1 Appendix A - Technical Solution Code

main.py

```
"""
main.py
=======

This is the file from which the entire program is run
Includes the entry point into the program
"""

import sys
from PyQt5 import QtCore, QtGui, QtWidgets
from program import MainMenu, create_database


def main():
    """ The main entry point to the program """
    app = QtWidgets.QApplication(sys.argv)
    create_database()
    application = MainMenu()
    sys.exit(app.exec_())


if __name__ == '__main__':
    main()
```

program/__init__.py

```
"""
__init__.py
===========
Imports from program for main.py
"""


from .main_section import MainMenu
from .user_interface import Ui_MainMenu
from .utils import create_database
```

program/main_section.py

```
"""
main_section.py
===============

Contains the class used to interact with the GUI for the
main menu in the project
```

```
Includes the Main Menu and progress screen
"""

from PyQt5 import QtWidgets
from PyQt5.QtWidgets import QTableWidget,QTableWidgetItem, QHeaderView
from .user_interface import Ui_MainMenu, Ui_ProgressScreen
from .login_section import Login
from .introduction_section import Introduction
from .investigation_section import GraphPlot
from .tutorial_section import Tutorial
from .summary_section import Summary
from .notes import TutorialNotes
from .utils import User, Screen, database_select, database_query
import sys

class MainMenu(Screen):

    """
    This class configures the GUI for the main menu, and allows the user
    to interact wih the GUI for this page
    """

    def __init__(self):
        super(MainMenu, self).__init__()
        self.ui = Ui_MainMenu()
        self.ui.setupUi(self)
        if User.GetSignedIn():
            self.ui.UsernameButton.setText(User.GetUsername())
            self.ui.UsernameButton.show()
        else:
            self.ui.UsernameButton.hide()
        self.ui.LogInButton.clicked.connect(self.goto_login)
        self.ui.TutorialButton.clicked.connect(self.goto_tutorial)
        self.ui.IntroductionButton.clicked.connect(self.goto_introduction)
        self.ui.InvestigationButton.clicked.connect(self.goto_investigation)
        self.ui.SummaryButton.clicked.connect(self.goto_summary)
        self.ui.ExitButton.clicked.connect(self.exit)
        self.ui.UsernameButton.clicked.connect(self.goto_progress)
        self.show()

    def goto_login(self):
        self.login = Login()
        self.hide()

    def goto_tutorial(self):
        self.tutorial = Tutorial()
        self.hide()

    def goto_introduction(self):
        self.introduction = Introduction()
        self.hide()

    def goto_investigation(self):
        self.investigation = GraphPlot()
        self.hide()

    def goto_summary(self):
        self.summary = Summary()
        self.hide()

    def goto_progress(self):
        self.progress = Progress()
        self.hide()

    def exit(self):
        sys.exit()


class Progress(Screen):

    """
    Progress
    """

    def __init__(self):
        super(Progress, self).__init__()
        self.ui = Ui_ProgressScreen()
        self.ui.setupUi(self)
        if User.GetSignedIn():
            self.ui.SubTitleText.setText(f'User: {User.GetUsername()}')
            self.setup_table()
        else:
            pass
        self.ui.BackButton.clicked.connect(self.goto_mainmenu)
        self.ui.NotesButton.clicked.connect(self.goto_notes)
        self.show()

    def goto_notes(self):
        self.notes = TutorialNotes()

    def setup_table(self):
        self.questions = database_select(['*'], ['Questions'])
        self.correct_answers = database_select(['*'], ['CorrectAnswers'])
        self.users_answers = database_query("SELECT Question_No,
            UsersAnswer FROM UsersAnswers WHERE Username=?",
            User.GetUsername())
        self.table_values = []
        for question_no, question in self.questions:
            if question_no == 0:
                continue
            answer = ''.join([str(items[1]) for items in
                self.users_answers if items[0] == question_no])
            correct_answers = [str(items[1]).lower() for items in
                self.correct_answers if items[0] == question_no]
            correct = answer in correct_answers
            self.table_values.append((question, answer, correct))
        self.ui.Table.setRowCount(len(self.table_values))
        for i, values in enumerate(self.table_values):
            for j, value in enumerate(values):
                self.ui.Table.setItem(i,j, QTableWidgetItem(str(value)))
        header = self.ui.Table.horizontalHeader()
        header.setSectionResizeMode(0, QtWidgets.QHeaderView.Stretch)
        header.setSectionResizeMode(1,
            QtWidgets.QHeaderView.ResizeToContents)
        header.setSectionResizeMode(2,
            QtWidgets.QHeaderView.ResizeToContents)
```

program/login_section.py

```
"""
login_section.py
================

Contains all of the classes used to interact with the GUI for the
Login section of the project

Includes the Login, Sign Up, Forgotten Password, and Reset Password
    Screens
"""


import re
import random
from PyQt5 import QtCore, QtGui, QtWidgets
from .user_interface import Ui_LoginScreen, Ui_SignUpScreen,
    Ui_ForgottenPasswordScreen, Ui_ForgottenPassword2Screen,
    Ui_ResetPasswordScreen, Ui_ResetPassword2Screen
from .utils import database_insert, database_select, database_query,
    database_print, hash_password, check_password,
    send_verification_email, User, Screen


class LoginSection(Screen):

    """
    A class inherited by all of the Screens/Page classes in the login
        section
    of the program
```

```
The functions defined in this class allow for different pages to be
    loaded
and hidden, so that the user is able to navigate to different parts
    of the
login section using the GUI

It also contains some functions which are commonly used in many of
    the
Classes that inherit this class
"""

def __init__(self):
    super(LoginSection, self).__init__()
    self.show_or_hide = 'Show'
    self.show_or_hide_2 = 'Show'

def setup_tabs(self):
    try:
        self.ui.BackButton.clicked.connect(self.goto_mainmenu)
    except AttributeError:
        pass
    try:
        self.ui.LoginTab.clicked.connect(self.goto_login)
    except AttributeError:
        pass
    try:
        self.ui.SignUpTab.clicked.connect(self.goto_signup)
    except AttributeError:
        pass
    try:
        self.ui.ForgottenPasswordTab.clicked.connect(self.goto_forgotten_password)
    except AttributeError:
        pass
    try:
        self.ui.ResetPasswordTab.clicked.connect(self.goto_reset_password)
    except AttributeError:
        pass
    try:
        self.ui.ShowHideButton.clicked.connect(self.show_hide)
    except AttributeError:
        pass
    try:
        self.ui.ShowHideButton_2.clicked.connect(self.show_hide_2)
    except AttributeError:
        pass
    try:
        self.ui.SubmitButton.clicked.connect(self.submit)
    except AttributeError:
        pass
```

```
def goto_login(self):
    self.login = Login()
    self.hide()

def goto_signup(self):
    self.signup = SignUp()
    self.hide()

def goto_forgotten_password(self):
    self.forgotten_password = ForgottenPassword()
    self.hide()

def goto_reset_password(self):
    self.reset_password = ResetPassword()
    self.hide()

def show_hide(self):
    if self.show_or_hide == ’Show’:
        self.ui.PasswordInput.setEchoMode(QtWidgets.QLineEdit.Normal)
        self.show_or_hide = ’Hide’
    else:
        self.ui.PasswordInput.setEchoMode(QtWidgets.QLineEdit.Password)
        self.show_or_hide = ’Show’
    self.ui.ShowHideButton.setText(self.show_or_hide)

def show_hide_2(self):
    if self.show_or_hide_2 == ’Show’:
        self.ui.PasswordInput_2.setEchoMode(QtWidgets.QLineEdit.Normal)
        self.show_or_hide_2 = ’Hide’
    else:
        self.ui.PasswordInput_2.setEchoMode(QtWidgets.QLineEdit.Password)
        self.show_or_hide_2 = ’Show’
    self.ui.ShowHideButton_2.setText(self.show_or_hide_2)

def login(self):
    """Used as default login behaviour"""
    from .main_section import MainMenu
    self.username = self.ui.UsernameInput.text()
    self.password = self.ui.PasswordInput.text()
    try:
        self.correct_hashed_password = database_query("SELECT
            Password FROM Users WHERE Username=?",
            self.username)[0][0]
    except IndexError:
        self.ui.ErrorLabel.setText("Username or password is not
            valid")
    else:
        if not check_password(self.password,
            self.correct_hashed_password):
                self.ui.ErrorLabel.setText("Username or password is not
                    valid")
            else:
                User.SetSignedIn(True)
                User.SetUsername(self.username)
                self.email = database_query("SELECT Email FROM Users
                    WHERE Username=?", self.username)[0][0]
                User.SetEmail(self.email)
                self.main_menu = MainMenu()
                self.hide()

    def are_invalid_passwords(self):

        """
        Inputs: self.password1: string, self.password2: string
        Outputs: strings or bool

        Checks to see if the input passwords are invalid

        If both of the input password are the same and meet the
            following criteria:
            - At least one uppercase letter
            - At least one lowercase letter
            - At least one digit
            - At least 8 characters long
        Then the bool value False is Output

        Otherwise, the passwords do not meet the criteria, so a string
            is output
        describing why they do not meet the criteria. This string has
            the bool
        value True
        """

        if not re.fullmatch('(?=.*?[a-z])(?=.*?[A-Z])(?=.*?[0-9]).{8,}',
            self.password1):
            return "Password must contain lower case, upper case,\na
                number, and be at least 8 characters long"
        elif self.password1 != self.password2:
            return "Passwords do not match"
        else:
            return False


class ResetPassword2(LoginSection):

    """
    Displays the Second Screen in the Reset Password part of the Login
        Section
    This is the screen where the user is actually able to permenantly
        change
    their password
    """

    def __init__(self):
        super(ResetPassword2, self).__init__()
        self.ui = Ui_ResetPassword2Screen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.show()

    def submit(self):
        self.password1 = self.ui.PasswordInput.text()
        self.password2 = self.ui.ConfirmPasswordInput.text()
        self.pwds_invalid = self.are_invalid_passwords()
        if self.pwds_invalid:
            self.ui.ErrorLabel.setText(self.pwds_invalid)
        else:
            database_query("UPDATE Users SET Password=? WHERE
                Username=?", hash_password(self.password1),
                User.GetUsername())
            from .main_section import MainMenu
            self.main_menu = MainMenu()
            self.hide()


class ResetPassword(LoginSection):

    """
    Displays the first screen in the Reset Password part of the Login
        Section

    On this screen, the user is required to login to confirm that the
        user is
    actually the person who owns the account
    """

    def __init__(self):
        super(ResetPassword, self).__init__()
        self.ui = Ui_ResetPasswordScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.show()

    def submit(self):
        from .main_section import MainMenu
        if self.login():
            self.reset_password_2 = ResetPassword2()
            self.hide()
```

```
class ForgottenPassword2(LoginSection):

    """
    Displays the second screen in the Forgotten Password part of the
        Login Section

    This is the screen where the user enter's the verification code that
        they
    have been emailed to confirm that they are the owners of that account

    The user will then be immediately taken to the ResetPassword screen,
        once
    they submit the correct verification code
    """

    def __init__(self, verification_code):
        super(ForgottenPassword2, self).__init__()
        self.verification_code = verification_code
        self.ui = Ui_ForgottenPassword2Screen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.show()

    def submit(self):
        # Check verification code
        self.user_input = self.ui.VerificationCodeInput.text()
        if self.user_input == self.verification_code:
            User.SetSignedIn(True)
            self.reset_password_2 = ResetPassword2()
            self.hide()
        else:
            self.ui.ErrorLabel.setText("Verification Code Incorrect")


class ForgottenPassword(LoginSection):

    """
    Displays the first screen in the Forgotten Password part of the
        Login Section

    This is the screen where the user is asked to enter the email
        associated
    with their account. An email is then sent to the user containing a
    6 digit pseudorandom security code that they will tehn have to enter
        on the
    following screen
    """
    def __init__(self):
        super(ForgottenPassword, self).__init__()
        self.ui = Ui_ForgottenPasswordScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.show()

    def submit(self):
        self.email = self.ui.EmailInput.text()
        self.selection = database_query("SELECT Email FROM Users WHERE
             Email=?", (self.email,))
        if len(self.selection) == 0:
            self.ui.ErrorLabel.setText("Email is not registered")
        else:
            # Send Email
            self.verificaton_code = ’’.join(list(map(str,
                 [random.randint(0, 9) for _ in range(6)])))
            self.username = database_query("SELECT Username FROM Users
                 WHERE Email=?", self.email)[0][0]
            User.SetUsername(self.username)
            User.SetEmail(self.email)
            send_verification_email(self.verificaton_code)
            self.forgotten_password_2 =
                 ForgottenPassword2(self.verificaton_code)
            self.hide()


class SignUp(LoginSection):

    """
    Displays Sign Up screen as part of the Login Section

    This is the screen where the user is able to create an account
    They enter their username, email, and password (twice). Given that
         this data
    is all valid, the data is stored to the database, and the user’s
         account
    has been permenantly created.
    """

    def __init__(self):
        super(SignUp, self).__init__()
        self.ui = Ui_SignUpScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.show()

    def submit(self):
        from .main_section import MainMenu
        self.username = self.ui.UsernameInput.text()
        self.email = self.ui.EmailInput.text()
        self.password1 = self.ui.PasswordInput.text()
        self.password2 = self.ui.PasswordInput_2.text()
        self.pwds_invalid = self.are_invalid_passwords()
        # check is of right form
        if not re.fullmatch('\w{1,20}', self.username):
            self.ui.ErrorLabel.setText("Username must be at 1-20
                 characters long\nand not contain special characters")
        elif not re.fullmatch('.+@.+\..+', self.email):
            self.ui.ErrorLabel.setText("Email address must be valid")
        elif self.pwds_invalid:
            self.ui.ErrorLabel.setText(self.pwds_invalid)
        else:
            self.username_query = database_select(['Username'], ['Users'])
            self.usernames = set([row[0].lower() for row in
                 self.username_query])
            if self.username.lower() in self.usernames:
                self.ui.ErrorLabel.setText("Username already taken")
            else:
                self.emaill_query = database_select(['Email'], ['Users'])
                self.emails = set([row[0].lower() for row in
                     self.emaill_query])
                if self.email.lower() in self.emails:
                    self.ui.ErrorLabel.setText("Email already taken")
                else:
                    self.hashed_password = hash_password(self.password1)
                    database_insert('Users', self.username, self.email,
                         self.hashed_password)
                    User.SetSignedIn(True)
                    User.SetUsername(self.username)
                    User.SetEmail(self.email)
                    self.main_menu = MainMenu()
                    self.hide()


class Login(LoginSection):

    """
    Displays Login screen as part of the Login Section
    This screen is the main entry point to the login section

    The user can use this screen to sign in to an account that they have
    previously created
    """

    def __init__(self):
        super(Login, self).__init__()
        self.ui = Ui_LoginScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.show()

    def submit(self):
        from .main_section import MainMenu
        if self.login():
            self.main_menu = MainMenu()
            self.hide()
```

program/tutorial_section.py

```
"""
tutorial_section.py
===================

Contains all of the classes used to interact with the GUI for the
tutorial section of the project

Includes the Program Structure, Login, Introduction, Investigation
and Summary Tutorial Screens
"""

from PyQt5 import QtWidgets, QtCore
from .user_interface import Ui_TutorialScreen,
     Ui_ProgramStructureTutorialScreen, Ui_IntroductionTutorialScreen,
     Ui_InvestigationTutorialScreen, Ui_LoginTutorialScreen,
     Ui_SummaryTutorialScreen
from .utils import User, Screen, StaticGraphScreen, database_query
from .notes import TutorialNotes


class TutorialSection(Screen):

    """
    A class inherited by all of the Screens/Page classes in the tutorial
         section
    of the program

    The functions defined in this class allow for different pages to be
         loaded
    and hidden, so that the user is able to navigate to different parts
         of the
    program using the GUI
    """

    def __init__(self):
        super(TutorialSection, self).__init__()

    def setup_tabs(self):
```

```
    """
    Allows the tabs and buttons to run a function once clicked, if
        they
    exists on the web page that the tab/button was clicked on
    """

    try:
        self.ui.TutorialTab.clicked.connect(self.goto_tutorial)
    except AttributeError:
        pass
    try:
        self.ui.ProgramStructureTab.clicked.connect(self.goto_program_structure)
    except AttributeError:
        pass
    try:
        self.ui.LoginTab.clicked.connect(self.goto_login)
    except AttributeError:
        pass
    try:
        self.ui.IntroductionTab.clicked.connect(self.goto_introduction)
    except AttributeError:
        pass
    try:
        self.ui.InvestigationTab.clicked.connect(self.goto_investigation)
    except AttributeError:
        pass
    try:
        self.ui.SummaryTab.clicked.connect(self.goto_summary)
    except AttributeError:
        pass

"""
The goto functions are run when a tab is clicked. They load a new
    page,
and hide the old page.
"""

def goto_tutorial(self):
    self.tutorial= Tutorial()
    self.hide()

def goto_program_structure(self):
    self.program_structure = ProgramStructure()
    self.hide()

def goto_login(self):
    self.login = LoginTutorial()
    self.hide()

def goto_introduction(self):
        self.introduction = IntroductionTutorial()
        self.hide()

    def goto_investigation(self):
        self.investigation = InvestigationTutorial()
        self.hide()

    def goto_summary(self):
        self.summary = SummaryTutorial()
        self.hide()

    def goto_tutorial_notes(self):
        self.tutorial_notes = TutorialNotes()


class Tutorial(TutorialSection):

    """
    The Tutorial Screen is the main entry point to the tutorial section
        of the
    program

    This class displays said screen to the user
    """

    def __init__(self):
        super(Tutorial, self).__init__()
        self.ui = Ui_TutorialScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_mainmenu)
        self.ui.NextButton.clicked.connect(self.goto_program_structure)
        self.show()


class ProgramStructure(TutorialSection):

    """
    This class displays the second screen in the tutorial section: the
        program
    structure tutorial
    """

    def __init__(self):
        super(ProgramStructure, self).__init__()
        self.ui = Ui_ProgramStructureTutorialScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_tutorial)
        self.ui.NextButton.clicked.connect(self.goto_login)
        self.show()


class LoginTutorial(TutorialSection):

    """
    This class displays the third screen in the tutorial section: the
    login tutorial
    """

    def __init__(self):
        super(LoginTutorial, self).__init__()
        self.ui = Ui_LoginTutorialScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_program_structure)
        self.ui.NextButton.clicked.connect(self.goto_introduction)
        self.show()


class IntroductionTutorial(TutorialSection):

    """
    This class displays the fourth screen in the tutorial section: the
    introduction tutorial
    """

    def __init__(self):
        super(IntroductionTutorial, self).__init__()
        self.question_no = 1
        self.ui = Ui_IntroductionTutorialScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_login)
        self.ui.NextButton.clicked.connect(self.goto_investigation)
        self.show()


class InvestigationTutorial(TutorialSection):

    """
    This class displays the fifth screen in the tutorial section: the
    investigation tutorial
    """

    def __init__(self):
        super(InvestigationTutorial, self).__init__()
        self.gradient = 0
        self.y_intercept = 0
        self.question_no = 2
        self.ui = Ui_InvestigationTutorialScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_introduction)
        self.ui.NextButton.clicked.connect(self.goto_summary)
        self.ui.MSlider.valueChanged.connect(self.M_value_changed)
        self.ui.CSlider.valueChanged.connect(self.C_value_changed)
        self.ui.GraphButton.clicked.connect(self.graph)
        self.show()

    def M_value_changed(self):
        """Update the gradient display when the slider is changed"""
        self.gradient = self.ui.MSlider.value()
        self.ui.MDisplay.setText(self.center_text(str(self.gradient)))

    def C_value_changed(self):
        """Update the y-intercept display when the slider is changed"""
        self.y_intercept = self.ui.CSlider.value()
        self.ui.CDisplay.setText(self.center_text(str(self.y_intercept)))

    def graph(self):
        """Display a new screen with the graph on it"""
        self.plot = GraphMatPlot(self.gradient, self.y_intercept)


class GraphMatPlot(StaticGraphScreen):

    """
    The class will display the graph in the the investigation tutorial
        section
    It will graph the function y=mx+c where m and c are given by the user
    """

    def __init__(self, gradient, y_intercept):
        super(GraphMatPlot, self).__init__()
        self.y_intercept = y_intercept
        self.gradient = gradient
        self.x_vals = list(range(6))
        self.y_vals = [self.gradient * num + self.y_intercept for num in
            self.x_vals]
        self.show()
        self.graph()

    def get_label(self, value, gradient=False):

        """
        Format the y-intercept and gradient properly so that it looks
            correct
        when printed out in the graph legend.
        """

        if gradient:
            match value:
                case -1:
                    return '-x'
                case 0:
                    return ''
                case 1:
                    return 'x'
                case _:
                    return f'{value}x'
        elif value == 0:
            if self.gradient == 0:
                return '0'
            else:
                return ''
        elif value > 0 and self.gradient != 0:
            return f'+{value}'
        else:
            return str(value)

    def graph(self):
        """Plot the graph of the function y=mx+c"""
        self.gradient_label = self.get_label(self.gradient, True)
        self.intercept_label = self.get_label(self.y_intercept, False)
        self.matplotlibwidget.axes.plot(
                self.x_vals, self.y_vals,
                label=f'y={self.gradient_label}{self.intercept_label}',
                color='blue')
        self.matplotlibwidget.axes.grid()
        self.matplotlibwidget.axes.legend(loc='upper left')
        self.matplotlibwidget.canvas.draw()


class SummaryTutorial(TutorialSection):

    """
    This class displays the sixth and last screen in the tutorial
        section: the
    summary tutorial
    """

    def __init__(self):
        super(SummaryTutorial, self).__init__()
        self.ui = Ui_SummaryTutorialScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_investigation)
        self.ui.NextButton.clicked.connect(self.goto_mainmenu)
        self.ui.NotesButton.clicked.connect(self.goto_tutorial_notes)
        self.show()
```

