269     this->asymmetric_euclidean_algorithm = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
270             Wt::LinkType::InternalPath, "/asymmetric/ea-eea"));
271     auto button_cryptography_asymmetric_euclidean_algorithm = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("EA and EEA Algorithms");
272     button_cryptography_asymmetric_euclidean_algorithm->setStyleClass("↩
            navigation-grid-item");
273     this->asymmetric_euclidean_algorithm->addWidget(std::move(↩
            button_cryptography_asymmetric_euclidean_algorithm));
274     this->addWidget(std::move(asymmetric_euclidean_algorithm),18,0, Wt::↩
            AlignmentFlag::Center);
275
276     this->asymmetric_eulers_phi_function = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
277             Wt::LinkType::InternalPath, "/asymmetric/euler's-phi-function↩
                ");
278     auto button_cryptography_asymmetric_phi_function = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Euler's Phi Function");
279     button_cryptography_asymmetric_phi_function->setStyleClass("↩
            navigation-grid-item");
280     this->asymmetric_eulers_phi_function->addWidget(std::move(↩
            button_cryptography_asymmetric_phi_function));
281     this->addWidget(std::move(asymmetric_eulers_phi_function),18,1, Wt::↩
            AlignmentFlag::Center);
282
283     this->asymmetric_fermats_little_theorem_and_eulers_theorem = Wt::↩
            cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
284             Wt::LinkType::InternalPath, "/asymmetric/fermats-little-↩
                theorem-eulers-theorem"));
285     auto button_cryptography_fermats_little_theorem_and_eulers_theorem= ↩
            Wt::cpp14::make_unique<Wt::WPushButton>("Fermats Little Theorem ↩
            and Eulers Theorem");
286     button_cryptography_fermats_little_theorem_and_eulers_theorem->↩
            setStyleClass("navigation-grid-item");
287     this->asymmetric_fermats_little_theorem_and_eulers_theorem->addWidget↩
            (std::move(↩
            button_cryptography_fermats_little_theorem_and_eulers_theorem));
288     this->addWidget(std::move(↩
            asymmetric_fermats_little_theorem_and_eulers_theorem),18,2, Wt::↩
            AlignmentFlag::Center);
289
290     this->subtitle_rsa_cryptosystem = Wt::cpp14::make_unique<Wt::WText>("↩
            The RSA Cryptosystem");
291     this->subtitle_rsa_cryptosystem->setStyleClass("↩
            navigation-grid-subtitle");
292     this->addWidget(std::move(this->subtitle_rsa_cryptosystem),19,1);
293
294     this->asymmetric_rsa_introduction = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
```

```
295             Wt::LinkType::InternalPath, "/asymmetric/rsa-introduction"));
296     auto button_cryptography_asymmetric_rsa_introduction = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("RSA Introduction");
297     button_cryptography_asymmetric_rsa_introduction->setStyleClass("↩
            navigation-grid-item");
298     this->asymmetric_rsa_introduction->addWidget(std::move(↩
            button_cryptography_asymmetric_rsa_introduction));
299     this->addWidget(std::move(asymmetric_rsa_introduction),20,0, Wt::↩
            AlignmentFlag::Center);
300
301     this->asymmetric_rsa_encryption_decryption = Wt::cpp14::make_unique<↩
            Wt::WAnchor>(Wt::WLink(
302             Wt::LinkType::InternalPath, "/asymmetric/rsa-encryption-↩
                decryption"));
303     auto button_cryptography_asymmetric_rsa_encryption_decryption = Wt::↩
            cpp14::make_unique<Wt::WPushButton>("RSA Encryption and Decryption↩
            ");
304     button_cryptography_asymmetric_rsa_encryption_decryption->↩
            setStyleClass("navigation-grid-item");
305     this->asymmetric_rsa_encryption_decryption->addWidget(std::move(↩
            button_cryptography_asymmetric_rsa_encryption_decryption));
306     this->addWidget(std::move(asymmetric_rsa_encryption_decryption),20,1,↩
             Wt::AlignmentFlag::Center);
307
308     this->asymmetric_rsa_key_generation = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
309             Wt::LinkType::InternalPath, "/asymmetric/rsa-key-generation")↩
                );
310     auto button_cryptography_asymmetric_rsa_key_generation = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("RSA Key Generation");
311     button_cryptography_asymmetric_rsa_key_generation->setStyleClass("↩
            navigation-grid-item");
312     this->asymmetric_rsa_key_generation->addWidget(std::move(↩
            button_cryptography_asymmetric_rsa_key_generation));
313     this->addWidget(std::move(asymmetric_rsa_key_generation),20,2, Wt::↩
            AlignmentFlag::Center);
314
315     this->asymmetric_rsa_encryption_decryption_fast_exponentiation= Wt::↩
            cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
316             Wt::LinkType::InternalPath, "/asymmetric/rsa-fast-↩
                exponentiation"));
317     auto ↩
            button_cryptography_asymmetric_rsa_encryption_decryption_fast_exponentiation↩
             =
318             Wt::cpp14::make_unique<Wt::WPushButton>("RSA Encryption and ↩
                Decryption with Fast Exponentiation");
319     button_cryptography_asymmetric_rsa_encryption_decryption_fast_exponentiation↩
            ->setStyleClass("navigation-grid-item");
320     this->asymmetric_rsa_encryption_decryption_fast_exponentiation->↩
            addWidget(std::move(↩
            button_cryptography_asymmetric_rsa_encryption_decryption_fast_exponentiation↩
            ));
321     this->addWidget(std::move(↩
            asymmetric_rsa_encryption_decryption_fast_exponentiation),21,0, Wt↩
            ::AlignmentFlag::Center);
322
323     this->asymmetric_rsa_speed_up_techniques = Wt::cpp14::make_unique<Wt↩
            ::WAnchor>(Wt::WLink(
```

```
324             Wt::LinkType::InternalPath, "/asymmetric/rsa-speed-up-techniques"));
325     auto button_cryptography_asymmetric_rsa_speed_up_techniques = Wt::cpp14::make_unique<Wt::WPushButton>("RSA Speed Up Techniques");
326     button_cryptography_asymmetric_rsa_speed_up_techniques->setStyleClass("navigation_grid_item");
327     this->asymmetric_rsa_speed_up_techniques->addWidget(std::move(button_cryptography_asymmetric_rsa_speed_up_techniques));
328     this->addWidget(std::move(asymmetric_rsa_speed_up_techniques),21,1, Wt::AlignmentFlag::Center);
329
330     this->asymmetric_rsa_finding_large_primes = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
331             Wt::LinkType::InternalPath, "/asymmetric/rsa-finding-large-primes"));
332     auto button_cryptography_asymmetric_rsa_finding_large_primes = Wt::cpp14::make_unique<Wt::WPushButton>("RSA Finding Large Primes");
333     button_cryptography_asymmetric_rsa_finding_large_primes->setStyleClass("navigation_grid_item");
334     this->asymmetric_rsa_finding_large_primes->addWidget(std::move(button_cryptography_asymmetric_rsa_finding_large_primes));
335     this->addWidget(std::move(asymmetric_rsa_finding_large_primes),21,2, Wt::AlignmentFlag::Center);
336
337     this->asymmetric_rsa_padding = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
338             Wt::LinkType::InternalPath, "/asymmetric/rsa-padding"));
339     auto button_cryptography_asymmetric_rsa_padding = Wt::cpp14::make_unique<Wt::WPushButton>("RSA Padding");
340     button_cryptography_asymmetric_rsa_padding->setStyleClass("navigation_grid_item");
341     this->asymmetric_rsa_padding->addWidget(std::move(button_cryptography_asymmetric_rsa_padding));
342     this->addWidget(std::move(asymmetric_rsa_padding),22,0, Wt::AlignmentFlag::Center);
343