program/tutorial_section.py

```
"""
introduction_section.py
=======================

Contains all of the classes used to interact with the GUI for the
introduction section of the project

Includes the classes:
    - IntroductionSection
    - Introduction
    - HistoricalBackground
    - WhatIsTheRiemannHypothesis
    - PracticalApplications
"""

from PyQt5 import QtWidgets
from .user_interface import Ui_IntroductionScreen,
    Ui_HistoricalBackgroundScreen, Ui_WhatIsTheRiemannHypothesisScreen,
    Ui_PracticalApplicationsScreen
from .utils import User, Screen
from .notes import IntroductionNotes


class IntroductionSection(Screen):

    """
    A class inherited by all of the Screens/Page classes in the
        introduction
    section of the program

    The functions defined in this class allow for different pages to be
        loaded
    and hidden, so that the user is able to navigate to different parts
        of the
    program using the GUI
    """

    def __init__(self):
        super(IntroductionSection, self).__init__()

    def setup_tabs(self):
        try:
            self.ui.IntroductionTab.clicked.connect(self.goto_introduction)
        except AttributeError:
            pass
        try:
            self.ui.HistoricalBackgroundTab.clicked.connect(self.goto_historical_background)
        except AttributeError:
            pass
        try:
            self.ui.WhatIsTheRHTab.clicked.connect(self.goto_what_is_the_riemann_hypothesis)
        except AttributeError:
            pass
        try:
            self.ui.PracticalApplicationsTab.clicked.connect(self.goto_practical_applications)
        except AttributeError:
            pass
        try:
            self.ui.NotesButton.clicked.connect(self.goto_introduction_notes)
        except AttributeError:
            pass

    def goto_introduction(self):
        self.introduction = Introduction()
        self.hide()

    def goto_historical_background(self):
        self.historical_background = HistoricalBackground()
        self.hide()

    def goto_what_is_the_riemann_hypothesis(self):
        self.what_is_the_rh = WhatIsTheRiemannHypothesis()
        self.hide()

    def goto_practical_applications(self):
        self.practical_applications = PracticalApplications()
        self.hide()

    def goto_introduction_notes(self):
        self.introduction_notes = IntroductionNotes()


class Introduction(IntroductionSection):

    """
    The Introduction Screen is the main entry point to the introduction
        section of the
    program
    """

    def __init__(self):
        super(Introduction, self).__init__()
        self.ui = Ui_IntroductionScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_mainmenu)
        self.ui.NextButton.clicked.connect(self.goto_historical_background)
        self.show()


class HistoricalBackground(IntroductionSection):

    """
    A class used to interact with the Historical Background GUI screen
    """

    def __init__(self):
        super(HistoricalBackground, self).__init__()
        self.ui = Ui_HistoricalBackgroundScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_introduction)
        self.ui.NextButton.clicked.connect(self.goto_what_is_the_riemann_hypothesis)
        self.show()


class WhatIsTheRiemannHypothesis(IntroductionSection):

    """
    A class used to interact with the What Is The Riemann Hypothesis GUI
        screen
    """

    def __init__(self):
        super(WhatIsTheRiemannHypothesis, self).__init__()
        self.question_no = 3
        self.ui = Ui_WhatIsTheRiemannHypothesisScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_historical_background)
        self.ui.NextButton.clicked.connect(self.goto_practical_applications)
        self.show()


class PracticalApplications(IntroductionSection):

    """
    Practical Applications
    """

    def __init__(self):
        super(PracticalApplications, self).__init__()
        self.question_no = 4
        self.ui = Ui_PracticalApplicationsScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_what_is_the_riemann_hypothesis)
        self.ui.NextButton.clicked.connect(self.goto_mainmenu)
        self.show()
```

program/investigation_section.py

```
"""
investigation_section.py
========================

Contains all of the classes used to interact with the GUI for the
investigation section of the project

Includes the screens for the investigation section
"""

import sys
import matplotlib
import numpy as np
from matplotlib.figure import Figure
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QTableWidget,QTableWidgetItem, QHeaderView
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as
    FigureCanvas
from .notes import InvestigationNotes
from .user_interface import Ui_PolarGraphScreen,
    Ui_PrimeCountingFunctionScreen, Ui_GraphPlotsScreen,
    Ui_ZetaZeroesPlotScreen, Ui_PrimeNumbersScreen,
    Ui_CalculatorScreen, Ui_SingleCalculatorScreen,
    Ui_TableCalculatorScreen, Ui_TableCalculator2Screen,
    Ui_CalculateZeroesScreen, Ui_CalculateZeroes2Screen,
    Ui_CalculatorLeaderboardScreen, Ui_MatPlotScreen, Ui_ZeroesScreen,
    Ui_ZetaApproximationScreen
from .utils import zeta, sieve_of_eratosthenes, prime_power_function,
    prime_counting_function_estimation, logarithmic_integral,
    binary_insertion_sort, save_zeta_zeroes_to_file,
    save_zeta_values_to_file, make_int, make_complex, is_zeta_zero,
    Screen, User, database_query, database_insert, database_select,
    get_id, database_print, DynamicGraphScreen, Complex


class InvestigationSection(Screen):

    """
```

```
A class inherited by all of the Screens/Page classes in the
    investigation section
of the program

The functions defined in this class allow for different pages to be
    loaded
and hidden, so that the user is able to navigate to different parts
    of the
program using the GUI
"""

def __init__(self):
    super(InvestigationSection, self).__init__()

def setup_tabs(self):
    try:
        self.ui.NotesButton.clicked.connect(self.goto_investigation_notes)
    except AttributeError:
        pass
    try:
        self.ui.ZetaZeroesPlotTab.clicked.connect(self.goto_zeta_zeroes_plot)
    except AttributeError:
        pass
    try:
        self.ui.PrimeTab.clicked.connect(self.goto_prime)
    except AttributeError:
        pass
    try:
        self.ui.PrimesTab.clicked.connect(self.goto_primes)
    except AttributeError:
        pass
    try:
        self.ui.CalculatorTab.clicked.connect(self.goto_calculator)
    except AttributeError:
        pass
    try:
        self.ui.ZeroesTab.clicked.connect(self.goto_zeroes)
    except AttributeError:
        pass
    try:
        self.ui.PolarTab.clicked.connect(self.goto_polar)
    except AttributeError:
        pass
    try:
        self.ui.GraphsTab.clicked.connect(self.goto_graph_plots)
    except AttributeError:
        pass
    try:
        self.ui.TableTab.clicked.connect(self.goto_table_calculator)
    except AttributeError:
        pass
    try:
        self.ui.LeaderboardTab.clicked.connect(self.goto_calculator_leaderboard)
    except AttributeError:
        pass
    try:
        self.ui.SingleTab.clicked.connect(self.goto_single)
    except AttributeError:
        pass
    try:
        self.ui.ZetaApproximationTab.clicked.connect(self.goto_zeta_approximation)
    except AttributeError:
        pass

def goto_polar(self):
    self.polar = PolarGraph()
    self.hide()

def goto_zeroes(self):
    self.zeroes = Zeroes()
    self.hide()

def goto_prime(self):
    self.prime = PrimeCountingFunction()
    self.hide()

def goto_zeta_zeroes_graph(self):
    self.zeta_zeroes_graph = ZetaZeroesMatPlot()

def goto_pcf_graph(self):
    self.graph = PrimeCountingFunctionMatPlot()

def goto_graph_plots(self):
    self.polar = GraphPlot()
    self.hide()

def goto_primes(self):
    self.primes = PrimeNumbers()
    self.hide()

def goto_calculator(self):
    self.calculator = Calculator()
    self.hide()

def goto_zeta_zeroes_plot(self):
    self.zeroes_plot = ZetaZeroesPlot()
    self.hide()

def goto_calculate_zeroes(self):
    self.zeroes = CalculateZeroes()
    self.hide()

def goto_calculate_zeroes_2(self):
    self.ui.ErrorLabel.setText(self.center_text(f’Calculating...’))
    self.calculate_zeroes()
    if self.zeroes_calculated:
        self.calculate_zeroes_2 = CalculateZeroes2(self.zeroes)
        self.hide()

def goto_single(self):
    self.single = SingleCalculator()
    self.hide()

def goto_table_calculator(self):
    self.tale_calculator = TableCalculator()
    self.hide()

def goto_table_calculator_2(self):
    self.calculate_zeta()
    if self.table_values:
        self.table_calculator_2 = TableCalculator2(self.table_values)
        self.hide()

def goto_calculator_leaderboard(self):
    self.calculator_leaderboard = CalculatorLeaderboard()
    self.hide()

def goto_investigation_notes(self):
    self.investigation_notes = InvestigationNotes()

def goto_zeta_approximation(self):
    self.zeta_approximation = ZetaApproximation()
    self.hide()

def validate_input(self, complex_user_input, split_char):
    split_input = complex_user_input.split(split_char)
    assert 0 < len(split_input) <= 2
    if len(split_input) == 2:
        assert complex_user_input[-1] in [’i’, ’j’]
        split_input[1] = split_input[1][:-1]
    return Complex(*split_input)

def get_valid_complex_input(self, complex_user_input):
    is_real_negative = False
    try:
        complex_input = self.validate_input(complex_user_input, ’+’)
    except (AssertionError, ValueError):
        try:
            if complex_user_input[0] == ’-’:
                complex_user_input = complex_user_input[1:]
```

```
                    is_real_negative = True
                complex_input = self.validate_input(complex_user_input,
                     '-')
                if is_real_negative:
                    complex_input = Complex((-1)*complex_input.get_real(),
                         (-1)*complex_input.get_imag())
                else:
                    complex_input = Complex(complex_input.get_real(),
                         (-1)*complex_input.get_imag())
            except (AssertionError, ValueError):
                complex_input = None
        return complex_input


class CalculateZeroes2(InvestigationSection):

    """
    The CalculateZeroes2 class is used to display the zeta zeroes that
         have
    been calculated by the user

    The user then has the option to save these to the database or to a
         file
    """

    def __init__(self, zeroes):
        super(CalculateZeroes2, self).__init__()
        self.zeroes = zeroes
        self.ui = Ui_CalculateZeroes2Screen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_calculate_zeroes)
        self.ui.NextButton.clicked.connect(self.goto_zeroes)
        self.ui.DatabaseButton.clicked.connect(self.saveto_database)
        self.ui.FileButton.clicked.connect(self.saveto_file)
        self.ui.ZetaTable.setRowCount(len(self.zeroes))
        for i, values in enumerate(self.zeroes):
            for j in range(len(values)):
                self.ui.ZetaTable.setItem(i,j,
                     QTableWidgetItem(str(values[j])))
        self.ui.ZetaTable.horizontalHeader().setStretchLastSection(True)
        self.ui.ZetaTable.horizontalHeader().setSectionResizeMode(
            QHeaderView.Stretch)
        self.ui.ZetaTable.setColumnWidth(1, 100)
        self.show()

    def saveto_database(self):
        if User.GetUsername():
            database_inputs = database_select(['Zero_Real_Input',
                 'Zero_Imag_Input'], ['Zeroes'])
            for real, imag in self.zeroes:
                # if there is a row with the same real and imag
                in_table = False
                for ri, ii in database_inputs:
                    if ri == real and ii == imag:
                        in_table = True
                # only add to database if not already in database
                if not in_table:
                    self.Zeta_Zero_ID = get_id('Zero_ID', 'Zeroes')
                    database_insert('Zeroes', self.Zeta_Zero_ID, real,
                        imag)
                    database_insert('UserZeroes', self.Zeta_Zero_ID,
                        User.GetUsername())
            self.ui.ErrorLabel.setText(self.center_text('Zeroes saved to
                database'))
        else:
            self.ui.ErrorLabel.setText(self.center_text(f'You must be
                signed in to be able to '
                    'save to the database'))

    def saveto_file(self):
        filepath = 'files/zeta_zeroes.csv'
        fieldnames = ['InputReal', 'InputImag']
        save_zeta_zeroes_to_file(self.zeroes, filepath,
            fieldnames=fieldnames)
        self.ui.ErrorLabel.setText(self.center_text(f'Table contents
            written to {filepath}'))


class CalculateZeroes(InvestigationSection):

    """
    The CalculateZeroes class is used to ask for input from the user as
        to how
    many zeta zeroes they want to calculate. It then calculates these
        values
    and displays them in the CalculateZeroes2 screen
    """

    def __init__(self):
        super(CalculateZeroes, self).__init__()
        self.ui = Ui_CalculateZeroesScreen()
        self.zeroes = []
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_zeroes)
        self.ui.NextButton.clicked.connect(self.goto_zeroes)
        self.ui.CalculateButton.clicked.connect(self.goto_calculate_zeroes_2)
        self.show()
    def calculate_zeroes(self):
        self.no_of_zeroes_input = self.ui.NoOfZeroesInput.text()
        self.no_of_zeroes = make_int(self.no_of_zeroes_input)
        self.zeroes_calculated = False
        if self.no_of_zeroes:
            if 0 < self.no_of_zeroes <= 100:
                self.zeroes_calculated = True
                self.zeroes = []
                count = 0
                while len(self.zeroes) < self.no_of_zeroes:
                    accuracy = count // 500 + 100
                    real = 1/2
                    imag = count / accuracy
                    if is_zeta_zero(real, imag) and (real, round(imag, 1))
                         not in self.zeroes:
                        self.zeroes.append((real, round(imag, 1)))
                    count += 1
            else:
                self.ui.ErrorLabel.setText(self.center_text('No. of
                    Zeroes must be between 1 and 100'))
        else:
                self.ui.ErrorLabel.setText(self.center_text('No. Of
                    Zeroes must be a positive integer between 1 and 100'))


class Zeroes(InvestigationSection):

    """
    """

    def __init__(self):
        super(Zeroes, self).__init__()
        self.question_no = 5
        self.ui = Ui_ZeroesScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_calculator)
        self.ui.NextButton.clicked.connect(self.goto_mainmenu)
        self.ui.CalculateButton.clicked.connect(self.goto_calculate_zeroes)
        self.show()


class CalculatorLeaderboard(InvestigationSection):

    """
    The CalculatorLeaderboard class is used to display how many zeta
        values
    been calculated by each user
    """
    def __init__(self):
        super(CalculatorLeaderboard, self).__init__()
        self.ui = Ui_CalculatorLeaderboardScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.get_rows()
        self.sort_rows()
        self.ui.PrevButton.clicked.connect(self.goto_table_calculator)
        self.ui.NextButton.clicked.connect(self.goto_zeroes)
        self.ui.ZetaTable.setRowCount(len(self.sorted_rows))
        for i, row in enumerate(self.sorted_rows):
            for j in range(len(row)):
                self.ui.ZetaTable.setItem(i,j,
                     QTableWidgetItem(str(row[j])))
        self.ui.ZetaTable.horizontalHeader().setStretchLastSection(True)
        self.ui.ZetaTable.horizontalHeader().setSectionResizeMode(
            QHeaderView.Stretch)
        self.ui.ZetaTable.setColumnWidth(1, 100)
        self.show()

    def get_rows(self):
        self.rows = []
        self.usernames = [username[0] for username in
             database_select(['Username'], ['Users'])]
        for username in self.usernames:
            number_of_zeta_values_calculated = len(database_query(
                    'SELECT * FROM UserZeta WHERE Username=?', username))
            self.rows.append((username, number_of_zeta_values_calculated))

    def sort_rows(self):
        # sort rows by number of zeta values calculated
        self.sorted_numbers = binary_insertion_sort(
                set([row[-1] for row in self.rows]), descending=True)
        self.sorted_rows = []
        for _ in range(len(self.rows)):
            for row in self.rows:
                if row[-1] == self.sorted_numbers[0]:
                    self.sorted_rows.append(row)
                    self.rows.remove(row)
                    if self.sorted_numbers[0] not in [row[-1] for row in
                         self.rows]:
                        del self.sorted_numbers[0]


class TableCalculator2(InvestigationSection):

    """
    The TableCalculator2 class is used to display the output values of
         the zeta
```

```
function for a range of input values, that were entered by the user
    on the
previous page

The user then has the option to save these to the database or to a
    file
"""

def __init__(self, table_values):
    super(TableCalculator2, self).__init__()
    self.table_values = table_values
    self.ui = Ui_TableCalculator2Screen()
    self.ui.setupUi(self)
    self.setup_tabs()
    self.ui.PrevButton.clicked.connect(self.goto_table_calculator)
    self.ui.NextButton.clicked.connect(self.goto_calculator_leaderboard)
    self.ui.DatabaseButton.clicked.connect(self.saveto_database)
    self.ui.FileButton.clicked.connect(self.saveto_file)
    self.ui.ZetaTable.setRowCount(len(self.table_values))
    for i, values in enumerate(self.table_values):
        for j in range(len(values)):
            self.ui.ZetaTable.setItem(i,j,
                QTableWidgetItem(str(values[j])))
    self.ui.ZetaTable.horizontalHeader().setStretchLastSection(True)
    self.ui.ZetaTable.horizontalHeader().setSectionResizeMode(
        QHeaderView.Stretch)
    self.ui.ZetaTable.setColumnWidth(1, 100)
    self.show()

def saveto_database(self):
    """NEEDS FIXING, which ones to save to db"""
    if User.GetUsername():
        database_inputs = database_select([’Input_Real’,
            ’Input_Imag’], [’Zeta’])
        print(’dbi’, database_inputs)
        for input, output in self.table_values:
            print(f’in {input} out{output}’)
            if (input.get_real(), input.get_imag()) not in
                database_inputs:
                self.Zeta_ID = get_id(’Zeta_ID’, ’Zeta’)
                database_insert(’Zeta’, self.Zeta_ID,
                    input.get_real(), input.get_imag(),
                    output.get_real(), output.get_imag())
                database_insert(’UserZeta’, self.Zeta_ID,
                    User.GetUsername())
        self.ui.ErrorLabel.setText(self.center_text(’Values saved to
            database’))
    else:
        self.ui.ErrorLabel.setText(self.center_text(f’You must be
            signed in to be able to ’
```

```
                    'save to the database'))

    def saveto_file(self):
        FILEPATH = 'files/zeta_values.csv'
        save_zeta_values_to_file(self.table_values, FILEPATH)
        self.ui.ErrorLabel.setText(self.center_text(f'Table contents
            written to {FILEPATH}'))


class TableCalculator(InvestigationSection):

    """
    The TableCalculator class is used to display a calculator where
    the user is able to calculate the value of the zeta function for a
        range of
    input values of their choosing
    """

    def __init__(self):
        super(TableCalculator, self).__init__()
        self.ui = Ui_TableCalculatorScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_single)
        self.ui.NextButton.clicked.connect(self.goto_calculator_leaderboard)
        self.ui.CalculateButton.clicked.connect(self.goto_table_calculator_2)
        self.show()

    def calculate_zeta(self):
        self.start_input = self.ui.StartInput.text()
        self.step_input = self.ui.StepInput.text()
        self.range_input = self.ui.NoOfValuesInput.text()
        self.start_complex =
            self.get_valid_complex_input(self.start_input)
        self.step_complex = self.get_valid_complex_input(self.step_input)
        self.range = make_int(self.range_input)
        print(self.start_complex, self.step_complex)
        if self.start_complex is not None and self.step_complex is not
            None:
            if 1 <= self.range <= 100:
                self.ui.ErrorLabel.setText('')
                self.input_values = [
                        self.start_complex + self.step_complex * i
                        for i in range(self.range)]
                zetas = [zeta(value.get_real(), value.get_imag()) for
                    value in self.input_values]
                self.output_values = [
                        Complex(round(i.get_real(), 3),
                        round(i.get_imag(), 3)) for i in zetas]
                self.table_values = list(zip(
                    self.input_values, self.output_values))
            else:
                self.ui.ErrorLabel.setText(self.center_text('No. Of
                     Values must be a positive \
                        integer between 1 and 100'))
                self.table_values = False
        else:
            self.ui.ErrorLabel.setText(self.center_text('Start Value and
                 Step must be complex \
                    numbers of the form a+bi'))
            self.table_values = False


class SingleCalculator(InvestigationSection):

    """
    The SingleCalculator class is used to display a calculator where
    the user is able to calculate the value of the zeta function for a
         given
    input of their choosing

    The user then has the option to save this values to the database or
         to a file
    """

    def __init__(self):
        super(SingleCalculator, self).__init__()
        self.ui = Ui_SingleCalculatorScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.zeta_value = []
        self.valid_input = False
        self.ui.PrevButton.clicked.connect(self.goto_calculator)
        self.ui.NextButton.clicked.connect(self.goto_table_calculator)
        self.ui.CalculateButton.clicked.connect(self.calculate_zeta)
        self.ui.DatabaseButton.clicked.connect(self.saveto_database)
        self.ui.FileButton.clicked.connect(self.saveto_file)
        self.show()

    def calculate_zeta(self):
        self.zeta_user_input = str(self.ui.ZetaInput.text()).strip()
        self.zeta_input =
             self.get_valid_complex_input(self.zeta_user_input)
        print(self.zeta_input)
        if self.zeta_input is not None:
            self.zeta_output = zeta(self.zeta_input.get_real(),
                 self.zeta_input.get_imag())
            self.zeta_output_printable =
                 Complex(round(self.zeta_output.get_real(), 3),
                 round(self.zeta_output.get_imag(), 3))
            self.zeta_value = [(self.zeta_input,
                self.zeta_output_printable)]
            self.ui.ZetaOutput.setText(str(self.zeta_output_printable)[1:-1])
            self.ui.ErrorLabel.setText(’’)
        else:
            self.ui.ErrorLabel.setText(self.center_text(’Input must be a
                complex number of the form a+bi’))
            self.ui.ZetaOutput.setText(’’)

    def saveto_database(self):
        self.calculate_zeta()
        # may have toa dd error checking here
        #  self.zeta_user_input = str(self.ui.ZetaInput.text()).strip()
        #  self.zeta_input =
            self.get_valid_complex_input(self.zeta_user_input)
        if self.zeta_input is not None:
            if User.GetSignedIn():
                database_inputs = database_select([’Input_Real’,
                    ’Input_Imag’], [’Zeta’])
                self.zeta_input_real = self.zeta_input.get_real()
                self.zeta_input_imag = self.zeta_input.get_imag()
                self.zeta_output_real =
                    self.zeta_output_printable.get_real()
                self.zeta_output_imag =
                    self.zeta_output_printable.get_imag()
                if (self.zeta_input_real, self.zeta_input_imag) not in
                    database_inputs:
                    self.Zeta_ID = get_id(’Zeta_ID’, ’Zeta’)
                    database_insert(’Zeta’,
                            self.Zeta_ID,
                            self.zeta_input_real,
                            self.zeta_input_imag,
                            self.zeta_output_real,
                            self.zeta_output_imag)
                    database_insert(’UserZeta’, self.Zeta_ID,
                        User.GetUsername())
                    self.ui.ErrorLabel.setText(self.center_text(’Value
                        saved to database’))
                else:
                    self.ui.ErrorLabel.setText(self.center_text(’Value has
                        already been recorded in the database’))
            else:
                self.ui.ErrorLabel.setText(
                        self.center_text(’You must be signed in to be able
                            to ’
                        ’save to the database’))
        else:
            self.ui.ErrorLabel.setText(
                    self.center_text(’Input value is not valid’))
    def saveto_file(self):
        self.calculate_zeta()
        FILEPATH = 'files/zeta_values.csv'
        save_zeta_values_to_file(self.zeta_value, FILEPATH)
        self.ui.ErrorLabel.setText(self.center_text(f'Values written to
            {FILEPATH}'))


class Calculator(InvestigationSection):

    """
    The Calculator class is used to display the calculator screen where
    the user can choose to calculate the value of the zeta function for
        a single
    value or for a table of values
    """

    def __init__(self):
        super(Calculator, self).__init__()
        self.question_no = 7
        self.ui = Ui_CalculatorScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_primes)
        self.ui.NextButton.clicked.connect(self.goto_zeroes)
        self.ui.ZetaCalculatorButton.clicked.connect(self.goto_single)
        self.show()


class PrimeNumbers(InvestigationSection):

    """
    The PrimeNumbers class is used to display the prime numbers screen
        where
    the user is given information about the prime numbers and how they
        relate
    to the riemann zeta function
    """

    def __init__(self):
        super(PrimeNumbers, self).__init__()
        self.ui = Ui_PrimeNumbersScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_graph_plots)
        self.ui.NextButton.clicked.connect(self.goto_calculator)
        self.show()
```

```
class ZetaApproximationMatPlot(DynamicGraphScreen):

    """
    The ZetaApprozimation Mat Plot
    """

    def __init__(self, complex_input):
        super(ZetaApproximationMatPlot, self).__init__()
        self.complex_input = complex_input
        self.show()

    def update_figure(self):
        self.zeta_value = zeta(self.complex_input.get_real(),
            self.complex_input.get_imag(), self.count)
        self.x_vals.append(self.zeta_value.get_real())
        self.y_vals.append(self.zeta_value.get_imag())
        self.matplotlibwidget.axes.cla()
        self.matplotlibwidget.axes.plot(self.x_vals, self.y_vals,
            color='green')
        self.matplotlibwidget.canvas.draw()
        self.count += 1


class ZetaApproximation(InvestigationSection):

    """
    The ZetaApproximation
    """

    def __init__(self):
        super(ZetaApproximation, self).__init__()
        self.ui = Ui_ZetaApproximationScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_prime)
        self.ui.NextButton.clicked.connect(self.goto_primes)
        self.ui.GraphButton.clicked.connect(self.get_complex_input)
        self.show()

    def get_complex_input(self):
        self.graph_user_input = str(self.ui.GraphInput.text()).strip()
        self.graph_input =
            self.get_valid_complex_input(self.graph_user_input)
        # make sure that the value entered is within range
        if self.graph_input is None:
            self.ui.ErrorLabel.setText(self.center_text('Input must be a
                complex number of the form a+bi'))
            self.ui.ZetaOutput.setText('')
        else:
            self.zeta_approximation_graph =
                ZetaApproximationMatPlot(self.graph_input)


class PrimeCountingFunctionMatPlot(DynamicGraphScreen):

    """
    The PrimeCountingFunction class is used to display a graph of the
    prime coutning function, the prime power function, the logarithmic
        integral
    function and (x/log x) as an approximation for the prime counting
        function
    """

    def __init__(self):
        super(PrimeCountingFunctionMatPlot, self).__init__()
        self.y_vals_pcf = []
        self.y_vals_x_logx= []
        self.y_vals_li = []
        self.y_vals_ppf = []
        self.count = 2
        self.show()

    def update_figure(self):
        self.x_vals.append(self.count)
        self.y_vals_pcf.append(sieve_of_eratosthenes(self.count).size)
        self.y_vals_x_logx.append(prime_counting_function_estimation(self.count))
        self.y_vals_li.append(logarithmic_integral(self.count))
        self.y_vals_ppf.append(prime_power_function(self.count))
        self.matplotlibwidget.axes.cla()
        self.matplotlibwidget.axes.scatter(self.x_vals, self.y_vals_pcf,
            label='Prime Counting Function')
        self.matplotlibwidget.axes.plot(self.x_vals, self.y_vals_x_logx,
            label='x / log(x)', color='red')
        self.matplotlibwidget.axes.plot(self.x_vals, self.y_vals_li,
            label='Logarithmic Integral', color='green')
        self.matplotlibwidget.axes.plot(self.x_vals, self.y_vals_ppf,
            label='Prime Power Function', color='blue')
        self.matplotlibwidget.axes.legend(loc='upper left')
        self.matplotlibwidget.canvas.draw()
        self.count += 1


class PrimeCountingFunction(InvestigationSection):

    """
    The PrimeCountingFunction class is used to display the prime
        counting function
    screen
    """

    def __init__(self):
        super(PrimeCountingFunction, self).__init__()
        self.ui = Ui_PrimeCountingFunctionScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_zeta_zeroes_plot)
        self.ui.NextButton.clicked.connect(self.goto_graph_plots)
        self.ui.GraphButton.clicked.connect(self.goto_pcf_graph)
        self.show()


class ZetaZeroesMatPlot(DynamicGraphScreen):

    """
    The ZetaZeroesMatPlot class is used to display a graph of the zeroes
        of the
    riemann zeta function
    """

    def __init__(self):
        super(ZetaZeroesMatPlot, self).__init__()
        self.show()

    def update_figure(self):
        self.accuracy = self.count//500 + 100
        if is_zeta_zero(1/2, self.count/self.accuracy):
            self.x_vals.append(1/2)
            self.y_vals.append(self.count/self.accuracy)
        self.matplotlibwidget.axes.cla()

        self.matplotlibwidget.axes.scatter(self.x_vals, self.y_vals)
        self.matplotlibwidget.axes.set_ylim(0)
        self.matplotlibwidget.axes.set_xlim(0, 1)
        self.matplotlibwidget.canvas.draw()
        self.count += 1


class ZetaZeroesPlot(InvestigationSection):

    """
    The ZetaZeroes class is used to display the Zeroes screen in the
        investigation
    section

    This is where the user is able to read about what the zeta zeroes
        are, and
    be able to display a graph of the zeta zeroes
    """
    def __init__(self):
        super(ZetaZeroesPlot, self).__init__()
        self.question_no=6
        self.ui = Ui_ZetaZeroesPlotScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_polar)
        self.ui.NextButton.clicked.connect(self.goto_prime)
        self.ui.GraphButton.clicked.connect(self.goto_zeta_zeroes_graph)
        self.show()



class PolarGraphMatPlot(DynamicGraphScreen):

    """
    The PolarGraphMatPlot class is used to display the polar graph of
        the riemann
    zeta function
    """

    def __init__(self, real_input):
        super(PolarGraphMatPlot, self).__init__()
        self.real_input = real_input
        self.show()

    def update_figure(self):
        new_zeta = zeta(self.real_input, self.count/25)
        self.x_vals.append(new_zeta.get_real())
        self.y_vals.append(new_zeta.get_imag())
        self.matplotlibwidget.axes.cla()
        self.matplotlibwidget.axes.plot(self.x_vals, self.y_vals, 'r')
        self.matplotlibwidget.canvas.draw()
        self.count += 1


class PolarGraph(InvestigationSection):

    """
    The PolarGraph class is used to display the Polar Graph Screen

    This is where the user is able to read about polar graphs, and be
        able to
    display a polar graph of the riemann zeta function
    """

    def __init__(self):
        super(PolarGraph, self).__init__()
        self.ui = Ui_PolarGraphScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_graph_plots)
        self.ui.NextButton.clicked.connect(self.goto_zeta_zeroes_plot)
        self.ui.GraphButton.clicked.connect(self.polar_graph)
        self.show()

    def polar_graph(self):
        self.real_input = self.ui.GraphInput.text()
        try:
            self.real_input = float(self.real_input)
        except ValueError:
            self.ui.ErrorLabel.setText(self.center_text("Error: Input
                must be whole number or a decimal"))
        else:
            if self.real_input == 1:
                self.ui.ErrorLabel.setText(self.center_text("Error: Input
                    must not be equal to 1"))
            elif not -10 < self.real_input < 45:
                self.ui.ErrorLabel.setText(self.center_text("Error: Input
                    value must be between -10 and 45"))
            else:
                self.ui.ErrorLabel.setText('')
                self.graph = PolarGraphMatPlot(self.real_input)


class GraphPlot(InvestigationSection):

    """
    This class is used to display the Graph Plots screen in the
        investigation section

    This is the first screen that the user will see in the investigation
        section
    and will allow them to display many different types of graphs
    """

    def __init__(self):
        super(GraphPlot, self).__init__()
        self.ui = Ui_GraphPlotsScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.GraphPlotsButton.clicked.connect(self.goto_polar)
        self.ui.PrevButton.clicked.connect(self.goto_mainmenu)
        self.ui.NextButton.clicked.connect(self.goto_primes)
        self.show()
```