344     this->asymmetric_rsa_attacks = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
345             Wt::LinkType::InternalPath, "/asymmetric/rsa-finding-large-primes"));
346     auto button_cryptography_asymmetric_rsa_attacks = Wt::cpp14::make_unique<Wt::WPushButton>("RSA Attacks");
347     button_cryptography_asymmetric_rsa_attacks->setStyleClass("navigation_grid_item");
348     this->asymmetric_rsa_attacks->addWidget(std::move(button_cryptography_asymmetric_rsa_attacks));
349     this->addWidget(std::move(asymmetric_rsa_attacks),22,1, Wt::AlignmentFlag::Center);
350
351     this->asymmetric_rsa_implementations_in_hardware_software = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
352             Wt::LinkType::InternalPath, "/asymmetric/rsa-finding-large-primes"));
353     auto button_cryptography_asymmetric_rsa_implementations_in_hardware_software =
354             Wt::cpp14::make_unique<Wt::WPushButton>("RSA Implementations in Hardware and Software");
```

```
355     button_cryptography_asymmetric_rsa_implementations_in_hardware_software↩
            ->setStyleClass("navigation_grid_item");
356     this->asymmetric_rsa_implementations_in_hardware_software->addWidget(↩
            std::move(↩
            button_cryptography_asymmetric_rsa_implementations_in_hardware_software↩
            ));
357     this->addWidget(std::move(↩
            asymmetric_rsa_implementations_in_hardware_software),22,2, Wt::↩
            AlignmentFlag::Center);
358
359     this->subtitle_dlp_cryptosystems = Wt::cpp14::make_unique<Wt::WText>(↩
            "Public-Key Cryptosystems based on the Discrete Logarithm Problem"↩
            );
360     this->subtitle_dlp_cryptosystems->setStyleClass("↩
            navigation_grid_subtitle");
361     this->addWidget(std::move(this->subtitle_dlp_cryptosystems),23,1);
362
363     this->asymmetric_dh_dhke = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
            WLink(
364             Wt::LinkType::InternalPath, "/asymmetric/dh-dhke"));
365     auto button_cryptography_asymmetric_dh_dhke = Wt::cpp14::make_unique<↩
            Wt::WPushButton>("Diffie-Hellman Key Exchange");
366     button_cryptography_asymmetric_dh_dhke->setStyleClass("↩
            navigation_grid_item");
367     this->asymmetric_dh_dhke->addWidget(std::move(↩
            button_cryptography_asymmetric_dh_dhke));
368     this->addWidget(std::move(asymmetric_dh_dhke),24,0, Wt::AlignmentFlag↩
            ::Center);
369
370     this->asymmetric_dh_algebra= Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
            WLink(
371             Wt::LinkType::InternalPath, "/asymmetric/dh-algebra"));
372     auto button_cryptography_asymmetric_dh_algebra = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Diffie-Hellman Algebra");
373     button_cryptography_asymmetric_dh_algebra->setStyleClass("↩
            navigation_grid_item");
374     this->asymmetric_dh_algebra->addWidget(std::move(↩
            button_cryptography_asymmetric_dh_algebra));
375     this->addWidget(std::move(asymmetric_dh_algebra),24,1, Wt::↩
            AlignmentFlag::Center);
376
377     this->asymmetric_dh_dlp = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
            WLink(
378             Wt::LinkType::InternalPath, "/asymmetric/dh-dlp"));
379     auto button_cryptography_asymmetric_dh_dlp = Wt::cpp14::make_unique<↩
            Wt::WPushButton>("The Discrete Logarithm Problem");
380     button_cryptography_asymmetric_dh_dlp->setStyleClass("↩
            navigation_grid_item");
381     this->asymmetric_dh_dlp->addWidget(std::move(↩
            button_cryptography_asymmetric_dh_dlp));
382     this->addWidget(std::move(asymmetric_dh_dlp),24,2, Wt::AlignmentFlag↩
            ::Center);
383
384     this->asymmetric_dh_security = Wt::cpp14::make_unique<Wt::WAnchor>(Wt↩
            ::WLink(
385             Wt::LinkType::InternalPath, "/asymmetric/dh-security"));
386     auto button_cryptography_asymmetric_dh_security = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Diffie-Hellman Security");
```

```
387     button_cryptography_asymmetric_dh_security->setStyleClass("↩
            navigation-grid-item");
388     this->asymmetric_dh_security->addWidget(std::move(↩
            button_cryptography_asymmetric_dh_security));
389     this->addWidget(std::move(asymmetric_dh_security),25,0, Wt::↩
            AlignmentFlag::Center);
390
391     this->asymmetric_dh_elgamal_encryption_scheme = Wt::cpp14::↩
            make_unique<Wt::WAnchor>(Wt::WLink(
392             Wt::LinkType::InternalPath, "/asymmetric/dh-dlp"));
393     auto button_cryptography_asymmetric_dh_encryption_scheme = Wt::cpp14↩
            ::make_unique<Wt::WPushButton>("The Elgamal Encryption Scheme");
394     button_cryptography_asymmetric_dh_encryption_scheme->setStyleClass("↩
            navigation-grid-item");
395     this->asymmetric_dh_elgamal_encryption_scheme->addWidget(std::move(↩
            button_cryptography_asymmetric_dh_encryption_scheme));
396     this->addWidget(std::move(asymmetric_dh_elgamal_encryption_scheme)↩
            ,25,1, Wt::AlignmentFlag::Center);
397
398     this->asymmetric_dh_elgamal_security = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
399             Wt::LinkType::InternalPath, "/asymmetric/dh-dlp"));
400     auto button_cryptography_asymmetric_dh_elgamal_security  = Wt::cpp14↩
            ::make_unique<Wt::WPushButton>("Elgamal Security");
401     button_cryptography_asymmetric_dh_elgamal_security->setStyleClass("↩
            navigation-grid-item");
402     this->asymmetric_dh_elgamal_security->addWidget(std::move(↩
            button_cryptography_asymmetric_dh_elgamal_security));
403     this->addWidget(std::move(asymmetric_dh_elgamal_security),25,2, Wt::↩
            AlignmentFlag::Center);
404
405     this->subtitle_ec_cryptosystems = Wt::cpp14::make_unique<Wt::WText>("↩
            Elliptic Curve Cryptosystems");
406     this->subtitle_ec_cryptosystems->setStyleClass("↩
            navigation-grid-subtitle");
407     this->addWidget(std::move(this->subtitle_ec_cryptosystems),26,1);
408
409     this->asymmetric_ec_definition = Wt::cpp14::make_unique<Wt::WAnchor>(↩
            Wt::WLink(
410             Wt::LinkType::InternalPath, "/asymmetric/ec-definition"));
411     auto button_cryptography_asymmetric_ec_definition  = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Elliptic Curve Definition");
412     button_cryptography_asymmetric_ec_definition->setStyleClass("↩
            navigation-grid-item");
413     this->asymmetric_ec_definition->addWidget(std::move(↩
            button_cryptography_asymmetric_ec_definition));
414     this->addWidget(std::move(asymmetric_ec_definition),27,0, Wt::↩
            AlignmentFlag::Center);
415
416     this->asymmetric_ec_group_operations = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
417             Wt::LinkType::InternalPath, "/asymmetric/ec-group-operations"↩
                ));
418     auto button_cryptography_asymmetric_ec_group_operations = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Elliptic Curve Group Operations");
419     button_cryptography_asymmetric_ec_group_operations->setStyleClass("↩
            navigation-grid-item");
420     this->asymmetric_ec_group_operations->addWidget(std::move(↩
```

```
        button_cryptography_asymmetric_ec_group_operations));
421     this->addWidget(std::move(asymmetric_ec_group_operations),27,1, Wt::↩
        AlignmentFlag::Center);
422
423     this->asymmetric_ec_dlp = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
        WLink(
424             Wt::LinkType::InternalPath, "/asymmetric/ec-dlp"));
425     auto button_cryptography_asymmetric_ec_dlp  = Wt::cpp14::make_unique<↩