program/summary_section.py

```
"""
summary_section.py
==================

Contains all of the classes used to interact with the GUI for the
summary section of the project

Includes the main SummarySection class which is inherited by the Summary,
TheoryRecap, InvestigationResults, Conclusion, Impact
"""

from PyQt5 import QtWidgets
from PyQt5.QtWidgets import QTableWidget,QTableWidgetItem, QHeaderView
from .user_interface import Ui_SummaryScreen, Ui_TheoryRecapScreen,
    Ui_InvestigationResultsScreen, Ui_ConclusionScreen, Ui_ImpactScreen
from .utils import User, Screen, database_select, Complex
from .notes import SummaryNotes


class SummarySection(Screen):

    """
    A class inherited by all of the Screens/Page classes in the summary
    section of the program

    The functions defined in this class allow for different pages to be
        loaded
    and hidden, so that the user is able to navigate to different parts
        of the
    program using the GUI
    """

    def __init__(self):
        super(SummarySection, self).__init__()

    def setup_tabs(self):

        """
        Allows the tabs and buttons to run a function once clicked, if
            they
        exists on the web page that the tab/button was clicked on
        """

        self.ui.SummaryTab.clicked.connect(self.goto_summary)
        self.ui.TheoryRecapTab.clicked.connect(self.goto_theory_recap)
        self.ui.InvestigationResultsTab.clicked.connect(self.goto_investigation_results)
        self.ui.ConclusionTab.clicked.connect(self.goto_conclusion)
        self.ui.ImpactTab.clicked.connect(self.goto_impact)
        try:
            self.ui.NotesButton.clicked.connect(self.goto_summary_notes)
        except AttributeError:
            pass

    """
    The goto functions are run when a tab is clicked. They load a new
         page,
    and hide the old page.
    """

    def goto_summary(self):
        self.summary = Summary()
        self.hide()

    def goto_theory_recap(self):
        self.theory_recap = TheoryRecap()
        self.hide()

    def goto_investigation_results(self):
        self.investigation_results = InvestigationResults()
        self.hide()

    def goto_conclusion(self):
        self.conclusion = Conclusion()
        self.hide()

    def goto_impact(self):
        self.impact = Impact()
        self.hide()

    def goto_summary_notes(self):
        self.summary_notes = SummaryNotes()


class Summary(SummarySection):

    """
    The Summary Screen is the main entry point ot the summary section of
         the
    program. This class displays the summary screen to the user.
    """

    def __init__(self):
        super(Summary, self).__init__()
        self.question_no = 8
        self.ui = Ui_SummaryScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_mainmenu)
        self.ui.NextButton.clicked.connect(self.goto_theory_recap)
        self.show()


class TheoryRecap(SummarySection):

    """
    Theory Recap Screen class displays this screen to the user as part
    of the sumary section of this program
    """

    def __init__(self):
        super(TheoryRecap, self).__init__()
        self.question_no = 9
        self.ui = Ui_TheoryRecapScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_summary)
        self.ui.NextButton.clicked.connect(self.goto_investigation_results)
        self.show()


class InvestigationResults(SummarySection):

    """
    The Investigation Results Screen class displays this screen to the
        user as part
    of the sumary section of this program
    """

    def __init__(self):
        super(InvestigationResults, self).__init__()
        self.ui = Ui_InvestigationResultsScreen()
        self.ui.setupUi(self)
        self.ui.PrevButton.clicked.connect(self.goto_theory_recap)
        self.ui.NextButton.clicked.connect(self.goto_conclusion)
        self.setup_tabs()
        self.setup_table()
        self.show()

    def setup_table(self):
        """
        Populates the table on this screen with a list of inputs and
        outputs of the zeta function
        """
        self.values = database_select([’*’], [’Zeta’])
        self.table_values = [(Complex(value[1], value[2]),
            Complex(value[3], value[4])) for value in self.values]
        self.ui.ZetaTable.setRowCount(len(self.table_values))
        for i, values in enumerate(self.table_values):
            for j in range(len(values)):
                self.ui.ZetaTable.setItem(i,j,
                    QTableWidgetItem(str(values[j])))
        self.ui.ZetaTable.horizontalHeader().setStretchLastSection(True)
        self.ui.ZetaTable.horizontalHeader().setSectionResizeMode(
            QHeaderView.Stretch)
        self.ui.ZetaTable.setColumnWidth(1, 100)


class Conclusion(SummarySection):

    """
    The Conclusion Screen class displays this screen to the user as part
    of the sumary section of this program
    """

    def __init__(self):
        super(Conclusion, self).__init__()
        self.ui = Ui_ConclusionScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.ui.PrevButton.clicked.connect(self.goto_investigation_results)
        self.ui.NextButton.clicked.connect(self.goto_impact)
        self.show()


class Impact(SummarySection):

    """
    The Impact Screen class displays this screen to the user as part
    of the sumary section of this program
    """

    def __init__(self):
        super(Impact, self).__init__()
        self.question_no = 10
        self.ui = Ui_ImpactScreen()
        self.ui.setupUi(self)
        self.setup_tabs()
        self.setup_question()
        self.ui.PrevButton.clicked.connect(self.goto_conclusion)
        self.ui.NextButton.clicked.connect(self.goto_mainmenu)
        self.show()
```

program/notes.py

```
"""
```

```
notes.py
========

Contains all of the classes used to interact with the GUI for the
user to be able to take notes in the program

Includes notes screens for the tutorial, introduction, investigation and
    summary
screens
"""


from PyQt5 import QtCore, QtGui, QtWidgets
from .utils import Screen, database_query, database_insert, User, get_id
from .user_interface import Ui_TutorialNotesScreen,
    Ui_IntroductionNotesScreen, Ui_InvestigationNotesScreen,
    Ui_SummaryNotesScreen


class Notes(Screen):

    """
    A class inherited by all of the Screens/Page classes in the notes
        section
    of the program

    The functions defined in this class allow for different pages to be
        loaded
    and hidden, so that the user is able to navigate to different parts
        of the
    program using the GUI
    """

    def __init__(self):
        super(Notes, self).__init__()

    def goto_tutorial_notes(self):
        self.tutorial_notes = TutorialNotes()
        self.hide()

    def goto_introduction_notes(self):
        self.introduction_notes = IntroductionNotes()
        self.hide()

    def goto_investigation_notes(self):
        self.investigation_notes = InvestigationNotes()
        self.hide()

    def goto_summary_notes(self):
        self.summary_notes = SummaryNotes()
        self.hide()

    def exit_notes(self):
        self.hide()

    def saveto_database(self):
        self.text = self.ui.NotesText.toPlainText()
        database_query("DELETE FROM Notes WHERE Section=? AND
            Username=?", self.section, User.GetUsername())
        database_insert(’Notes’, User.GetUsername(), self.section,
            self.text)
        self.set_text_saved()

    def set_text_saved(self):
        self.ui.SavedText.setStyleSheet("color: rgb(0, 140, 0);\n"
               "font: 18pt \"Sans Serif\";")
        self.ui.SavedText.setText(’Saved!’)

    def set_text_unsaved(self):
        self.ui.SavedText.setStyleSheet("color: rgb(255, 0, 0);\n"
               "font: 18pt \"Sans Serif\";")
        self.ui.SavedText.setText(’Unsaved’)

    def set_text(self):
        self.db_text = database_query("SELECT Text FROM Notes WHERE
            Section=? AND Username=?", self.section, User.GetUsername())
        if len(self.db_text) == 0:
            self.text = ’’
        else:
            self.text = self.db_text[0][0]
        self.set_note_text(self.text)

    def set_note_text(self, text, color=’rgb(69, 69, 69)’):
        self.ui.NotesText.setHtml("<!DOCTYPE HTML PUBLIC \"-//W3C//DTD
            HTML 4.0//EN\"
            \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
        "<html><head><meta name=\"qrichtext\" content=\"1\" /><style
            type=\"text/css\">\n"
        "p, li { white-space: pre-wrap; }\n"
        f"</style></head><body style=\" font-family:\’Sans Serif\’;
            font-size:18pt; font-weight:400; font-style:normal;
            color:{color};\">\n"
        f"<p style=\" margin-top:0px; margin-bottom:0px;
            margin-left:0px; margin-right:0px; -qt-block-indent:0;
            text-indent:0px;\">{text}</p></body></html>")

    def not_signed_in(self):
        self.set_note_text(’You must be signed in to be able to make
            notes’,
               color=’rgb(255, 0, 0)’)
        self.ui.NotesText.setReadOnly(True)

    def signed_in(self):
        self.ui.NotesText.setReadOnly(False)
        self.set_text()
        self.ui.NotesText.textChanged.connect(self.set_text_unsaved)
        self.ui.TutorialTab.clicked.connect(self.goto_tutorial_notes)
        self.ui.IntroductionTab.clicked.connect(self.goto_introduction_notes)
        self.ui.InvestigationTab.clicked.connect(self.goto_investigation_notes)
        self.ui.SummaryTab.clicked.connect(self.goto_summary_notes)
        self.ui.SaveButton.clicked.connect(self.saveto_database)


class TutorialNotes(Notes):

    """
    The TutorialNotes class is used to allow the user to take notes on
        the
    tutorial section of the program
    """

    def __init__(self):
        super(TutorialNotes, self).__init__()
        self.section = 'Tutorial'
        self.ui = Ui_TutorialNotesScreen()
        self.ui.setupUi(self)
        self.ui.BackButton.clicked.connect(self.exit_notes)
        if User.GetSignedIn():
            self.signed_in()
            self.ui.NextButton.clicked.connect(self.goto_introduction_notes)
        else:
            self.not_signed_in()
        self.show()


class IntroductionNotes(Notes):

    """
    The IntroductionNotes class is used to allow the user to take notes
        on the
    introduction section of the program
    """

    def __init__(self):
        super(IntroductionNotes, self).__init__()
        self.section = 'Introduction'
        self.ui = Ui_IntroductionNotesScreen()
        self.ui.setupUi(self)
        self.ui.BackButton.clicked.connect(self.exit_notes)
        if User.GetSignedIn():
            self.signed_in()
            self.ui.NextButton.clicked.connect(self.goto_investigation_notes)
        else:
            self.not_signed_in()
        self.show()


class InvestigationNotes(Notes):

    """
    The InvestigationNotes class is used to allow the user to take notes
        on the
    investigation section of the program
    """

    def __init__(self):
        super(InvestigationNotes, self).__init__()
        self.section = 'Investigation'
        self.ui = Ui_InvestigationNotesScreen()
        self.ui.setupUi(self)
        self.ui.BackButton.clicked.connect(self.exit_notes)
        if User.GetSignedIn():
            self.signed_in()
            self.ui.NextButton.clicked.connect(self.goto_summary_notes)
        else:
            self.not_signed_in()
        self.show()


class SummaryNotes(Notes):

    """
    The SummaryNotes class is used to allow the user to take notes on the
    summary section of the program
    """

    def __init__(self):
        super(SummaryNotes, self).__init__()
        self.section = 'Summary'
        self.ui = Ui_SummaryNotesScreen()
        self.ui.setupUi(self)
        self.ui.BackButton.clicked.connect(self.exit_notes)
        if User.GetSignedIn():
            self.signed_in()
        else:
            self.not_signed_in()
        self.show()
```