        Wt::WPushButton>("Building a DLP with Elliptic Curves");
426     button_cryptography_asymmetric_ec_dlp->setStyleClass("↩
        navigation_grid_item");
427     this->asymmetric_ec_dlp->addWidget(std::move(↩
        button_cryptography_asymmetric_ec_dlp));
428     this->addWidget(std::move(asymmetric_ec_dlp),27,2, Wt::AlignmentFlag↩
        ::Center);
429
430     this->asymmetric_ec_dhke = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
        WLink(
431             Wt::LinkType::InternalPath, "/asymmetric/ec-dhke"));
432     auto button_cryptography_asymmetric_ec_dhke = Wt::cpp14::make_unique<↩
        Wt::WPushButton>("DH-Key Exchange with Elliptic Curves");
433     button_cryptography_asymmetric_ec_dhke->setStyleClass("↩
        navigation_grid_item");
434     this->asymmetric_ec_dhke->addWidget(std::move(↩
        button_cryptography_asymmetric_ec_dhke));
435     this->addWidget(std::move(asymmetric_ec_dhke),28,0, Wt::AlignmentFlag↩
        ::Center);
436
437     this->asymmetric_ec_security = Wt::cpp14::make_unique<Wt::WAnchor>(Wt↩
        ::WLink(
438             Wt::LinkType::InternalPath, "/asymmetric/ec-security"));
439     auto button_cryptography_asymmetric_ec_security = Wt::cpp14::↩
        make_unique<Wt::WPushButton>("Elliptic Curve Security");
440     button_cryptography_asymmetric_ec_security->setStyleClass("↩
        navigation_grid_item");
441     this->asymmetric_ec_security->addWidget(std::move(↩
        button_cryptography_asymmetric_ec_security));
442     this->addWidget(std::move(asymmetric_ec_security),28,1, Wt::↩
        AlignmentFlag::Center);
443
444     this->asymmetric_ec_implementations_in_hardware_software= Wt::cpp14::↩
        make_unique<Wt::WAnchor>(Wt::WLink(
445             Wt::LinkType::InternalPath, "/asymmetric/ec-implementations")↩
                ;
446     auto ↩
        button_cryptography_asymmetric_ec_implementations_in_hardware_software↩
          =
447             Wt::cpp14::make_unique<Wt::WPushButton>("Elliptic Curve ↩
                Implementations in Hardware and Software");
448     button_cryptography_asymmetric_ec_implementations_in_hardware_software↩
        ->setStyleClass("navigation_grid_item");
449     this->asymmetric_ec_implementations_in_hardware_software->addWidget(↩
        std::move(↩
        button_cryptography_asymmetric_ec_implementations_in_hardware_software↩
        ));
450     this->addWidget(std::move(↩
        asymmetric_ec_implementations_in_hardware_software),28,2, Wt::↩
        AlignmentFlag::Center);
```

```
451
452     this->subtitle_digital_signatures = Wt::cpp14::make_unique<Wt::WText>("Digital Signatures");
453     this->subtitle_digital_signatures->setStyleClass("navigation_grid_subtitle");
454     this->addWidget(std::move(this->subtitle_digital_signatures),29,1);
455
456     this->asymmetric_ds_introduction= Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
457             Wt::LinkType::InternalPath, "/asymmetric/ds-introduction"));
458     auto button_cryptography_asymmetric_ds_introduction = Wt::cpp14::make_unique<Wt::WPushButton>("Introduction to Digital Signatures");
459     button_cryptography_asymmetric_ds_introduction->setStyleClass("navigation_grid_item");
460     this->asymmetric_ds_introduction->addWidget(std::move(button_cryptography_asymmetric_ds_introduction));
461     this->addWidget(std::move(asymmetric_ds_introduction),30,0, Wt::AlignmentFlag::Center);
462
463     this->asymmetric_ds_rsa = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
464             Wt::LinkType::InternalPath, "/asymmetric/ds-rsa"));
465     auto button_cryptography_asymmetric_ds_rsa = Wt::cpp14::make_unique<Wt::WPushButton>("RSA Digital Signatures");
466     button_cryptography_asymmetric_ds_rsa->setStyleClass("navigation_grid_item");
467     this->asymmetric_ds_rsa->addWidget(std::move(button_cryptography_asymmetric_ds_rsa));
468     this->addWidget(std::move(asymmetric_ds_rsa),30,1, Wt::AlignmentFlag::Center);
469
470     this->asymmetric_ds_elgamal = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
471             Wt::LinkType::InternalPath, "/asymmetric/ds-elgamal"));
472     auto button_cryptography_asymmetric_ds_elgamal = Wt::cpp14::make_unique<Wt::WPushButton>("Elgamal Digital Signatures");
473     button_cryptography_asymmetric_ds_elgamal->setStyleClass("navigation_grid_item");
474     this->asymmetric_ds_elgamal->addWidget(std::move(button_cryptography_asymmetric_ds_elgamal));
475     this->addWidget(std::move(asymmetric_ds_elgamal),30,2, Wt::AlignmentFlag::Center);
476
477     this->asymmetric_ds_dsa = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
478             Wt::LinkType::InternalPath, "/asymmetric/ds-dsa"));
479     auto button_cryptography_asymmetric_ds_dsa = Wt::cpp14::make_unique<Wt::WPushButton>("Digital Signature Algorithm");
480     button_cryptography_asymmetric_ds_dsa->setStyleClass("navigation_grid_item");
481     this->asymmetric_ds_dsa->addWidget(std::move(button_cryptography_asymmetric_ds_dsa));
482     this->addWidget(std::move(asymmetric_ds_dsa),31,0, Wt::AlignmentFlag::Center);
483
484     this->asymmetric_ds_ecdsa = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::WLink(
```

```
485             Wt::LinkType::InternalPath, "/asymmetric/ec-security"));
486     auto button_cryptography_asymmetric_ds_ecdsa = Wt::cpp14::make_unique↩
            <Wt::WPushButton>("Elliptic Curve Digital Signatures");
487     button_cryptography_asymmetric_ds_ecdsa->setStyleClass("↩
            navigation-grid-item");
488     this->asymmetric_ds_ecdsa->addWidget(std::move(↩
            button_cryptography_asymmetric_ds_ecdsa));
489     this->addWidget(std::move(asymmetric_ds_ecdsa),31,2, Wt::↩
            AlignmentFlag::Center);
490
491 }
492
493 void crypto_online_navigation_grid::setup_applications_contents() {
494     this->title_applications = Wt::cpp14::make_unique<Wt::WText>("↩
            Protocols");
495     this->title_applications->setStyleClass("navigation-grid-title");
496     this->addWidget(std::move(this->title_applications),32,1);
497
498     this->subtitle_hash_functions = Wt::cpp14::make_unique<Wt::WText>("↩
            Hash Functions");
499     this->subtitle_hash_functions->setStyleClass("↩
            navigation-grid-subtitle");
500     this->addWidget(std::move(this->subtitle_hash_functions),33,1);
501
502     this->protocols_hf_motivation = Wt::cpp14::make_unique<Wt::WAnchor>(↩
            Wt::WLink(
503             Wt::LinkType::InternalPath, "/protocols/hash-function-↩
                motivation"));
504     auto button_cryptography_protocol_hf_motivation = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Why we need Hash Functions");
505     button_cryptography_protocol_hf_motivation->setStyleClass("↩
            navigation-grid-item");
506     this->protocols_hf_motivation->addWidget(std::move(↩
            button_cryptography_protocol_hf_motivation));
507     this->addWidget(std::move(this->protocols_hf_motivation),34,0, Wt::↩
            AlignmentFlag::Center);
508
509     this->protocols_hf_security_requirements = Wt::cpp14::make_unique<Wt↩
            ::WAnchor>(Wt::WLink(
510             Wt::LinkType::InternalPath, "/protocols/hash-function-↩
                security-requirements"));
511     auto button_cryptography_protocol_hf_security_requirements = Wt::↩
            cpp14::make_unique<Wt::WPushButton>("Security Requirements");
512     button_cryptography_protocol_hf_security_requirements->setStyleClass(↩
            "navigation-grid-item");
513     this->protocols_hf_security_requirements->addWidget(std::move(↩
            button_cryptography_protocol_hf_security_requirements));