program/utils/__init__.py

```
"""
__init__.py
===========
Imports for utils
"""


from .mathematical_functions import zeta, sieve_of_eratosthenes,
    prime_power_function, prime_counting_function_estimation,
    logarithmic_integral, is_zeta_zero, make_int, make_complex
from .computational_functions import binary_insertion_sort,
    save_zeta_zeroes_to_file, save_zeta_values_to_file
from .database_functions import database_insert, database_select,
    database_query, database_print, create_database, get_id
from .cryptography_functions import hash_password, check_password
from .email_functions import send_verification_email
from .user import User
from .screen_design import Screen, StaticGraphScreen, DynamicGraphScreen
from .number_systems import Number, Complex
```

program/utils/computational_functions.py

```
"""
computational_functions.py
==========================

Contains subroutines and classes that are used for various different
    algorithms

Subroutines:
    - binary_search
    - binary_insertion_sort
    - save_zeta_zeroes_to_file
    - save_zeta_values_to_file
    - save_zeta_to_file
    - change_datatype

Classes
    - Queue
"""


import random
import time
import csv
import re
import os
```

```
def binary_search(data, target):

    """
    For a set of unique elements that are sorted in ascending order
        (data),
    this function will return the index where target is in the data,
        with time
    complexity O(log n)
    """

    low = 0
    high = len(data)
    if target < min(data):
        return 0
    elif target > max(data):
        return len(data)
    else:
        while True:
            mid = (high+low) // 2
            if data[mid] < target < data[mid+1]:
                return mid+1
            elif target < data[mid]:
                high=mid
            elif target > data[mid]:
                low=mid
            else:
                raise ValueError('Error: Input list was not a set')


def binary_insertion_sort(data, descending=False):

    """
    Sorts a list of data into ascending order using a binary insertion
        sort
    """

    try:
        array = list(map(float, data ))
    except ValueError:
        array = data
    queue = Queue(array)
    sorted = [queue.deQueue()]
    while not queue.is_empty():
        item = queue.deQueue()
        index = binary_search(sorted, item)
        bigger = sorted[index:]
        del sorted[index:]
        sorted.append(item)
        sorted.extend(bigger)
    if not descending:
        return sorted
    else:
        return sorted[::-1]


class Queue:

    """
    Implementation of a circular queue

    Contains the subroutines:
        - enQueue
        - deQueue
        - is_full
        - is_empty

    """

    def __init__(self, input_queue, **kwargs):
        self.input_queue = input_queue
        self.size = len(self.input_queue)
        self.front = 0
        self.rear = len(self.input_queue)
        if ’max_size’ in kwargs.keys():
            self.max_size = kwargs[’max_size’]
        else:
            self.max_size = len(self.input_queue)
        if self.size > self.max_size:
            raise IndexError("max_size must be greater than or equal to
                the \
                   size of the input queue")
        else:
            self.blanks = [False for i in range(self.max_size -
                self.size)]
            self.queue = self.input_queue + self.blanks

    def enQueue(self, item):

        """
        Appends an item to the rear of the circular queue
        """

        if self.is_full():
            raise IndexError("Tried to enqueue to a full queue")
        else:
            self.queue[self.rear] = item
            self.rear = (self.rear+1) % self.max_size
            self.size += 1
    def deQueue(self):

        """
        Remove and return the value at the front of the circular queue
        """

        if self.is_empty():
            raise IndexError("Tried to dequeue from an empty queue")
        else:
            item = self.queue[self.front]
            self.queue[self.front] = False
            self.front = (self.front+1) % self.max_size
            self.size -= 1
            return item

    def is_full(self):

        """
        Check if the circular queue is full
        """

        return self.size == self.max_size

    def is_empty(self):

        """
        Check if the circular queue is empty
        """

        return self.size == 0

    def __str__(self):

        """
        Return a printable value of the queue
        """

        return 'Queue(' + ', '.join([str(i) for i in self.queue]) + ')'


def save_zeta_values_to_file(table_values, filepath,
        fieldnames=['InputReal', 'InputImag', 'OutputReal',
            'OutputImag']):

    """
    Given a list of corresponding inputs and outputs of the zeta
         function,
    save these sets of values to a file.
    """
    csv_values = [list(map(str,
        [input.get_real(), input.get_imag(), output.get_real(),
            output.get_imag()]))
        for input, output in table_values]
    regex = r'-?\d+\.\d+'
    index = 0
    save_zeta_to_file(csv_values, filepath, regex, index, fieldnames)


def save_zeta_zeroes_to_file(table_values, filepath,
        fieldnames=['InputReal', 'InputImag', 'OutputReal',
            'OutputImag']):

    """
    Given a list of zeroes/roots of the zeta function,
    save these zeroes to a file
    """

    csv_values = [list(map(str, [real, imag])) for real, imag in
        table_values]
    regex = r'-?\d+\.\d+'
    index = 1
    save_zeta_to_file(csv_values, filepath, regex, index, fieldnames)


def save_zeta_to_file(csv_values, filepath, regex, index, fieldnames):

    """
    Given a list of imaginary numbers, combine these with the contents
        of the
    file that they are going to be saved to, sort these values using the
        first
    real number, and save them back into the csv file
    """

    if not os.path.isfile(filepath):
        os.mknod(filepath)
    with open(filepath, 'r') as csv_file:
        csv_reader = csv.reader(csv_file)
        for row in csv_reader:
            if row != fieldnames:
                csv_values.append(list(map(str, row)))
    sorting_dict = {list(map(float,
        re.findall(regex, ','.join(row))))[index]: row for row in
            csv_values}
    sorted_keys = binary_insertion_sort(list(set(sorting_dict.keys())))
    sorted_values = [sorting_dict[key] for key in sorted_keys]
    with open(filepath, 'w') as csv_file:
        csv_writer = csv.writer(csv_file)
        csv_writer.writerow(fieldnames)
        for row in sorted_values:
            csv_writer.writerow(row)


def change_datatype(value, datatype):

    """
    Change the datatype of a value, if the datatype is given as a
        string, and not
    as a python keyword
    """

    match str(datatype):
        case 'int':
            return int(value)
        case 'str':
            return str(value)
        case 'float':
            return float(value)
        case 'bool':
            return bool(value)
        case 'list':
            return list(value)
        case 'complex':
            return Complex(value)
        case 'fraction':
            return Fraction(value)
        case _:
            raise TypeError(f'Trying to change vlaue \'{value}\' \
                    to invalid datatype \'{datatype}\'')
```

program/utils/cryptography_functions.py

```
"""
cryptography_functions.py
=========================

Contains subroutines that are used for encyption and hashing

These subroutines include:
    - get_pepper
    - hash_password
    - check_password
"""


import argon2
from argon2 import PasswordHasher as ph
```

```
def get_pepper():

    """
    Returns the value of the pepper that is used in this program to help
    secure the hashes in the database
    """

    return 'Sr4QkXyIhv4SiijxAWxU'


def hash_password(password):

    """
    Takes in a password as a parameter and returns the hashed version of
        this
    password using the pepper and the argon2 encryption algorithm
    """

    hash = ph().hash(get_pepper() + password)
    return hash


def check_password(password, hash):

    """
    Used to check whether a given hash is the hashed version of a
        password

    Because hashing cannot be undone to check whether the hash is the
        same as a
    password, it is required that the password is hashed, and if this
        hash is the
    same as the one we are comparing it to, then the passwords are the
        same.
    """

    try:
        valid = ph().verify(hash, get_pepper() + password)
    except argon2.exceptions.VerifyMismatchError:
        return False
    else:
        return True
```

program/utils/database_functions.py

```
"""
database_functions.py
=====================
Contains subroutines that are used to query and interact with SQL
    databases

These subroutines include:
    - database_select
    - database_insert
    - database_query
    - database_print
    - database_print
    - create_users_table
    - create_correct_answers_table
    - create_questions_table
    - create_user_answer_table
    - create_notes_table
    - create_zeta_table
    - create_user_zeta_table
    - create_zeroes_table
    - create_user_zeroes_table
    - create_database
    - delete_table
    - delete_database
    - reset_database
    - get_next_id
    - get_id
"""


import sqlite3
import os
from .file_handling import touch, remove


def database_select(headings, tables):

    """
    database_select takes in a list of heading and tables as inputs, and
        returns
    the query from selecting these from the SQL database
    """

    query = f"SELECT {’, ’.join(headings)} FROM {’, ’.join(tables)}"
    return database_query(query)


def database_insert(table, *args):

    """
    database_insert inserts the arguments into a desired table in the
        database
    """

    query = f"INSERT INTO {table} VALUES ({’, ’.join(’?’ for _ in
        args)})"
    database_query(query, *args)


def database_query(query, *args, database=’database.db’):

    """
    database_query takes in a query as an input, along with any requried
        arguments.
    the function executes the query on the database.
    This allows the database to be queried from any point in the program
        without
    having to pass a database variable into every function, or by using
        a global
    variable
    """

    conn = sqlite3.connect(database)
    cursor = conn.cursor()
    cursor.execute(query, args)
    rows = cursor.fetchall()
    conn.commit()
    conn.close()
    return rows


def database_print():

    """
    database_print was mainly used for testing purposese. This function
        gets the
    name of every table in the database, prints to the console, each
        table, with
    the data in each table
    """

    print(’=’*16)
    print(’=  DATABASE  =’)
    print(’=’*16)
    tables = [table[0] for table in database_query("""SELECT name FROM
        sqlite_master
    WHERE type=’table’""")]
    for table in tables:
        name = f"- Table: {table} -"
        border = ’-’ * len(name)
        print(f’{border}\n{name}\n{border}’)
        print(’, ’.join([column[1] for column in database_query(f"PRAGMA
             table_info({table})")]))
        rows = database_select([’*’], [table])
        for row in rows:
            print(row)
    print(’=’ * 20)


def create_users_table():

    """ Create the Users table in the database """

    database_query(""" CREATE TABLE Users(
    Username text PRIMARY KEY,
    Email text,
    Password text
    )""")


def create_correct_answers_table(questions_and_answers):

    """ Create the Answers table in the database """

    database_query(""" CREATE TABLE CorrectAnswers(
    Question_No integer,
    CorrectAnswer text,
    PRIMARY KEY (Question_No, CorrectAnswer)
    )""")
    for question_no, dict in enumerate(questions_and_answers):
        for answer in dict["Answers"]:
            database_insert(’CorrectAnswers’, question_no, answer)


def create_questions_table(questions_and_answers):

    """ Create the Questions table in the database """

    database_query(""" CREATE TABLE Questions(
    Question_No integer PRIMARY KEY,
    Question text
    )""")
    for question_no, dict in enumerate(questions_and_answers):
        database_insert(’Questions’, question_no, dict["Question"])


def create_user_answer_table():

    """ Create the User Answer table in the database """

    database_query(""" CREATE TABLE UsersAnswers(
    Question_No integer,
    Username integer,
    UsersAnswer integer,
    PRIMARY KEY (Question_No, Username)
    )""")


def create_notes_table():

    """ Create the Notes table in the database """

    database_query(""" CREATE TABLE Notes(
    Username integer,
    Section text,
    Text text,
    PRIMARY KEY (Username, Section)
    )""")


def create_zeta_table():

    """ Create the Zeta table in the database """

    database_query(""" CREATE TABLE Zeta(
    Zeta_ID integer PRIMARY KEY,
    Input_Real REAL,
    Input_Imag REAL,
    Output_Real REAL,
    Output_imag REAL
    )""")


def create_user_zeta_table():

    """ Create the User Zeta table in the database """

    database_query(""" CREATE TABLE UserZeta(
    Zeta_ID integer,
    Username integer,
    PRIMARY KEY (Zeta_ID, Username)
    )""")


def create_zeroes_table():

    """ Create the Zeroes table in the database """

    database_query(""" CREATE TABLE Zeroes(
    Zero_ID integer PRIMARY KEY,
    Zero_Real_Input real,
    Zero_Imag_Input real
    )""")


def create_user_zeroes_table():

    """ Create the User Zeta Zeroes table in the database """

    database_query(""" CREATE TABLE UserZeroes(
    Zero_ID integer,
    Username integer,
    PRIMARY KEY (Zero_ID, Username)
    )""")


def delete_table(table):
    try:
        database_query(f"DROP TABLE IF EXISTS {table}")
    except sqlite3.OperationalError as error:
        print(error)


def create_database(database=’database.db’):

    """
    Create the database and all of the tables if it doesnt already exist
    """

    if not os.path.isfile(database):
        touch(database)
        QUESTIONS_AND_ANSWERS = [
        {’Question’: ’Error’,
            ’Answers’: [’Error’]},
        {’Question’: ’What is the name of this program? Visualising the
            ___ Hypothesis’,
            ’Answers’: [’Riemann’]},
        {’Question’: ’What is 1+1?’,
            ’Answers’: [’2’, ’Two’]},
        {’Question’: ’Which character is used to denote the imaginary
            unit?’,
            ’Answers’: [’i’, ’j’]},
        {’Question’: ’What is one practical appication of the Riemann
            Hypothesis?’,
            ’Answers’: [’Cryptography’, ’Quantum Physics’, ’Prime
                Numbers’]},
        {’Question’: ’What is the value of the first non-trivial zeta
            zero?’,
            ’Answers’: [’14.1’, ’0.5+14.1i’, ’0.5 + 14.1i’, ’0.5+14.1j’,
                ’0.5 + 14.1j’]},
        {’Question’: ’What is hypothesised to be the real part of every
            non-trivial zero of the riemann zeta function?’,
            ’Answers’: [’0.5’, ’1/2’]},
        {’Question’: ’What is the value of (5+5i)’,
            ’Answers’: [’0.974+0.012i’, ’0.974+0.012j’]},
        {’Question’: ’What prize would you get from the Clay Mathematics
             Institute if you managed to prove the Riemann Hypothesis?’,
            ’Answers’: [’$1000000’, ’$1,000,000’, ’One Million Dollars’,
                ’A Million Dollars’]},
        {’Question’: ’What is the name of Riemann\’s 1859 paper where he
             first conjectured the Riemann Hypothesis?’,
            ’Answers’: [’On The Number Of Primes Less Than a Given
                Magnitude’]},
        {’Question’: ’What type of numbers would proving the Riemann
             Hypothesis have an impact over?’,
            ’Answers’: [’Prime Numbers’, ’Primes’, ’Prime’]}
        ]
        create_users_table()
        create_correct_answers_table(QUESTIONS_AND_ANSWERS)
        create_questions_table(QUESTIONS_AND_ANSWERS)
        create_user_answer_table()
        create_notes_table()
        create_zeta_table()
        create_user_zeta_table()
        create_zeroes_table()
        create_user_zeroes_table()


def delete_database(database=’database.db’):

    """ remove the database file, thus deleting the database """

    remove(database)


def reset_database(database=’database.db’):

    """ clear all of the data from inside the database """

    delete_database()
    create_database()


def get_next_id(IDs, ID=0):
    if ID not in IDs:
        return ID
    else:
        return get_next_id(IDs, ID+1)


def get_id(ID, table):
    selection = database_select([ID], [table])
    IDs = set([row[0] for row in selection])
    ID_Number = get_next_id(IDs)
    return ID_Number
```

program/utils/email_functions.py

```
"""
email_functions.py
==================

Contains function that are used to send emails to the user’s email
    address

These Functions include:
    - send_verification_email
    - send_email
"""


from dotenv import load_dotenv, find_dotenv
import smtplib
import os
from .user import User


def send_verification_email(code):

    """
    The send_verification_email function is used when the user has
        forgotten
    their password, so an email is sent to them, with a verification code

    This function loads the sending email’s address and password from a
        .env file
    And then sends and email to the user with the verification code
    """

    load_dotenv(find_dotenv())
    from_addr = os.getenv("EMAIL")
    password = os.getenv("PASSWORD")
    message = f"Dear {User.GetUsername()}\nThank you for using the
        Riemann " \
           f"Hypothesis Program\nYour verification code is: {code}\n" \
           "If this was not you, please make sure that your account is
               secured"
    send_email(from_addr, User.GetEmail(), ’Verification’, message,
        password)
```

```
def send_email(from_addr, to_addr, subject, message, password,
            smtpserver='smtp.gmail.com:587'):

    """
    The send_email function is used to send an email to a given email
        address
    """

    header = f'Subject: {subject}\n'
    header += f'To: {to_addr}\n'
    message = header + message
    server = smtplib.SMTP(smtpserver)
    server.starttls()
    server.login(from_addr,password)
    problems = server.sendmail(from_addr, to_addr, message)
    server.quit()
```

program/utils/file_handling.py

```
"""
file_handling.py
================

Contains subroutines that are used to create and delete files

These subroutines include:
    - touch
    - remove
"""


import os


def touch(path):
    """ Create a file if it does not already exist """
    try:
        with open(path, 'a'):
            os.utime(path, None)
    except Error:
        pass


def remove(path):
    """ Remove a file if it exists """
    try:
        os.remove(path)
    except Error:
        pass
```