514     this->addWidget(std::move(this->protocols_hf_security_requirements)↩
            ,34,1, Wt::AlignmentFlag::Center);
515
516     this->protocols_hf_overview = Wt::cpp14::make_unique<Wt::WAnchor>(Wt↩
            ::WLink(
517             Wt::LinkType::InternalPath, "/protocols/hash-function-↩
                overview"));
518     auto button_cryptography_protocol_hf_overview = Wt::cpp14::↩
            make_unique<Wt::WPushButton>("Hash Function Overview");
519     button_cryptography_protocol_hf_overview->setStyleClass("↩
            navigation-grid-item");
```

```
520     this->protocols_hf_overview->addWidget(std::move(↩
            button_cryptography_protocol_hf_overview));
521     this->addWidget(std::move(this->protocols_hf_overview),34,2, Wt::↩
            AlignmentFlag::Center);
522
523     this->protocols_hf_sha1 = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
            WLink(
524             Wt::LinkType::InternalPath, "/protocols/hash-function-sha1"))↩
                ;
525     auto button_cryptography_protocol_hf_sha1 = Wt::cpp14::make_unique<Wt↩
            ::WPushButton>("SHA-1");
526     button_cryptography_protocol_hf_sha1->setStyleClass("↩
            navigation_grid_item");
527     this->protocols_hf_sha1->addWidget(std::move(↩
            button_cryptography_protocol_hf_sha1));
528     this->addWidget(std::move(this->protocols_hf_sha1),35,0, Wt::↩
            AlignmentFlag::Center);
529
530     this->protocols_hf_sha3 = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
            WLink(
531             Wt::LinkType::InternalPath, "/protocols/hash-function-↩
                overview"));
532     auto button_cryptography_protocol_hf_sha3=  Wt::cpp14::make_unique<Wt↩
            ::WPushButton>("SHA-3");
533     button_cryptography_protocol_hf_sha3->setStyleClass("↩
            navigation_grid_item");
534     this->protocols_hf_sha3->addWidget(std::move(↩
            button_cryptography_protocol_hf_sha3));
535     this->addWidget(std::move(this->protocols_hf_sha3),35,1, Wt::↩
            AlignmentFlag::Center);
536
537     this->protocols_hf_bcrypt = Wt::cpp14::make_unique<Wt::WAnchor>(Wt::↩
            WLink(
538             Wt::LinkType::InternalPath, "/protocols/hash-function-↩
                overview"));
539     auto button_cryptography_protocol_hf_bcrypt = Wt::cpp14::make_unique<↩
            Wt::WPushButton>("bcrypt");
540     button_cryptography_protocol_hf_bcrypt->setStyleClass("↩
            navigation_grid_item");
541     this->protocols_hf_bcrypt->addWidget(std::move(↩
            button_cryptography_protocol_hf_bcrypt));
542     this->addWidget(std::move(this->protocols_hf_bcrypt),35,2, Wt::↩
            AlignmentFlag::Center);
543
544     this->subtitle_macs = Wt::cpp14::make_unique<Wt::WText>("Message ↩
            Authentication Codes");
545     this->subtitle_macs->setStyleClass("navigation_grid_subtitle");
546     this->addWidget(std::move(this->subtitle_macs),36,1);
547
548     this->protocols_macs_principles = Wt::cpp14::make_unique<Wt::WAnchor↩
            >(Wt::WLink(
549             Wt::LinkType::InternalPath, "/protocols/mac-principles"));
550     auto button_cryptography_protocol_mac_principles =
551             Wt::cpp14::make_unique<Wt::WPushButton>("Message ↩
                Authentication Code Principles");
552     button_cryptography_protocol_mac_principles->setStyleClass("↩
            navigation_grid_item");
553     this->protocols_macs_principles->addWidget(std::move(↩
```

```
        button_cryptography_protocol_mac_principles));
554     this->addWidget(std::move(this->protocols_macs_principles),37,0, Wt::↩
        AlignmentFlag::Center);
555
556     this->protocols_macs_hash_functions= Wt::cpp14::make_unique<Wt::↩
        WAnchor>(Wt::WLink(
557             Wt::LinkType::InternalPath, "/protocols/mac-hmac"));
558     auto button_cryptography_protocol_macs_hash_functions =
559             Wt::cpp14::make_unique<Wt::WPushButton>("Message ↩
                Authentication Codes From Hash Functions");
560     button_cryptography_protocol_macs_hash_functions->setStyleClass("↩
        navigation-grid-item");
561     this->protocols_macs_hash_functions->addWidget(std::move(↩
        button_cryptography_protocol_macs_hash_functions));
562     this->addWidget(std::move(this->protocols_macs_hash_functions),37,1, ↩
        Wt::AlignmentFlag::Center);
563
564     this->protocols_macs_block_cipher = Wt::cpp14::make_unique<Wt::↩
        WAnchor>(Wt::WLink(
565             Wt::LinkType::InternalPath, "/protocols/mac-cbcmac"));
566     auto button_cryptography_protocol_mac_block_cipher =
567             Wt::cpp14::make_unique<Wt::WPushButton>("Message ↩
                Authentication Codes From Block Ciphers");
568     button_cryptography_protocol_mac_block_cipher->setStyleClass("↩
        navigation-grid-item");
569     this->protocols_macs_block_cipher->addWidget(std::move(↩
        button_cryptography_protocol_mac_block_cipher));
570     this->addWidget(std::move(this->protocols_macs_block_cipher),37,2, Wt↩
        ::AlignmentFlag::Center);
571
572     this->subtitle_key_establishment = Wt::cpp14::make_unique<Wt::WText>(↩
        "Key Establishment");
573     this->subtitle_key_establishment->setStyleClass("↩
        navigation-grid-subtitle");
574     this->addWidget(std::move(this->subtitle_key_establishment),38,1);
575
576     this->protocols_key_est_introduction = Wt::cpp14::make_unique<Wt::↩
        WAnchor>(Wt::WLink(
577             Wt::LinkType::InternalPath, "/protocols/key-est-introduction"↩
                ));
578     auto button_cryptography_protocol_key_est_introduction =
579             Wt::cpp14::make_unique<Wt::WPushButton>("Introduction to Key ↩
                Establishment");
580     button_cryptography_protocol_key_est_introduction->setStyleClass("↩
        navigation-grid-item");
581     this->protocols_key_est_introduction->addWidget(std::move(↩
        button_cryptography_protocol_key_est_introduction));
582     this->addWidget(std::move(this->protocols_key_est_introduction),39,0,↩
         Wt::AlignmentFlag::Center);
583
584     this->protocols_key_est_symmetric = Wt::cpp14::make_unique<Wt::↩
        WAnchor>(Wt::WLink(
585             Wt::LinkType::InternalPath, "/protocols/key-est-symmetric"));
586     auto button_cryptography_protocol_key_est_symmetric =
587             Wt::cpp14::make_unique<Wt::WPushButton>("Key Establishment ↩
                with Symmetric Techniques");
588     button_cryptography_protocol_key_est_symmetric->setStyleClass("↩
        navigation-grid-item");
```

```
589     this->protocols_key_est_symmetric->addWidget(std::move(↩
            button_cryptography_protocol_key_est_symmetric));
590     this->addWidget(std::move(this->protocols_key_est_symmetric),39,1, Wt↩
            ::AlignmentFlag::Center);
591 
592     this->protocols_key_est_asymmetric = Wt::cpp14::make_unique<Wt::↩
            WAnchor>(Wt::WLink(
593             Wt::LinkType::InternalPath, "/protocols/key-est-asymmetric"))↩
                ;
594     auto button_cryptography_protocol_key_est_asymmetric =
595             Wt::cpp14::make_unique<Wt::WPushButton>("Key Establishment ↩
                with Asymmetric Techniques");
596     button_cryptography_protocol_key_est_asymmetric->setStyleClass("↩
            navigation_grid_item");
597     this->protocols_key_est_asymmetric->addWidget(std::move(↩
            button_cryptography_protocol_key_est_asymmetric));
598     this->addWidget(std::move(this->protocols_key_est_asymmetric),39,2, ↩
            Wt::AlignmentFlag::Center);
599 
600 }
601 
602 void crypto_online_navigation_grid::clear_grid() {
603 
604 }
```

Chapter 4

Testing

Chapter 5

Evaluation

List of Figures