program/utils/mathematical_functions.py

```
"""
mathematical_functions.py
=========================

Contains many mathematical functions that are used throughout the program

These Functions include:
    - ncr
    - count
    - zeta
    - is_zeta_zero
    - sieve_of_eratosthenes
    - prime_counting_function_estimation
    - integration
    - exponential_integral
    - logarithmic_integral
    - prime_power_function
    - make_complex
    - make_int
"""


from itertools import islice
from functools import reduce
import numpy as np
from operator import mul
from math import ceil, sqrt, log, floor
import scipy.integrate as integrate
from .number_systems import Complex


def ncr(n, r):

    """
    Binomial Coefficient Calculator
    {n \choose r} = \frac{n!}{r! (n-r)!} , \textrm{ for } n \geq r > 0
    """

    r = min(r, n-r)
    numerator = reduce(mul, range(n, n-r, -1), 1)
    denominator = reduce(mul, range(1, r+1), 1)
    return numerator // denominator


def count(start=0, step=1):

    """
    Returns a generator object which contains all values from start
    to whenever the generator will be stopped iterating over, with a
        given step
    e.g count() -> 1 2 3 4 5 ...
        count(10, 2) -> 10 12 14 16 ...
    """

    n = start
    while True:
        yield n
        n += step


def zeta(real_term, imag_term, number_of_terms=100):

    """
    Riemann Zeta Function
    \zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}
            = \frac{1}{1-2^{1-s}}\sum_{n=0}^{\infty} \frac{1}{2^{n+1}}
            \sum_{k=0}^{n} (-1)^k {n \choose k}(k+1)^{-s}
    This function is implemented using generators
    """

    s = complex(real_term, imag_term)
    if s == 1:
        return Complex('inf')
    else:
        const = 1 / (1 - 2 ** (1 - s))
        term = (1 / 2 ** (n + 1) * sum((-1) ** k * ncr(n, k) * (k + 1)
            ** (-s)
                                   for k in range(n + 1)) for n in
                                       count())
        summation = sum(islice(term, number_of_terms))
        zeta = const * summation
        return Complex(zeta)


def is_zeta_zero(real, imag):

    """
    Given a complex number, the function checks to see if this number is
    approximately a root (zero) of the Riemann Zeta Function
    """

    zeta_value = zeta(real, imag)
    return abs(zeta_value) < 10e-3


def sieve_of_eratosthenes(limit):

    """
    The Sieve of Eratosthenes return a list of all of the prime numbers
        up to
    a given limit
    """

    limit = floor(limit)
    possible_primes = np.ones(limit, dtype=bool)
    possible_primes[0:2:1] = False
    for i in range(2, ceil(sqrt(limit))):
        possible_primes[i*i:limit:i] = False
    return np.flatnonzero(possible_primes)


def prime_counting_function_estimation(N):

    """
    Computes \frac{n}{log(n)} in order to estimate the prime counting
        function
    """

    return N/log(N)


def integration(func, lower_limit, upper_limit, strips=int(1e6)):

    """
    For a given function func, return the definite integral between
    limits lower_limit and upper_limit with accuracy: strips
    This is an implementation of the trapezium formula
    """

    strip_width = (upper_limit - lower_limit) / strips
    term1 = sum(func(lower_limit + strip_width * h) for h in range(1,
        strips))
    term2 = 0.5*(func(upper_limit) + func(lower_limit))
    area = strip_width * (term1 + term2)
    return area


def exponential_integral(x, lower_limit=1e-7, upper_limit=10000):

    """
    Exponential integral function
    \mathrm{Ei}(x) = \int_{-x}^{\infty} \frac{e^{-t}}{t}dt
    """

    def func(t):
        return np.exp(t) / t

    lower_limit = min(np.abs(x), lower_limit)
    upper_limit = max(np.abs(x), upper_limit)
    if x > 0:
        return (integrate.quad(func, -upper_limit, -lower_limit)[0]
               + integrate.quad(func, lower_limit, x)[0])
    else:
        return integrate.quad(func, -upper_limit, x)[0]


def logarithmic_integral(N):

    """
    Logaritmic Integral
    \mathrm{Li}(x) = \int_{0}^{x} \frac{dt}{\ln t}
    """

    return exponential_integral(log(N))


def prime_power_function(N):

    """
    Prime Power Function
    \Pi(N) = \pi(N) + \frac{1}{2}\pi(N^{\frac{1}{2}}) +
            \frac{1}{3}\pi(N^{\frac{1}{3}}) + \dots
          =  \sum_{r=1}^{\lfloor log_2 N\rfloor} \pi(N^{\frac{1}{r}})
    """

    total = 0
    for r in range(1, floor(log(N, 2))+1):
        total += sieve_of_eratosthenes(N**(1/r)).size
    return total


def make_complex(number):

    """
    Given the variable number which is of form a+bi or a+bj and of any
        datatype,
    and where a and b are integers, return this number as one with
    the complex datatype
    """

    try:
        number_complex = Complex(number.replace('i', 'j'))
    except ValueError as e:
        return False
    else:
        return number_complex
```

```
def make_int(number):

    """
    Given the variable number, return this number with the int datatype
        if possible
    otherwise, return False
    """

    try:
        number_int = int(number)
    except ValueError as e:
        return False
    else:
        return number_int
```

program/utils/number_systems.py

```
"""
number_systems.py
=================

Contains classess used to represent different types of numbers using
a user-definied abstract datatype,

Classes:
    - Number
    - Complex
"""


from math import sin, cos, atan2, sqrt, exp, log
import re


class Number:

    """ A generic class for numbers"""

    def __init__(self, number):
        self.__number = number

    def get_number(self):
        return self.__number

    def __str__(self):
        print(self.__number)


class Complex(Number):
```

```
"""
Operations and arithmetic involving complex numbers
Operations include:
    + addition
    - subtraction
    * multiplication
    / division
    ** exponentiation
    == equal to
    != not equal to
    abs
    str
    repr
    conjugate
    phase
    dump
    polar
    rect
    get_real
    get_imag
"""

def __init__(self, *args, rect=True):
    super().__init__([*args])
    self.args = args
    self.rect = rect
    if not self.rect:
        # make complex if input in polar coordinates form
        r = float(self.args[0])
        phi = float(self.args[1])
        self.__real = r * cos(phi)
        self.__imag = r * sin(phi)
    else:
        if isinstance(self.args[0], Complex):
            # make complex if input in Complex form
            self.__real = self.args[0].get_real()
            self.__imag = self.args[0].get_imag()
        elif isinstance(self.args[0], complex):
            # make complex if input in python complex form
            self.__real = self.args[0].real
            self.__imag = self.args[0].imag
        else:
            # make complex if input in rect coordinates form
            self.__real = float(self.args[0])
            if len(self.args) > 1:
                self.__imag = float(args[1])
            elif self.args[0] == "inf":
                self.__imag = float(self.args[0])
            else:
                self.__imag = 0

def __correct_type(self, number):

    """
    correct_type uses polymorphism to change number to the Complex
    datatype
    """

    if isinstance(number, Complex):
        return number
    elif isinstance(number, (float,int)):
        number = Complex(number)
    elif not (hasattr(number, 'real') and hasattr(number, 'imag')):
        raise TypeError('Number must have a real and imaginary part')
    else:
        raise TypeError(f'Number of type {type(number)} not of
             correct format')
    return number


def __illegal(self, op):
    """ Run when an illegal operation for complex numbers is tring
        to be computed """
    print(f'Unable to compute \"{op}\"\nThis operation is illegal
        for complex numbers')


### Arithmetic Operations ###
def __abs__(self):
    """
    abs(self)
    returns the absolute value (magnitude)
    """
    return sqrt(self.__real**2 + self.__imag**2)


def __add__(self, other):
    """ self + other """
    other = self.__correct_type(other)
    return Complex(self.__real + other.__real, self.__imag +
        other.__imag)


def __radd__(self,other):
    """ other + self """
    return self.__add__(other)


def __sub__(self, other):
    """ self - other """
    other = self.__correct_type(other)
    return Complex(self.__real - other.__real, self.__imag -
        other.__imag)


def __rsub__(self, other):
    """ other - self """
    return self.__sub__(other)


def __mul__(self, other):
    """
    self * other
    uses formula (a+bi)(c+di) = (ac-bd) + (ad+bc)i
    """
    other = self.__correct_type(other)
    return Complex(self.__real*other.__real -
        self.__imag*other.__imag,
           self.__real*other.__imag + self.__imag*other.__real)


def __rmul__(self, other):
    """ other * self """
    return self.__mul__(other)


def __truediv__(self, other):
    """ self / other """
    other = self.__correct_type(other)
    denominator = float(other.__real**2 + other.__imag**2)
    return
        Complex((self.__real*other.__real+self.__imag*other.__imag)/denominator,
           (self.__imag*other.__real-self.__real*other.__imag)/denominator)


def __rtruediv__(self, other):
    """ other / self """
    return self.__truediv__(other)


def __pow__(self, other):
    """
    self ** other
    self^other = \rho^c e^{-d\theta}(\cos(d\ln\rho + c\theta)+i
        \sin(d\ln\rho + c\theta))
    where self = a+bi, other=c+di, \theta=\arctan(\frac{b}{a}),
        \rho=sqrt{a^2 + b^2}
    """
    other = self.__correct_type(other)
    rho, theta = self.polar()
    c = other.get_real()
    d = other.get_imag()
    mod = rho ** c * exp(-d * theta)
    arg = d * log(rho) + c * theta
    return Complex(mod, arg, rect=False)


def __rpow__(self, other):
    """ other ** self """
    return self.__pow__(other)


def __eq__(self, other):
    """ self == other """
    other = self.__correct_type(other)
    return self.__real == other.__real and self.__imag ==
        other.__imag


def __ne__(self,other):
    """ self != other """
    return not(self.__eq__(other))


def __neg__(self):
    """ -self """
    return Complex(-self.__real, -self.__imag)


def __pos__(self):
    """ +self """
    return Complex(+self.__real, +self.__imag)


### Printing and Display ###
def __str__(self):
    """ str(self) """
    if self.__imag >= 0:
        return '(%s+%si)' % (self.__real, self.__imag)
    else:
        return '(%s-%si)' % (self.__real, abs(self.__imag))


def __repr__(self):
    """ repr(self) """
    return 'Complex(%s, %s)' % (self.__real, self.__imag)


### illegal operations ###
```

```
def __gt__(self, other):
    """ self > other"""
    self.__illegal(f’{self} > {other}’)


def __ge__(self, other):
    """ self >= other"""
    self.__illegal(f’{self} >= {other}’)


def __lt__(self, other):
    """ self < other"""
    self.__illegal(f’{self} < {other}’)


def __le__(self, other):
    """ self <= other"""
    self.__illegal(f’{self} <= {other}’)


### Miscellaneous Functions ###

def conjugate(self):
    """ (a+bi).conjugate() returns (a-bi) """
    return Complex(self.__real, -self.__imag)


def phase(self):
     """ self.phase() returns the argument of the complex number"""
     return atan2(self.__imag, self.__real)


def dump(self):
    """ self.dump() returns all of the functions attributes"""
    return self.__dict__


def polar(self):
    """ self.polar() returns the modulus and argument of the complex
        number"""
    return (self.__abs__(), self.phase())


def rect(self):
    """ self.rect() returns the rect coordinates of the complex
        number"""
    return (self.__real, self.__imag)

def get_real(self):
    """ self.get_real() returns the real part of the complex
```

```
            number"""
        return self.__real

    def get_imag(self):
        """ self.get_imag() returns the imaginary part of the complex
            number"""
        return self.__imag
```

program/utils/screen$_{d}esign.py$

```
"""
screen_design.py
================

Contains the Screen class
"""


from PyQt5 import QtCore, QtGui, QtWidgets
import matplotlib
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as
    FigureCanvas
from matplotlib.figure import Figure
from ..user_interface import Ui_MatPlotScreen
from .database_functions import database_query, database_insert,
    database_select, get_id
from .user import User


class Screen(QtWidgets.QDialog):

    """
    The Screen Class is inherited by all of the other classes that are
        used
    to interact witht the GUI

    The prupose of this class is to set some default values, and
        automatically
    run functions that are common to every class that inherits it

    It also contains some functions which are commonly run by classes
        that
    inherit it
    """

    def __init__(self):
        super(Screen, self).__init__()
        self.setFixedWidth(1340)
        self.setFixedHeight(720)
```

```
def goto_mainmenu(self):
    from ..main_section import MainMenu
    self.main_menu = MainMenu()
    self.hide()

def setup_question(self):
    """
    This function is only called if there is a question that the
        user can answer on that page

    This function will set the QuestionText label in the gui to the
        question that is being asked, and if the user is signed in,
        and has previously answered the question correctly, then
        their previous answer will be displayed
    """
    self.ui.SubmitButton.clicked.connect(self.check_answer)
    self.ui.QuestionText.setStyleSheet("font-size: 16pt;
        font-weight: 600;")
    self.text = database_query("SELECT Question FROM Questions WHERE
        Question_No=?", self.question_no)[0][0]
    self.correct_answers = [answer[0] for answer in
        database_query("SELECT CorrectAnswer From CorrectAnswers
        WHERE Question_No=?", self.question_no)]
    self.lowercase_correct_answers = list(map(lambda answer :
        answer.lower(), self.correct_answers))
    self.ui.QuestionText.setText(self.center_text(self.text))
    if User.GetSignedIn():
        self.usernames = [username[0] for username in
            database_query("SELECT Username FROM UsersAnswers WHERE
            Question_No=?", self.question_no)]
        if User.GetUsername() in self.usernames:
            self.users_answer = str(database_query("SELECT
                UsersAnswer FROM UsersAnswers WHERE Question_No=? AND
                Username=?", self.question_no,
                User.GetUsername())[0][0])
            if self.users_answer.lower() in
                self.lowercase_correct_answers:
                self.ui.QuestionInput.setText(self.users_answer)
                self.set_label_correct()

def check_answer(self):
    self.users_answer = self.ui.QuestionInput.text().lower()
    if self.users_answer.lower() in self.lowercase_correct_answers:
        self.set_label_correct()
    else:
        self.set_label_incorrect()
    if User.GetSignedIn():
        self.add_answer_to_db()
    def set_label_correct(self):
        self.ui.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
                "font: 18pt \"Sans Serif\";")
        self.ui.MessageLabel.setText(self.center_text('Correct!'))
        self.ui.QuestionInput.setReadOnly(True)

    def set_label_incorrect(self):
        self.ui.MessageLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
                "font: 18pt \"Sans Serif\";")
        self.ui.MessageLabel.setText(self.center_text('Incorrect, try
            again'))

    def add_answer_to_db(self):
        """
        This function is only run if the user is already signed in
        Add the User's Answer to the question to the database, whether
            it's
        right or wrong

        Will add the user's answer to UserAnswer table
        If User had already answered the question, the record in the
            database
        will need to be deleted before the new one is inserted.
        """
        database_query("DELETE FROM UsersAnswers WHERE Username=? AND
            Question_No=?", User.GetUsername(), self.question_no)
        database_insert('UsersAnswers', self.question_no,
            User.GetUsername(), self.users_answer)

    def center_text(self, text):
        return f'<html><head/><body><p
            align=\"center\">{text}</p></body></html>'


class MplWidget(Screen):
    """ A Matplotlib Widget """

    def __init__(self, parent=None):
        super(MplWidget, self).__init__()
        self.figure = Figure()
        self.canvas = FigureCanvas(self.figure)
        self.axes = self.figure.add_subplot(111)
        self.layoutvertical = QtWidgets.QVBoxLayout(self)
        self.layoutvertical.addWidget(self.canvas)


class StaticGraphScreen(Screen):

    """
    Static Graph Screen
    """

    def __init__(self):
        super(StaticGraphScreen, self).__init__()
        self.ui = Ui_MatPlotScreen()
        self.ui.setupUi(self)
        self.init_widget()
        self.x_vals = []
        self.y_vals= []

    def init_widget(self):
        self.matplotlibwidget = MplWidget()
        self.layoutvertical = QtWidgets.QVBoxLayout(self)
        self.layoutvertical.addWidget(self.matplotlibwidget)


class DynamicGraphScreen(StaticGraphScreen):

    """
    Dynamic Graph Screen
    """

    def __init__(self):
        super(DynamicGraphScreen, self).__init__()
        self.timer = QtCore.QTimer(self)
        self.timer.timeout.connect(self.update_figure)
        self.timer.start(100)
        self.count = 0

    def update_figure(self):
        self.x_vals.append(self.count)
        self.y_vals.append(self.count)
        self.matplotlibwidget.axes.cla()
        self.matplotlibwidget.axes.plot(self.x_vals, self.y_vals,
            label=f’y=x’, color=’blue’)
        self.matplotlibwidget.axes.legend(loc=’upper left’)
        self.matplotlibwidget.canvas.draw()
        self.count += 1
```

program/utils/user.py

```
"""
user.py
=======

Contains the ProgramUser class

A single instance of this class is used to store the
current user’s credentials and data
```

```
"""


class ProgramUser():

    """
    The ProgramUser class is used to store the user's credentials

    The class contains Setter and Getter methods in order to interact
        with these
    credentials throughout the runtime of the program
    """

    def __init__(self, signed_in=False, username=None, email=None):
        super(ProgramUser, self).__init__()
        self.signed_in = signed_in
        self.username = username
        self.email = email

    def SetSignedIn(self, signed_in):
        self.signed_in = signed_in

    def SetUsername(self, username):
        self.username = username

    def SetEmail(self, email):
        self.email = email

    def GetSignedIn(self):
        return self.signed_in

    def GetUsername(self):
        return self.username

    def GetEmail(self):
        return self.email


"""
This instance of the ProgramUser is imported by various other namespaces
and is used throughout the program, in order to interact with the
    ProgramUser
class
"""
User = ProgramUser()
```

program/user_interface/__init__.py

```
"""
```

```
__init__.py
===========
Imports for the user_interface
"""


from .main_menu import Ui_MainMenu
from .login_ui import Ui_ForgottenPasswordScreen,
    Ui_ForgottenPassword2Screen, Ui_LoginScreen,
    Ui_ResetPasswordScreen, Ui_ResetPassword2Screen, Ui_SignUpScreen
from .tutorial_ui import Ui_TutorialScreen,
    Ui_ProgramStructureTutorialScreen, Ui_IntroductionTutorialScreen,
    Ui_InvestigationTutorialScreen, Ui_LoginTutorialScreen,
    Ui_SummaryTutorialScreen
from .investigation_ui import Ui_PolarGraphScreen,
    Ui_PrimeCountingFunctionScreen, Ui_GraphPlotsScreen,
    Ui_ZetaZeroesPlotScreen, Ui_PrimeNumbersScreen,
    Ui_CalculatorScreen, Ui_SingleCalculatorScreen,
    Ui_TableCalculatorScreen, Ui_TableCalculator2Screen,
    Ui_ZeroesScreen, Ui_CalculateZeroesScreen,
    Ui_CalculateZeroes2Screen, Ui_CalculatorLeaderboardScreen,
    Ui_ZetaApproximationScreen
from .introduction_ui import Ui_IntroductionScreen,
    Ui_HistoricalBackgroundScreen, Ui_WhatIsTheRiemannHypothesisScreen,
    Ui_PracticalApplicationsScreen
from .summary_ui import Ui_SummaryScreen, Ui_TheoryRecapScreen,
    Ui_InvestigationResultsScreen, Ui_ConclusionScreen, Ui_ImpactScreen
from .notes_ui import Ui_TutorialNotesScreen,
    Ui_IntroductionNotesScreen, Ui_InvestigationNotesScreen,
    Ui_SummaryNotesScreen
from .mat_plot import Ui_MatPlotScreen
from .progress import Ui_ProgressScreen
```

program/user_interface/main_menu.py

```
"""
main_menu.py
============
A GUI for the main menu page
"""

from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_MainMenu(object):

    def setupUi(self, MainMenu):
        MainMenu.setObjectName("MainMenu")
        MainMenu.setEnabled(True)
        MainMenu.resize(1340, 720)
        sizePolicy =
            QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding,
            QtWidgets.QSizePolicy.Expanding)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(MainMenu.sizePolicy().hasHeightForWidth())
        MainMenu.setSizePolicy(sizePolicy)
        self.MainWidget = QtWidgets.QWidget(MainMenu)
        self.MainWidget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        sizePolicy =
            QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding,
            QtWidgets.QSizePolicy.Expanding)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(self.MainWidget.sizePolicy().hasHeightForWidth())
        self.MainWidget.setSizePolicy(sizePolicy)
        self.MainWidget.setObjectName("MainWidget")
        self.SideWidget = QtWidgets.QWidget(self.MainWidget)
        self.SideWidget.setGeometry(QtCore.QRect(-50, 0, 512, 720))
        self.SideWidget.setStyleSheet("background-color:rgb(69, 69, 69)
            ;\n"
"border-radius:40px;")
        self.SideWidget.setObjectName("SideWidget")
        self.Title = QtWidgets.QLabel(self.SideWidget)
        self.Title.setGeometry(QtCore.QRect(70, 30, 421, 121))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.InvestigationButton = QtWidgets.QPushButton(self.SideWidget)
        self.InvestigationButton.setGeometry(QtCore.QRect(170, 410, 231,
            61))
        self.InvestigationButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationButton.setStyleSheet("background-color:
            rgb(239, 239, 239);\n"
"border-radius:20px;\n"
"font: 18pt \"Sans Serif\";")
        self.InvestigationButton.setObjectName("InvestigationButton")
        self.LogInButton = QtWidgets.QPushButton(self.SideWidget)
        self.LogInButton.setGeometry(QtCore.QRect(170, 170, 231, 61))
        self.LogInButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LogInButton.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius:20px;\n"
"font: 18pt \"Sans Serif\";")
        self.LogInButton.setCheckable(False)
        self.LogInButton.setChecked(False)
        self.LogInButton.setObjectName("LogInButton")
        self.TutorialButton = QtWidgets.QPushButton(self.SideWidget)
        self.TutorialButton.setGeometry(QtCore.QRect(170, 250, 231, 61))
        self.TutorialButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialButton.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"border-radius:20px;\n"
"font: 18pt \"Sans Serif\";")
        self.TutorialButton.setObjectName("TutorialButton")
        self.IntroductionButton = QtWidgets.QPushButton(self.SideWidget)
        self.IntroductionButton.setGeometry(QtCore.QRect(170, 330, 231,
            61))
        self.IntroductionButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionButton.setStyleSheet("background-color:
            rgb(239, 239, 239);\n"
"border-radius:20px;\n"
"font: 18pt \"Sans Serif\";")
        self.IntroductionButton.setObjectName("IntroductionButton")
        self.SummaryButton = QtWidgets.QPushButton(self.SideWidget)
        self.SummaryButton.setGeometry(QtCore.QRect(170, 490, 231, 61))
        self.SummaryButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryButton.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"border-radius:20px;\n"
"font: 18pt \"Sans Serif\";")
        self.SummaryButton.setObjectName("SummaryButton")
        self.ExitButton = QtWidgets.QPushButton(self.SideWidget)
        self.ExitButton.setGeometry(QtCore.QRect(170, 570, 231, 61))
        self.ExitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ExitButton.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius:20px;\n"
"font: 18pt \"Sans Serif\";")
        self.ExitButton.setObjectName("ExitButton")
        self.BackgroundImage = QtWidgets.QLabel(self.MainWidget)
        self.BackgroundImage.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        sizePolicy =
            QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Expanding,
            QtWidgets.QSizePolicy.Expanding)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(self.BackgroundImage.sizePolicy().hasHeightForWidth())
        self.BackgroundImage.setSizePolicy(sizePolicy)
        self.BackgroundImage.setText("")
        self.BackgroundImage.setPixmap(QtGui.QPixmap("ui/../media/zeta-graph.jpg"))
        self.BackgroundImage.setScaledContents(True)
        self.BackgroundImage.setObjectName("BackgroundImage")
        self.UsernameButton = QtWidgets.QPushButton(self.MainWidget)
        self.UsernameButton.setGeometry(QtCore.QRect(1140, 20, 180, 50))
        self.UsernameButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.UsernameButton.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"border-radius:20px;\n"
"font: 15pt \"Sans Serif\";\n"
"padding:3px;")
        self.UsernameButton.setObjectName("UsernameButton")
        self.BackgroundImage.raise_()
        self.SideWidget.raise_()
        self.UsernameButton.raise_()

        self.retranslateUi(MainMenu)
        QtCore.QMetaObject.connectSlotsByName(MainMenu)

    def retranslateUi(self, MainMenu):
        _translate = QtCore.QCoreApplication.translate
        MainMenu.setWindowTitle(_translate("MainMenu", "Visualing The
            Riemann Hypothesis - Main Menu"))
        self.Title.setText(_translate("MainMenu", "<html><head/><body><p
            align=\"center\"><span style=\" font-size:28pt;
            font-weight:600;\">Visualising The</span></p><p
            align=\"center\"><span style=\" font-size:28pt;
            font-weight:600;\">Riemann
            Hypothesis</span></p></body></html>"))
        self.InvestigationButton.setText(_translate("MainMenu",
            "Investigation"))
        self.LogInButton.setText(_translate("MainMenu", "Log In"))
        self.TutorialButton.setText(_translate("MainMenu", "Tutorial"))
        self.IntroductionButton.setText(_translate("MainMenu",
            "Introduction"))
        self.SummaryButton.setText(_translate("MainMenu", "Summary"))
        self.ExitButton.setText(_translate("MainMenu", "Exit"))
        self.UsernameButton.setText(_translate("MainMenu", "Username"))
```

program/user_interface/mat_plot.py

```
"""
mat_plot.py
============
A GUI for every mat plot graph page
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_MatPlotScreen(object):

    def setupUi(self, MatPlotScreen):
        MatPlotScreen.setObjectName("MatPlotScreen")
        MatPlotScreen.resize(1340, 720)
        MatPlotScreen.setSizeGripEnabled(False)
        self.retranslateUi(MatPlotScreen)
        QtCore.QMetaObject.connectSlotsByName(MatPlotScreen)

    def retranslateUi(self, MatPlotScreen):
        _translate = QtCore.QCoreApplication.translate
        MatPlotScreen.setWindowTitle(_translate("MatPlotScreen",
            "Visualising the Riemann Hypothesis - Zeta Zeroes"))
```

program/user_interface/progress.py

```
"""
progress.py
============
A GUI for the progress page
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ProgressScreen(object):

    def setupUi(self, ProgressScreen):
        ProgressScreen.setObjectName("ProgressScreen")
        ProgressScreen.resize(1340, 723)
        ProgressScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ProgressScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(575, 20, 191, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.ProgressTab = QtWidgets.QPushButton(self.TabBar)
        self.ProgressTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.ProgressTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ProgressTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
```

```
        self.ProgressTab.setObjectName("ProgressTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239, 239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.BackButton = QtWidgets.QPushButton(self.MainWidget)
        self.BackButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.BackButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.BackButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.BackButton.setObjectName("BackButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 251, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(410, 470, 530, 41))
        self.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.MessageLabel.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.MessageLabel.setObjectName("MessageLabel")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")
        self.Table = QtWidgets.QTableWidget(self.MainWidget)
        self.Table.setGeometry(QtCore.QRect(30, 70, 1261, 371))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.Table.setFont(font)
        self.Table.setSizeAdjustPolicy(QtWidgets.QAbstractScrollArea.AdjustToContents)
        self.Table.setEditTriggers(QtWidgets.QAbstractItemView.NoEditTriggers)
        self.Table.setAlternatingRowColors(True)
        self.Table.setSelectionMode(QtWidgets.QAbstractItemView.NoSelection)
        self.Table.setObjectName("Table")
        self.Table.setColumnCount(3)
        self.Table.setRowCount(0)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.Table.setHorizontalHeaderItem(0, item)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.Table.setHorizontalHeaderItem(1, item)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.Table.setHorizontalHeaderItem(2, item)

        self.retranslateUi(ProgressScreen)
        QtCore.QMetaObject.connectSlotsByName(ProgressScreen)

    def retranslateUi(self, ProgressScreen):
        _translate = QtCore.QCoreApplication.translate
        ProgressScreen.setWindowTitle(_translate("ProgressScreen",
            "Visualising the Riemann Hypothesis - Progress"))
        self.Title.setText(_translate("ProgressScreen", "Progress"))
        self.ProgressTab.setText(_translate("ProgressScreen",
            "Progress"))
        self.BackButton.setText(_translate("ProgressScreen", "Back"))
        self.SubTitleText.setText(_translate("ProgressScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Username</span></p></body></html>"))
        self.MessageLabel.setText(_translate("ProgressScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        self.NotesButton.setText(_translate("ProgressScreen", "Notes"))
        item = self.Table.horizontalHeaderItem(0)
        item.setText(_translate("ProgressScreen", "Question"))
        item = self.Table.horizontalHeaderItem(1)
        item.setText(_translate("ProgressScreen", "Answer"))
        item = self.Table.horizontalHeaderItem(2)
        item.setText(_translate("ProgressScreen", "Correct"))
```

program/user_interface/introduction_ui/__init__.py

```
"""
__init__.py
===========
Imports for the introduction_ui
"""


from .introduction import Ui_IntroductionScreen
from .historical_background import Ui_HistoricalBackgroundScreen
from .what_is_the_riemann_hypothesis import
    Ui_WhatIsTheRiemannHypothesisScreen
from .practical_applications import Ui_PracticalApplicationsScreen
```

program/user_interface/introduction_ui/historical_background.py

```
"""
historical_background.py
========================
A GUI for the historical background page of the introduction section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_HistoricalBackgroundScreen(object):

    def setupUi(self, HistoricalBackgroundScreen):
        HistoricalBackgroundScreen.setObjectName("HistoricalBackgroundScreen")
        HistoricalBackgroundScreen.resize(1340, 723)
        HistoricalBackgroundScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(HistoricalBackgroundScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(545, 20, 251, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.HistoricalBackgroundLabel = QtWidgets.QLabel(self.TabBar)
        self.HistoricalBackgroundLabel.setGeometry(QtCore.QRect(220, 5,
            200, 70))
        self.HistoricalBackgroundLabel.setStyleSheet("border: 2px
            solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
               "")
        self.HistoricalBackgroundLabel.setObjectName("HistoricalBackgroundLabel")
        self.HistoricalBackgroundTab = QtWidgets.QPushButton(self.TabBar)
        self.HistoricalBackgroundTab.setGeometry(QtCore.QRect(220, 5,
            200, 70))
        self.HistoricalBackgroundTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.HistoricalBackgroundTab.setStyleSheet("border-radius:
            20px;\n"
               "background-color: rgba(0, 0, 0, 0);\n"
               "font: 18pt \"Sans Serif\";\n"
               "")
        self.HistoricalBackgroundTab.setText("")
        self.HistoricalBackgroundTab.setObjectName("HistoricalBackgroundTab")
        self.WhatIsTheRHTab = QtWidgets.QPushButton(self.TabBar)
        self.WhatIsTheRHTab.setGeometry(QtCore.QRect(430, 5, 240, 70))
        self.WhatIsTheRHTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.WhatIsTheRHTab.setStyleSheet("border-radius: 20px;\n"
               "background-color: rgba(0, 0, 0, 0);\n"
               "font: 18pt \"Sans Serif\";\n"
               "")
        self.WhatIsTheRHTab.setText("")
        self.WhatIsTheRHTab.setObjectName("WhatIsTheRHTab")
        self.WhatIsTheRHLabel = QtWidgets.QLabel(self.TabBar)
        self.WhatIsTheRHLabel.setGeometry(QtCore.QRect(430, 5, 241, 70))
        self.WhatIsTheRHLabel.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.WhatIsTheRHLabel.setObjectName("WhatIsTheRHLabel")
        self.PracticalApplicationsLabel = QtWidgets.QLabel(self.TabBar)
        self.PracticalApplicationsLabel.setGeometry(QtCore.QRect(680, 5,
            200, 70))
        self.PracticalApplicationsLabel.setStyleSheet("border: 2px
            solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.PracticalApplicationsLabel.setObjectName("PracticalApplicationsLabel")
        self.PracticalApplicationsTab =
            QtWidgets.QPushButton(self.TabBar)
        self.PracticalApplicationsTab.setGeometry(QtCore.QRect(680, 5,
            201, 70))
        self.PracticalApplicationsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PracticalApplicationsTab.setStyleSheet("border-radius:
            20px;\n"
               "background-color: rgba(0, 0, 0, 0);\n"
               "font: 18pt \"Sans Serif\";\n"
               "")
        self.PracticalApplicationsTab.setText("")
        self.PracticalApplicationsTab.setObjectName("PracticalApplicationsTab")
        self.WhatIsTheRHLabel.raise_()
        self.IntroductionTab.raise_()
        self.HistoricalBackgroundLabel.raise_()
        self.HistoricalBackgroundTab.raise_()
        self.WhatIsTheRHTab.raise_()
        self.PracticalApplicationsLabel.raise_()
        self.PracticalApplicationsTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
               "border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 381, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 60, 1251, 401))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.NotesButton.setObjectName("NotesButton")
        self.SubTitleText.raise_()
        self.MainText.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.NotesButton.raise_()

        self.retranslateUi(HistoricalBackgroundScreen)
        QtCore.QMetaObject.connectSlotsByName(HistoricalBackgroundScreen)

    def retranslateUi(self, HistoricalBackgroundScreen):
        _translate = QtCore.QCoreApplication.translate
        HistoricalBackgroundScreen.setWindowTitle(_translate("HistoricalBackgroundScreen",
            "Visualising the Riemann Hypothesis - Introduction"))
        self.Title.setText(_translate("HistoricalBackgroundScreen",
            "Introduction"))
        self.IntroductionTab.setText(_translate("HistoricalBackgroundScreen",
            "Introduction"))
        self.HistoricalBackgroundLabel.setText(_translate("HistoricalBackgroundScreen",
            "<html><head/><body><p
            align=\"center\">Historical<br>Background</p></body></html>"))
        self.WhatIsTheRHLabel.setText(_translate("HistoricalBackgroundScreen",
            "<html><head/><body><p align=\"center\">What is
            the<br>Riemann Hypothesis</p></body></html>"))
        self.PracticalApplicationsLabel.setText(_translate("HistoricalBackgroundScreen",
            "<html><head/><body><p
            align=\"center\">Practical<br>Applications</p></body></html>"))
        self.PrevButton.setText(_translate("HistoricalBackgroundScreen",
            "Prev"))
        self.NextButton.setText(_translate("HistoricalBackgroundScreen",
            "Next"))
        self.SubTitleText.setText(_translate("HistoricalBackgroundScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Historical
            Background</span></p></body></html>"))
        self.MainText.setText(_translate("HistoricalBackgroundScreen",
            "<html><head/><body><p>The Riemann Hypothesis originates
            from a mathematical problem from hundreds of years back. In
            the 1730\'s, Leonhard Euler (A very smart mathematician),
            has been investigating the function that went on to be known
            as the Riemann Zeta Function. In 1737, Euler was able to
            prove that this function could be written as a product of
            prime numbers, this is known as Euler\'s product
            formula.</p><p>In 1859, Bernhard Riemann publsihed a paper
            \'On the Number of Primes Less Than a Given Magnitude\'. In
            this paper, he investigates the function previously looked
            at by Euler, but extends its definition to include Complex
            Numbers, as well as Real Numbers. It is in this paper that
            the Riemann Hypothesis was born. The Riemann Hypothesis
            states that the Riemann zeta function has its zeros only at
            the negative even integers and complex numbers with real
            part 0.5.</p><p>In 1900, the mathematician David Hibert
            released a list of mathematical problems that were all
            unsolved at the time. He called for other mathematicians to
            try and solve these problems. These problems included the a
            Proof of the Riemann Hypothesis, and many problems relating
            to it; many of which are still unsolved to this day</p><p>In
            1915, GH Hardy managed to prove that there were an inifinite
            amount of numbers for which the Riemann Hypothesis was true,
            this showed that proving the Riemann Hypothesis would
            require a full proof, rather than a proof by exhaustion.
            </p><p>In 1986 van de Lune, te Riele &amp; Winter managed to
            find values for 1500000001 Zeta Zeroes.</p><p>Then in 2000,
            the Clay Mathematics Institute named the Riemann Hypothesis
            as one of their \'Millenium Problems\', with a reward of
            $1,000,000 for anyone who manages to prove
            it.</p></body></html>"))
        self.NotesButton.setText(_translate("HistoricalBackgroundScreen",
            "Notes"))
```

program/user_interface/introduction_ui/introduction.py

```
"""
introduction.py
===============
A GUI for the introduction page of the introduction section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_IntroductionScreen(object):

    def setupUi(self, IntroductionScreen):
        IntroductionScreen.setObjectName("IntroductionScreen")
        IntroductionScreen.resize(1340, 723)
        IntroductionScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(IntroductionScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(545, 20, 251, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
              "border-radius: 20px;\n"
              "border-color:rgb(69, 69, 69);\n"
              "background-color: rgb(239, 239, 239);\n"
              "font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
              "")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.HistoricalBackgroundLabel = QtWidgets.QLabel(self.TabBar)
        self.HistoricalBackgroundLabel.setGeometry(QtCore.QRect(220, 5,
            200, 70))
        self.HistoricalBackgroundLabel.setStyleSheet("border: 2px
            solid;\n"
              "border-radius: 20px;\n"
              "border-color:rgb(69, 69, 69);\n"
              "background-color: rgb(69, 69, 69);\n"
              "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
```

```
       "")
self.HistoricalBackgroundLabel.setObjectName("HistoricalBackgroundLabel")
self.HistoricalBackgroundTab = QtWidgets.QPushButton(self.TabBar)
self.HistoricalBackgroundTab.setGeometry(QtCore.QRect(220, 5,
    200, 70))
self.HistoricalBackgroundTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.HistoricalBackgroundTab.setStyleSheet("border-radius:
    20px;\n"
       "background-color: rgba(0, 0, 0, 0);\n"
       "font: 18pt \"Sans Serif\";\n"
       "")
self.HistoricalBackgroundTab.setText("")
self.HistoricalBackgroundTab.setObjectName("HistoricalBackgroundTab")
self.WhatIsTheRHTab = QtWidgets.QPushButton(self.TabBar)
self.WhatIsTheRHTab.setGeometry(QtCore.QRect(430, 5, 240, 70))
self.WhatIsTheRHTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.WhatIsTheRHTab.setStyleSheet("border-radius: 20px;\n"
       "background-color: rgba(0, 0, 0, 0);\n"
       "font: 18pt \"Sans Serif\";\n"
       "")
self.WhatIsTheRHTab.setText("")
self.WhatIsTheRHTab.setObjectName("WhatIsTheRHTab")
self.WhatIsTheRHLabel = QtWidgets.QLabel(self.TabBar)
self.WhatIsTheRHLabel.setGeometry(QtCore.QRect(430, 5, 241, 70))
self.WhatIsTheRHLabel.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.WhatIsTheRHLabel.setObjectName("WhatIsTheRHLabel")
self.PracticalApplicationsLabel = QtWidgets.QLabel(self.TabBar)
self.PracticalApplicationsLabel.setGeometry(QtCore.QRect(680, 5,
    200, 70))
self.PracticalApplicationsLabel.setStyleSheet("border: 2px
    solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.PracticalApplicationsLabel.setObjectName("PracticalApplicationsLabel")
self.PracticalApplicationsTab =
    QtWidgets.QPushButton(self.TabBar)
self.PracticalApplicationsTab.setGeometry(QtCore.QRect(680, 5,
    201, 70))
self.PracticalApplicationsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.PracticalApplicationsTab.setStyleSheet("border-radius:
    20px;\n"
       "background-color: rgba(0, 0, 0, 0);\n"
       "font: 18pt \"Sans Serif\";\n"
       "")
self.PracticalApplicationsTab.setText("")
self.PracticalApplicationsTab.setObjectName("PracticalApplicationsTab")
self.WhatIsTheRHLabel.raise_()
self.IntroductionTab.raise_()
self.HistoricalBackgroundLabel.raise_()
self.HistoricalBackgroundTab.raise_()
self.WhatIsTheRHTab.raise_()
self.PracticalApplicationsLabel.raise_()
self.PracticalApplicationsTab.raise_()
self.MainWidget = QtWidgets.QWidget(self.widget)
self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
    239);\n"
       "border-radius: 20px;")
self.MainWidget.setObjectName("MainWidget")
self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.PrevButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.PrevButton.setObjectName("PrevButton")
self.NextButton = QtWidgets.QPushButton(self.MainWidget)
self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.NextButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.NextButton.setObjectName("NextButton")
self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 381, 41))
self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.SubTitleText.setObjectName("SubTitleText")
self.MainText = QtWidgets.QLabel(self.MainWidget)
self.MainText.setGeometry(QtCore.QRect(40, 60, 1251, 401))
self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.MainText.setWordWrap(True)
self.MainText.setObjectName("MainText")
```

```
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.NotesButton.setObjectName("NotesButton")
        self.SubTitleText.raise_()
        self.MainText.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.NotesButton.raise_()

        self.retranslateUi(IntroductionScreen)
        QtCore.QMetaObject.connectSlotsByName(IntroductionScreen)

    def retranslateUi(self, IntroductionScreen):
        _translate = QtCore.QCoreApplication.translate
        IntroductionScreen.setWindowTitle(_translate("IntroductionScreen",
            "Visualising the Riemann Hypothesis - Introduction"))
        self.Title.setText(_translate("IntroductionScreen",
            "Introduction"))
        self.IntroductionTab.setText(_translate("IntroductionScreen",
            "Introduction"))
        self.HistoricalBackgroundLabel.setText(_translate("IntroductionScreen",
            "<html><head/><body><p
            align=\"center\">Historical<br>Background</p></body></html>"))
        self.WhatIsTheRHLabel.setText(_translate("IntroductionScreen",
            "<html><head/><body><p align=\"center\">What is
            the<br>Riemann Hypothesis</p></body></html>"))
        self.PracticalApplicationsLabel.setText(_translate("IntroductionScreen",
            "<html><head/><body><p
            align=\"center\">Practical<br>Applications</p></body></html>"))
        self.PrevButton.setText(_translate("IntroductionScreen", "Prev"))
        self.NextButton.setText(_translate("IntroductionScreen", "Next"))
        self.SubTitleText.setText(_translate("IntroductionScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Introduction</span></p></body></html>"))
        self.MainText.setText(_translate("IntroductionScreen",
            "<html><head/><body><p>This is the introduction section of
            the program. The purpose of this section is to teach you
            about the Riemann Hypothesis.</p><p><br/></p><p>This section
            will cover some important theory behind the Hypothesis, such
            as the Historical Background behind the problem, What the
            Riemann Hypothesis actually is, and the Practical
            Applications of the hypothesis - why it is actually
            important.</p><p><br/></p><p>There is a lot of fundamental
            theory in this section that you should know to be able to
            use this progrma to it\'s full extent, so make sure that you
            keep on clicking the notes buttons on each screen so that
            you can record useful information.</p></body></html>"))
        self.NotesButton.setText(_translate("IntroductionScreen",
            "Notes"))
```

program/user_interface/introduction_ui/practical_applications.py

```
"""
practical_applications.py
=========================
A GUI for the practical applications page of the introduction section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_PracticalApplicationsScreen(object):

    def setupUi(self, PracticalApplicationsScreen):
        PracticalApplicationsScreen.setObjectName("PracticalApplicationsScreen")
        PracticalApplicationsScreen.resize(1340, 723)
        PracticalApplicationsScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(PracticalApplicationsScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(545, 20, 251, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.HistoricalBackgroundLabel = QtWidgets.QLabel(self.TabBar)
        self.HistoricalBackgroundLabel.setGeometry(QtCore.QRect(220, 5,
            200, 70))
        self.HistoricalBackgroundLabel.setStyleSheet("border: 2px
            solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.HistoricalBackgroundLabel.setObjectName("HistoricalBackgroundLabel")
        self.HistoricalBackgroundTab = QtWidgets.QPushButton(self.TabBar)
        self.HistoricalBackgroundTab.setGeometry(QtCore.QRect(220, 5,
            200, 70))
        self.HistoricalBackgroundTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.HistoricalBackgroundTab.setStyleSheet("border-radius:
            20px;\n"
               "background-color: rgba(0, 0, 0, 0);\n"
               "font: 18pt \"Sans Serif\";\n"
               "")
        self.HistoricalBackgroundTab.setText("")
        self.HistoricalBackgroundTab.setObjectName("HistoricalBackgroundTab")
        self.WhatIsTheRHTab = QtWidgets.QPushButton(self.TabBar)
        self.WhatIsTheRHTab.setGeometry(QtCore.QRect(430, 5, 240, 70))
        self.WhatIsTheRHTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.WhatIsTheRHTab.setStyleSheet("border-radius: 20px;\n"
               "background-color: rgba(0, 0, 0, 0);\n"
               "font: 18pt \"Sans Serif\";\n"
               "")
        self.WhatIsTheRHTab.setText("")
        self.WhatIsTheRHTab.setObjectName("WhatIsTheRHTab")
        self.WhatIsTheRHLabel = QtWidgets.QLabel(self.TabBar)
        self.WhatIsTheRHLabel.setGeometry(QtCore.QRect(430, 5, 241, 70))
        self.WhatIsTheRHLabel.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.WhatIsTheRHLabel.setObjectName("WhatIsTheRHLabel")
        self.PracticalApplicationsLabel = QtWidgets.QLabel(self.TabBar)
        self.PracticalApplicationsLabel.setGeometry(QtCore.QRect(680, 5,
            200, 70))
        self.PracticalApplicationsLabel.setStyleSheet("border: 2px
            solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
               "")
```

```
self.PracticalApplicationsLabel.setObjectName("PracticalApplicationsLabel")
self.PracticalApplicationsTab =
    QtWidgets.QPushButton(self.TabBar)
self.PracticalApplicationsTab.setGeometry(QtCore.QRect(680, 5,
    201, 70))
self.PracticalApplicationsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.PracticalApplicationsTab.setStyleSheet("border-radius:
    20px;\n"
       "background-color: rgba(0, 0, 0, 0);\n"
       "font: 18pt \"Sans Serif\";\n"
       "")
self.PracticalApplicationsTab.setText("")
self.PracticalApplicationsTab.setObjectName("PracticalApplicationsTab")
self.WhatIsTheRHLabel.raise_()
self.IntroductionTab.raise_()
self.HistoricalBackgroundLabel.raise_()
self.HistoricalBackgroundTab.raise_()
self.WhatIsTheRHTab.raise_()
self.PracticalApplicationsLabel.raise_()
self.PracticalApplicationsTab.raise_()
self.MainWidget = QtWidgets.QWidget(self.widget)
self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
    239);\n"
       "border-radius: 20px;")
self.MainWidget.setObjectName("MainWidget")
self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.PrevButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.PrevButton.setObjectName("PrevButton")
self.NextButton = QtWidgets.QPushButton(self.MainWidget)
self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.NextButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.NextButton.setObjectName("NextButton")
self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 561, 41))
self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
```

```
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.SubTitleText.setObjectName("SubTitleText")
self.MainText = QtWidgets.QLabel(self.MainWidget)
self.MainText.setGeometry(QtCore.QRect(40, 60, 1251, 401))
self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
self.MainText.setWordWrap(True)
self.MainText.setObjectName("MainText")
self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.NotesButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.NotesButton.setObjectName("NotesButton")
self.QuestionText = QtWidgets.QLabel(self.MainWidget)
self.QuestionText.setGeometry(QtCore.QRect(390, 290, 561, 51))
self.QuestionText.setStyleSheet("font: 13pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.QuestionText.setWordWrap(True)
self.QuestionText.setObjectName("QuestionText")
self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
self.QuestionInput.setGeometry(QtCore.QRect(410, 350, 231, 60))
self.QuestionInput.setStyleSheet("background-color: rgb(239,
    239, 239);\n"
       "color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\";\n"
       "border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);")
self.QuestionInput.setText("")
self.QuestionInput.setEchoMode(QtWidgets.QLineEdit.Normal)
self.QuestionInput.setCursorPosition(0)
self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
self.QuestionInput.setObjectName("QuestionInput")
self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
self.SubmitButton.setGeometry(QtCore.QRect(690, 350, 131, 60))
self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.SubmitButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
```

```
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(440, 410, 461, 41))
        self.MessageLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
               "font: 18pt \"Sans Serif\";")
        self.MessageLabel.setObjectName("MessageLabel")
        self.SubTitleText.raise_()
        self.MainText.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.NotesButton.raise_()
        self.QuestionInput.raise_()
        self.SubmitButton.raise_()
        self.QuestionText.raise_()
        self.MessageLabel.raise_()

        self.retranslateUi(PracticalApplicationsScreen)
        QtCore.QMetaObject.connectSlotsByName(PracticalApplicationsScreen)

    def retranslateUi(self, PracticalApplicationsScreen):
        _translate = QtCore.QCoreApplication.translate
        PracticalApplicationsScreen.setWindowTitle(_translate("PracticalApplicationsScreen",
            "Visualising the Riemann Hypothesis - Introduction"))
        self.Title.setText(_translate("PracticalApplicationsScreen",
            "Introduction"))
        self.IntroductionTab.setText(_translate("PracticalApplicationsScreen",
            "Introduction"))
        self.HistoricalBackgroundLabel.setText(_translate("PracticalApplicationsScreen",
            "<html><head/><body><p
            align=\"center\">Historical<br>Background</p></body></html>"))
        self.WhatIsTheRHLabel.setText(_translate("PracticalApplicationsScreen",
            "<html><head/><body><p align=\"center\">What is
            the<br>Riemann Hypothesis</p></body></html>"))
        self.PracticalApplicationsLabel.setText(_translate("PracticalApplicationsScreen",
            "<html><head/><body><p
            align=\"center\">Practical<br>Applications</p></body></html>"))
        self.PrevButton.setText(_translate("PracticalApplicationsScreen",
            "Prev"))
        self.NextButton.setText(_translate("PracticalApplicationsScreen",
            "Next"))
        self.SubTitleText.setText(_translate("PracticalApplicationsScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Practical
            Applications</span></p></body></html>"))
        self.MainText.setText(_translate("PracticalApplicationsScreen",
            "<html><head/><body><p>Although the Riemann Hypothesis uses
            a lot of theoretical mathematics, that isn\'t to say that it
            doesnt have any practical applications.</p><p>If the Riemann
            Hypothesis was proven to be true, then that would mean that
            many of theories and conjectures would alwasys be true. For
            example: The weak Goldbach conjecture - stating that all
            integers greater than 5 are the sum of three primes; Mills
            constants - numbers that allow you to generate prime
            numbers, The theory that there will always be at least one
            prime between consecutive cubes; and the theory that there
            is a maximum bound between consecutive prime
            numbers.</p><p>All of these conjectures involve prime
            numbers, and their distribution. If the Riemann Hypothesis
            and thus these conjectures were true, then very large prime
            numbers would be very easy to generate. This would make
            fields such as crypotography - that heavily rely on large
            prime numbers being hard to compute - change. Current
            crypotgraphy algorithms would become obsolete and would have
            to be replaced with more secure ones.</p><p>The Riemann
            Hypothesis also has a very interesting correlation to
            quantum physics. It was discovered in 1996 that the
            arrangement of the zeta zeroes exhibits the same pattern as
            the possible values of energy in a quantum chaotic system.
            </p></body></html>"))
        self.NotesButton.setText(_translate("PracticalApplicationsScreen",
            "Notes"))
        self.QuestionText.setText(_translate("PracticalApplicationsScreen",
            "<html><head/><body><p><br/></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("PracticalApplicationsScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("PracticalApplicationsScreen",
            "Submit"))
        self.MessageLabel.setText(_translate("PracticalApplicationsScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/introduction_ui/what_is_the_riemann_hypothesis.py

```
"""
what_is_the_riemann_hypothesis.py
=================================
A GUI for the what is the riemann hypothesis page of the introduction
    section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_WhatIsTheRiemannHypothesisScreen(object):

    def setupUi(self, WhatIsTheRiemannHypothesisScreen):
        WhatIsTheRiemannHypothesisScreen.setObjectName("WhatIsTheRiemannHypothesisScreen")
        WhatIsTheRiemannHypothesisScreen.resize(1340, 723)
        WhatIsTheRiemannHypothesisScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(WhatIsTheRiemannHypothesisScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(545, 20, 251, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.HistoricalBackgroundLabel = QtWidgets.QLabel(self.TabBar)
        self.HistoricalBackgroundLabel.setGeometry(QtCore.QRect(220, 5,
            200, 70))
        self.HistoricalBackgroundLabel.setStyleSheet("border: 2px
            solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.HistoricalBackgroundLabel.setObjectName("HistoricalBackgroundLabel")
        self.HistoricalBackgroundTab = QtWidgets.QPushButton(self.TabBar)
        self.HistoricalBackgroundTab.setGeometry(QtCore.QRect(220, 5,
            200, 70))
        self.HistoricalBackgroundTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.HistoricalBackgroundTab.setStyleSheet("border-radius:
            20px;\n"
               "background-color: rgba(0, 0, 0, 0);\n"
               "font: 18pt \"Sans Serif\";\n"
               "")
        self.HistoricalBackgroundTab.setText("")
        self.HistoricalBackgroundTab.setObjectName("HistoricalBackgroundTab")
        self.WhatIsTheRHTab = QtWidgets.QPushButton(self.TabBar)
        self.WhatIsTheRHTab.setGeometry(QtCore.QRect(430, 5, 240, 70))
        self.WhatIsTheRHTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.WhatIsTheRHTab.setStyleSheet("border-radius: 20px;\n"
               "background-color: rgba(0, 0, 0, 0);\n"
               "font: 18pt \"Sans Serif\";\n"
               "")
        self.WhatIsTheRHTab.setText("")
        self.WhatIsTheRHTab.setObjectName("WhatIsTheRHTab")
        self.WhatIsTheRHLabel = QtWidgets.QLabel(self.TabBar)
        self.WhatIsTheRHLabel.setGeometry(QtCore.QRect(430, 5, 241, 70))
        self.WhatIsTheRHLabel.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
               "")
        self.WhatIsTheRHLabel.setObjectName("WhatIsTheRHLabel")
        self.PracticalApplicationsLabel = QtWidgets.QLabel(self.TabBar)
        self.PracticalApplicationsLabel.setGeometry(QtCore.QRect(680, 5,
            200, 70))
        self.PracticalApplicationsLabel.setStyleSheet("border: 2px
            solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.PracticalApplicationsLabel.setObjectName("PracticalApplicationsLabel")
        self.PracticalApplicationsTab =
            QtWidgets.QPushButton(self.TabBar)
        self.PracticalApplicationsTab.setGeometry(QtCore.QRect(680, 5,
            201, 70))
        self.PracticalApplicationsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PracticalApplicationsTab.setStyleSheet("border-radius:
            20px;\n"
               "background-color: rgba(0, 0, 0, 0);\n"
               "font: 18pt \"Sans Serif\";\n"
               "")
        self.PracticalApplicationsTab.setText("")
        self.PracticalApplicationsTab.setObjectName("PracticalApplicationsTab")
        self.WhatIsTheRHLabel.raise_()
        self.IntroductionTab.raise_()
        self.HistoricalBackgroundLabel.raise_()
        self.HistoricalBackgroundTab.raise_()
        self.WhatIsTheRHTab.raise_()
        self.PracticalApplicationsLabel.raise_()
        self.PracticalApplicationsTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
               "border-radius: 20px;")
```

```
self.MainWidget.setObjectName("MainWidget")
self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.PrevButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.PrevButton.setObjectName("PrevButton")
self.NextButton = QtWidgets.QPushButton(self.MainWidget)
self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.NextButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.NextButton.setObjectName("NextButton")
self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 561, 41))
self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.SubTitleText.setObjectName("SubTitleText")
self.MainText = QtWidgets.QLabel(self.MainWidget)
self.MainText.setGeometry(QtCore.QRect(40, 60, 1251, 281))
self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
self.MainText.setWordWrap(True)
self.MainText.setObjectName("MainText")
self.QuestionText = QtWidgets.QLabel(self.MainWidget)
self.QuestionText.setGeometry(QtCore.QRect(390, 360, 561, 31))
self.QuestionText.setStyleSheet("font: 13pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.QuestionText.setWordWrap(True)
self.QuestionText.setObjectName("QuestionText")
self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
self.QuestionInput.setGeometry(QtCore.QRect(540, 400, 101, 60))
self.QuestionInput.setStyleSheet("background-color: rgb(239,
    239, 239);\n"
       "color: rgb(69, 69, 69);\n"
       "font: 18pt \"Sans Serif\";\n"
       "border: 2px solid;\n"
       "border-radius: 20px;\n"
```

```
               "border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setEchoMode(QtWidgets.QLineEdit.Normal)
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(690, 400, 131, 60))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(440, 480, 461, 41))
        self.MessageLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
               "font: 18pt \"Sans Serif\";")
        self.MessageLabel.setObjectName("MessageLabel")
        self.SubTitleText.raise_()
        self.MainText.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.QuestionText.raise_()
        self.QuestionInput.raise_()
        self.SubmitButton.raise_()
        self.MessageLabel.raise_()

        self.retranslateUi(WhatIsTheRiemannHypothesisScreen)
        QtCore.QMetaObject.connectSlotsByName(WhatIsTheRiemannHypothesisScreen)

    def retranslateUi(self, WhatIsTheRiemannHypothesisScreen):
        _translate = QtCore.QCoreApplication.translate
        WhatIsTheRiemannHypothesisScreen.setWindowTitle(_translate("WhatIsTheRiemannHypothesisScreen",
            "Visualising the Riemann Hypothesis - Introduction"))
        self.Title.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "Introduction"))
        self.IntroductionTab.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "Introduction"))
        self.HistoricalBackgroundLabel.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "<html><head/><body><p
            align=\"center\">Historical<br>Background</p></body></html>"))
        self.WhatIsTheRHLabel.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "<html><head/><body><p align=\"center\">What is
            the<br>Riemann Hypothesis</p></body></html>"))
        self.PracticalApplicationsLabel.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "<html><head/><body><p
            align=\"center\">Practical<br>Applications</p></body></html>"))
        self.PrevButton.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "Prev"))
        self.NextButton.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "Next"))
        self.SubTitleText.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">What is the Riemann
            Hypothesis</span></p></body></html>"))
        self.MainText.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "<html><head/><body><p>In his 1859 Paper \'On the Number of
            Primes Less Than a Given Magnitude\', Bernhard Riemann
            explored and researched the prime numbers. He did this using
            the riemann zeta function, which is the sum from 1 to
            infinity of n to the power -s, where s is the input to the
            function. </p><p>Riemann used a process called analytic
            continuation to allow this function to be true for not just
            all real numbers, but also imaginary and complex numbers.
            Imaginary numbers are denoted by the imaginary unit i, where
            i is the square root of -1, a number that is undefined using
            just the real numbers. A complex number is one that has a
            real part and an imaginary part. <br/>Riemann used complex
            numbers in the Riemann Zeta function and found something
            very interesting. First was that whenever the input to the
            function was a negative even integer, then the function
            always output 0. Second, was that the function will also
            output Zero when the imaginary part of the input is between
            zero and one. The first point was relatively easy to explain
            as to why it happened, leading to the negative even integers
            being known as the trivial zeroes for this function.
            However, the second point was a little bit harder to
            explain. Riemann managed to prove this point, but also
            noticed that these zeroes only occur when the imaginary part
            of the input is 1/2. This was not so easy to prove, and thus
            these zeroes were known as the nontrivial zeroes.</p><p>The
            Riemann Hypothesis states that \'the real part of every
            nontrivial zero of the Riemann Zeta Function is 1/2\'.
            </p></body></html>"))
        self.QuestionText.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "<html><head/><body><p><br/></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("WhatIsTheRiemannHypothesisScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "Submit"))
        self.MessageLabel.setText(_translate("WhatIsTheRiemannHypothesisScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/investigation_ui/__init__.py

```
"""
__init__.py
===========
Imports for the investigation_ui
"""


from .calculator import Ui_CalculatorScreen
from .single_calculator import Ui_SingleCalculatorScreen
from .table_calculator import Ui_TableCalculatorScreen
from .table_calculator_2 import Ui_TableCalculator2Screen
from .calculator_leaderboard import Ui_CalculatorLeaderboardScreen
from .graph_plots import Ui_GraphPlotsScreen
from .polar_graph import Ui_PolarGraphScreen
from .prime_counting_function import Ui_PrimeCountingFunctionScreen
from .prime_numbers import Ui_PrimeNumbersScreen
from .zeroes import Ui_ZeroesScreen
from .calculate_zeroes import Ui_CalculateZeroesScreen
from .calculate_zeroes_2 import Ui_CalculateZeroes2Screen
from .zeta_zeroes_plot import Ui_ZetaZeroesPlotScreen
from .zeta_approximation import Ui_ZetaApproximationScreen
```

program/user_interface/investigation_ui/calculate_zeroes.py

```
"""
calculate_zeroes.py
===================
A GUI for the calculate zeroes page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_CalculateZeroesScreen(object):

    def setupUi(self, CalculateZeroesScreen):
        CalculateZeroesScreen.setObjectName("CalculateZeroesScreen")
        CalculateZeroesScreen.resize(1340, 720)
        CalculateZeroesScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(CalculateZeroesScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(540, 20, 261, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.ZeroesTab = QtWidgets.QPushButton(self.TabBar)
        self.ZeroesTab.setGeometry(QtCore.QRect(10, 5, 220, 70))
        self.ZeroesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZeroesTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239,239);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
               "")
        self.ZeroesTab.setObjectName("ZeroesTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
               "border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.NextButton.setObjectName("NextButton")
        self.NoOfZeroesInput = QtWidgets.QLineEdit(self.MainWidget)
        self.NoOfZeroesInput.setGeometry(QtCore.QRect(405, 190, 531, 81))
        self.NoOfZeroesInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
               "color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\";\n"
               "border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);")
```

```
        self.NoOfZeroesInput.setText("")
        self.NoOfZeroesInput.setObjectName("NoOfZeroesInput")
        self.NoOfZeroesText = QtWidgets.QLabel(self.MainWidget)
        self.NoOfZeroesText.setGeometry(QtCore.QRect(120, 80, 1101, 61))
        self.NoOfZeroesText.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239); padding: 5px;\n"
               "text-decoration: underline;")
        self.NoOfZeroesText.setObjectName("NoOfZeroesText")
        self.CalculateButton = QtWidgets.QPushButton(self.MainWidget)
        self.CalculateButton.setGeometry(QtCore.QRect(590, 400, 141, 91))
        self.CalculateButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.CalculateButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.CalculateButton.setObjectName("CalculateButton")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(390, 300, 701, 61))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
               "font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")

        self.retranslateUi(CalculateZeroesScreen)
        QtCore.QMetaObject.connectSlotsByName(CalculateZeroesScreen)

    def retranslateUi(self, CalculateZeroesScreen):
        _translate = QtCore.QCoreApplication.translate
        CalculateZeroesScreen.setWindowTitle(_translate("CalculateZeroesScreen",
            "Visualising the Riemann Hypothesis - Zeta Zeroes"))
        self.Title.setText(_translate("CalculateZeroesScreen", "Zeta
            Zeroes"))
        self.ZeroesTab.setText(_translate("CalculateZeroesScreen",
            "Zeroes Calculator"))
        self.PrevButton.setText(_translate("CalculateZeroesScreen",
            "Prev"))
        self.NextButton.setText(_translate("CalculateZeroesScreen",
            "Next"))
        self.NoOfZeroesInput.setPlaceholderText(_translate("CalculateZeroesScreen",
            " Enter Number of Zeroes"))
        self.NoOfZeroesText.setText(_translate("CalculateZeroesScreen",
            "<html><head/><body><p align=\"right\"><span style=\"
            font-weight:600;\">How many zeroes would you like to
            calculate?</span></p></body></html>"))
        self.CalculateButton.setText(_translate("CalculateZeroesScreen",
            "Calculate\n"
"    "))
        self.ErrorLabel.setText(_translate("CalculateZeroesScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/investigation_ui/calculate_zeroes_2.py

```
"""
calculate_zeroes_2.py
====================
A GUI for the calculate zeroes 2 page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_CalculateZeroes2Screen(object):

    def setupUi(self, CalculateZeroes2Screen):
        CalculateZeroes2Screen.setObjectName("CalculateZeroes2Screen")
        CalculateZeroes2Screen.resize(1340, 735)
        CalculateZeroes2Screen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(CalculateZeroes2Screen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(540, 20, 261, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.ZeroesTab = QtWidgets.QPushButton(self.TabBar)
        self.ZeroesTab.setGeometry(QtCore.QRect(10, 5, 220, 70))
        self.ZeroesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZeroesTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239,239);\n"
               "font: 12pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
               "")
        self.ZeroesTab.setObjectName("ZeroesTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
```

```
       "border-radius: 20px;")
self.MainWidget.setObjectName("MainWidget")
self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.PrevButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 12pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.PrevButton.setObjectName("PrevButton")
self.NextButton = QtWidgets.QPushButton(self.MainWidget)
self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
self.NextButton.setStyleSheet("border: 2px solid;\n"
       "border-radius: 20px;\n"
       "border-color:rgb(69, 69, 69);\n"
       "background-color: rgb(69, 69, 69);\n"
       "font: 12pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
       "")
self.NextButton.setObjectName("NextButton")
self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 381, 41))
self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
    color:rgb(69, 69, 69);\n"
       "background-color: rgb(239, 239, 239); padding: 5px;")
self.SubTitleText.setObjectName("SubTitleText")
self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
self.ErrorLabel.setGeometry(QtCore.QRect(365, 390, 611, 61))
self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
       "font: 18pt \"Sans Serif\";")
self.ErrorLabel.setObjectName("ErrorLabel")
self.ZetaTable = QtWidgets.QTableWidget(self.MainWidget)
self.ZetaTable.setGeometry(QtCore.QRect(510, 30, 300, 351))
self.ZetaTable.setObjectName("ZetaTable")
self.ZetaTable.setColumnCount(2)
self.ZetaTable.setRowCount(0)
item = QtWidgets.QTableWidgetItem()
item.setTextAlignment(QtCore.Qt.AlignCenter)
font = QtGui.QFont()
font.setPointSize(16)
font.setBold(True)
font.setWeight(75)
item.setFont(font)
self.ZetaTable.setHorizontalHeaderItem(0, item)
item = QtWidgets.QTableWidgetItem()
item.setTextAlignment(QtCore.Qt.AlignCenter)
font = QtGui.QFont()
font.setPointSize(16)
```

```
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.ZetaTable.setHorizontalHeaderItem(1, item)
        self.DatabaseButton = QtWidgets.QPushButton(self.MainWidget)
        self.DatabaseButton.setGeometry(QtCore.QRect(440, 460, 200, 70))
        self.DatabaseButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.DatabaseButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 12pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.DatabaseButton.setObjectName("DatabaseButton")
        self.FileButton = QtWidgets.QPushButton(self.MainWidget)
        self.FileButton.setGeometry(QtCore.QRect(700, 460, 200, 70))
        self.FileButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.FileButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 12pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.FileButton.setObjectName("FileButton")
        self.retranslateUi(CalculateZeroes2Screen)
        QtCore.QMetaObject.connectSlotsByName(CalculateZeroes2Screen)

    def retranslateUi(self, CalculateZeroes2Screen):
        _translate = QtCore.QCoreApplication.translate
        CalculateZeroes2Screen.setWindowTitle(_translate("CalculateZeroes2Screen",
            "Visualising the Riemann Hypothesis - Zeta Zeroes"))
        self.Title.setText(_translate("CalculateZeroes2Screen", "Zeta
            Zeroes"))
        self.ZeroesTab.setText(_translate("CalculateZeroes2Screen",
            "Zeroes Calculator"))
        self.PrevButton.setText(_translate("CalculateZeroes2Screen",
            "Prev"))
        self.NextButton.setText(_translate("CalculateZeroes2Screen",
            "Next"))
        self.SubTitleText.setText(_translate("CalculateZeroes2Screen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Zeta Zeroes</span></p></body></html>"))
        self.ErrorLabel.setText(_translate("CalculateZeroes2Screen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        item = self.ZetaTable.horizontalHeaderItem(0)
        item.setText(_translate("CalculateZeroes2Screen", "Re(s)"))
        item = self.ZetaTable.horizontalHeaderItem(1)
        item.setText(_translate("CalculateZeroes2Screen", "Im(s)"))
        self.DatabaseButton.setText(_translate("CalculateZeroes2Screen",
            "Save to database"))
        self.FileButton.setText(_translate("CalculateZeroes2Screen",
            "Save to file"))
```

program/user_interface/investigation_ui/calculator.py

```
"""
calculator.py
=============
A GUI for the calculator page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_CalculatorScreen(object):

    def setupUi(self, CalculatorScreen):
        CalculatorScreen.setObjectName("CalculatorScreen")
        CalculatorScreen.resize(1340, 713)
        CalculatorScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(CalculatorScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(520, 20, 300, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setAlignment(QtCore.Qt.AlignCenter)
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.GraphsTab = QtWidgets.QPushButton(self.TabBar)
        self.GraphsTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.GraphsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphsTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69,69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.GraphsTab.setObjectName("GraphsTab")
        self.PrimesTab = QtWidgets.QPushButton(self.TabBar)
        self.PrimesTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.PrimesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrimesTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.PrimesTab.setObjectName("PrimesTab")
        self.CalculatorTab = QtWidgets.QPushButton(self.TabBar)
        self.CalculatorTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.CalculatorTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.CalculatorTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
               "")
        self.CalculatorTab.setObjectName("CalculatorTab")
        self.ZeroesTab = QtWidgets.QPushButton(self.TabBar)
        self.ZeroesTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ZeroesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZeroesTab.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.ZeroesTab.setObjectName("ZeroesTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
               "border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 681, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 80, 1251, 121))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.ZetaCalculatorButton =
            QtWidgets.QPushButton(self.MainWidget)
        self.ZetaCalculatorButton.setGeometry(QtCore.QRect(440, 460,
            200, 70))
        self.ZetaCalculatorButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaCalculatorButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.ZetaCalculatorButton.setObjectName("ZetaCalculatorButton")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(700, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.NotesButton.setObjectName("NotesButton")
        self.QuestionText = QtWidgets.QLabel(self.MainWidget)
        self.QuestionText.setGeometry(QtCore.QRect(420, 210, 501, 101))
        self.QuestionText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
               "background-color: rgb(239, 239, 239); padding: 5px;")
        self.QuestionText.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.QuestionText.setWordWrap(True)
        self.QuestionText.setObjectName("QuestionText")
        self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
        self.QuestionInput.setGeometry(QtCore.QRect(410, 330, 230, 60))
        self.QuestionInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
               "color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\";\n"
               "border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(700, 330, 121, 61))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
               "border-radius: 20px;\n"
               "border-color:rgb(69, 69, 69);\n"
               "background-color: rgb(69, 69, 69);\n"
               "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
               "")
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(410, 400, 530, 41))
        self.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
               "font: 18pt \"Sans Serif\";")
        self.MessageLabel.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.MessageLabel.setObjectName("MessageLabel")

        self.retranslateUi(CalculatorScreen)
        QtCore.QMetaObject.connectSlotsByName(CalculatorScreen)

    def retranslateUi(self, CalculatorScreen):
        _translate = QtCore.QCoreApplication.translate
        CalculatorScreen.setWindowTitle(_translate("CalculatorScreen",
            "Visualising the Riemann Hypothesis - Investigation"))
        self.Title.setText(_translate("CalculatorScreen",
            "Investigation"))
        self.GraphsTab.setText(_translate("CalculatorScreen", "Graphs"))
        self.PrimesTab.setText(_translate("CalculatorScreen", "Primes"))
        self.CalculatorTab.setText(_translate("CalculatorScreen",
            "Calculator"))
        self.ZeroesTab.setText(_translate("CalculatorScreen", "Zeroes"))
        self.PrevButton.setText(_translate("CalculatorScreen", "Prev"))
        self.NextButton.setText(_translate("CalculatorScreen", "Next"))
        self.SubTitleText.setText(_translate("CalculatorScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Calculating the Riemann Zeta
            Function</span></p></body></html>"))
        self.MainText.setText(_translate("CalculatorScreen",
            "<html><head/><body><p>By hand, working out values of the
            Riemann Zeta Function, is almost impossible and would take a
```

```
            lot of effort. However, using a computer program to do this
            instead is a much better idea.</p><p>Press the Zeta
            Calculator button below to calculate various values of the
            zeta function, you could even see if you manage to find a
            zeta zero.</p><p>Be sure to also answer this
            question!</p></body></html>"))
        self.ZetaCalculatorButton.setText(_translate("CalculatorScreen",
            "Zeta Calculator"))
        self.NotesButton.setText(_translate("CalculatorScreen", "Notes"))
        self.QuestionText.setText(_translate("CalculatorScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:16pt;
            font-weight:600;\">Question</span></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("CalculatorScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("CalculatorScreen",
            "Submit"))
        self.MessageLabel.setText(_translate("CalculatorScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/investigation_ui/calculator_leaderboard.py

```
"""
calculator_leaderboard.py
=========================
A GUI for the calculator leaderboard page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_CalculatorLeaderboardScreen(object):
    def setupUi(self, CalculatorLeaderboardScreen):
        CalculatorLeaderboardScreen.setObjectName("CalculatorLeaderboardScreen")
        CalculatorLeaderboardScreen.resize(1340, 720)
        CalculatorLeaderboardScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(CalculatorLeaderboardScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(560, 20, 221, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.SingleTab = QtWidgets.QPushButton(self.TabBar)
        self.SingleTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.SingleTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SingleTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69,69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SingleTab.setObjectName("SingleTab")
        self.TableTab = QtWidgets.QPushButton(self.TabBar)
        self.TableTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.TableTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TableTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.TableTab.setObjectName("TableTab")
        self.LeaderboardTab = QtWidgets.QPushButton(self.TabBar)
        self.LeaderboardTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.LeaderboardTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LeaderboardTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.LeaderboardTab.setObjectName("LeaderboardTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 381, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(400, 400, 541, 61))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")
        self.ZetaTable = QtWidgets.QTableWidget(self.MainWidget)
        self.ZetaTable.setGeometry(QtCore.QRect(370, 30, 600, 351))
        self.ZetaTable.setObjectName("ZetaTable")
        self.ZetaTable.setColumnCount(2)
        self.ZetaTable.setRowCount(0)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.ZetaTable.setHorizontalHeaderItem(0, item)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.ZetaTable.setHorizontalHeaderItem(1, item)

        self.retranslateUi(CalculatorLeaderboardScreen)
        QtCore.QMetaObject.connectSlotsByName(CalculatorLeaderboardScreen)

    def retranslateUi(self, CalculatorLeaderboardScreen):
        _translate = QtCore.QCoreApplication.translate
        CalculatorLeaderboardScreen.setWindowTitle(_translate("CalculatorLeaderboardScreen",
            "Visualising the Riemann Hypothesis - Calculator"))
        self.Title.setText(_translate("CalculatorLeaderboardScreen",
            "Calculator"))
        self.SingleTab.setText(_translate("CalculatorLeaderboardScreen",
            "Single"))
        self.TableTab.setText(_translate("CalculatorLeaderboardScreen",
            "Table"))
        self.LeaderboardTab.setText(_translate("CalculatorLeaderboardScreen",
            "Leaderboard"))
        self.PrevButton.setText(_translate("CalculatorLeaderboardScreen",
            "Prev"))
        self.NextButton.setText(_translate("CalculatorLeaderboardScreen",
            "Next"))
        self.SubTitleText.setText(_translate("CalculatorLeaderboardScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Zeta
            Leaderboard</span></p></body></html>"))
        self.ErrorLabel.setText(_translate("CalculatorLeaderboardScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        item = self.ZetaTable.horizontalHeaderItem(0)
        item.setText(_translate("CalculatorLeaderboardScreen",
            "Username"))
        item = self.ZetaTable.horizontalHeaderItem(1)
        item.setText(_translate("CalculatorLeaderboardScreen", "Number
            of Values Computed"))
```

program/user_interface/investigation_ui/graph_plots.py

```
"""
graph_plots.py
==============
A GUI for the graph plots page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_GraphPlotsScreen(object):

    def setupUi(self, GraphPlotsScreen):
        GraphPlotsScreen.setObjectName("GraphPlotsScreen")
        GraphPlotsScreen.resize(1340, 720)
        GraphPlotsScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(GraphPlotsScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(540, 20, 271, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.GraphsTab = QtWidgets.QPushButton(self.TabBar)
        self.GraphsTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.GraphsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphsTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.GraphsTab.setObjectName("GraphsTab")
        self.PrimesTab = QtWidgets.QPushButton(self.TabBar)
        self.PrimesTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.PrimesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrimesTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrimesTab.setObjectName("PrimesTab")
        self.CalculatorTab = QtWidgets.QPushButton(self.TabBar)
        self.CalculatorTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.CalculatorTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.CalculatorTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.CalculatorTab.setObjectName("CalculatorTab")
        self.ZeroesTab = QtWidgets.QPushButton(self.TabBar)
        self.ZeroesTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ZeroesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZeroesTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZeroesTab.setObjectName("ZeroesTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.GraphPlotsButton = QtWidgets.QPushButton(self.MainWidget)
        self.GraphPlotsButton.setGeometry(QtCore.QRect(440, 460, 200,
            70))
        self.GraphPlotsButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphPlotsButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.GraphPlotsButton.setObjectName("GraphPlotsButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 681, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 80, 1251, 341))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(450, 410, 461, 41))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 12pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(700, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")

        self.retranslateUi(GraphPlotsScreen)
        QtCore.QMetaObject.connectSlotsByName(GraphPlotsScreen)

    def retranslateUi(self, GraphPlotsScreen):
        _translate = QtCore.QCoreApplication.translate
        GraphPlotsScreen.setWindowTitle(_translate("GraphPlotsScreen",
            "Visualising the Riemann Hypothesis - Investigation"))
        self.Title.setText(_translate("GraphPlotsScreen",
            "Investigation"))
        self.GraphsTab.setText(_translate("GraphPlotsScreen", "Graphs"))
        self.PrimesTab.setText(_translate("GraphPlotsScreen", "Primes"))
        self.CalculatorTab.setText(_translate("GraphPlotsScreen",
            "Calculator"))
        self.ZeroesTab.setText(_translate("GraphPlotsScreen", "Zeroes"))
        self.PrevButton.setText(_translate("GraphPlotsScreen", "Prev"))
        self.NextButton.setText(_translate("GraphPlotsScreen", "Next"))
        self.GraphPlotsButton.setText(_translate("GraphPlotsScreen",
            "Graphs Plots"))
        self.SubTitleText.setText(_translate("GraphPlotsScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Visualising the Riemann
            Hypothesis</span></p></body></html>"))
        self.MainText.setText(_translate("GraphPlotsScreen",
            "<html><head/><body><p>As mentioned in the Investigation
            Section, there are numerous ways that the Riemann
            Hypothesiscan be visualised.</p><p>The main way of
            visualsing the Riemann Hypothesis is through the use of
            graphs, allowing for a visual representation of many
            different mathematical functions.</p><p>One of the most
            famous graphs of the Riemann Hypothesis is the polar graph
            of the Riemann Zeta Function, along the line Re(s) = 0.5.
            Polar Graphs allow for complex (2 dimensional) numbers to be
            represented visually. Unlike usual graphs where the x axis
            is the input to the function, and the y-axis is the output,
            the polar graph is only capable of displaying the output of
            the function, however, if the input domain is already
            defined and known, then this is not an
            issue<br/></p><p>Another graph used is the graph of the zeta
            zeroes. This graph plots complex numbers on an argand
            diagram, where these complex numbers are the roots, or
            zeroes of the Riemann Zeta Function. If that input is passed
            into the Riemann Zeta Function, and produces a result of 0,
            then that point is plotted onto the
            Graph.</p><p><br/></p><p>The last graph I am using to
            visualise the Riemann Hypothesis is the graph of the Prime
            Counting Function, and other functions that are used to
            approximate this. This allows you to visualsie Carl Gauss\'
            Prime Number Theorem, describing the distribution of prime
            numbers - A theorem that was proved using the Riemann
            Hypthesis</p><p><br/></p></body></html>"))
        self.ErrorLabel.setText(_translate("GraphPlotsScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        self.NotesButton.setText(_translate("GraphPlotsScreen", "Notes"))
```

program/user_interface/investigation_ui/polar_graph.py

```
"""
polar_graph.py
==============
A GUI for the polar graph page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_PolarGraphScreen(object):

    def setupUi(self, PolarGraphScreen):
        PolarGraphScreen.setObjectName("PolarGraphScreen")
        PolarGraphScreen.resize(1340, 720)
        PolarGraphScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(PolarGraphScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(550, 20, 281, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.PolarTab = QtWidgets.QPushButton(self.TabBar)
        self.PolarTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.PolarTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PolarTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.PolarTab.setObjectName("PolarTab")
        self.ZetaZeroesPlotTab = QtWidgets.QPushButton(self.TabBar)
        self.ZetaZeroesPlotTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.ZetaZeroesPlotTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaZeroesPlotTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZetaZeroesPlotTab.setObjectName("ZetaZeroesPlotTab")
        self.PrimeTab = QtWidgets.QPushButton(self.TabBar)
        self.PrimeTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.PrimeTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrimeTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrimeTab.setObjectName("PrimeTab")
        self.ZetaApproximationLabel = QtWidgets.QLabel(self.TabBar)
        self.ZetaApproximationLabel.setGeometry(QtCore.QRect(640, 5,
            200, 70))
        self.ZetaApproximationLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZetaApproximationLabel.setObjectName("ZetaApproximationLabel")
        self.ZetaApproximationTab = QtWidgets.QPushButton(self.TabBar)
        self.ZetaApproximationTab.setGeometry(QtCore.QRect(640, 5, 200,
            70))
        self.ZetaApproximationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaApproximationTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
```

```
        self.ZetaApproximationTab.setText("")
        self.ZetaApproximationTab.setObjectName("ZetaApproximationTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.GraphButton = QtWidgets.QPushButton(self.MainWidget)
        self.GraphButton.setGeometry(QtCore.QRect(680, 460, 200, 70))
        self.GraphButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.GraphButton.setObjectName("GraphButton")
        self.GraphInput = QtWidgets.QLineEdit(self.MainWidget)
        self.GraphInput.setGeometry(QtCore.QRect(460, 460, 200, 70))
        self.GraphInput.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"text-align: center;")
        self.GraphInput.setText("")
        self.GraphInput.setCursorPosition(0)
        self.GraphInput.setObjectName("GraphInput")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 821, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(50, 80, 1251, 341))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(450, 410, 461, 41))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 12pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")

        self.retranslateUi(PolarGraphScreen)
        QtCore.QMetaObject.connectSlotsByName(PolarGraphScreen)

    def retranslateUi(self, PolarGraphScreen):
        _translate = QtCore.QCoreApplication.translate
        PolarGraphScreen.setWindowTitle(_translate("PolarGraphScreen",
            "Visualising the Riemann Hypothesis - Investigation"))
        self.Title.setText(_translate("PolarGraphScreen", "Graph Plots"))
        self.PolarTab.setText(_translate("PolarGraphScreen", "Polar"))
        self.ZetaZeroesPlotTab.setText(_translate("PolarGraphScreen",
            "Zeroes"))
        self.PrimeTab.setText(_translate("PolarGraphScreen", "Prime"))
        self.ZetaApproximationLabel.setText(_translate("PolarGraphScreen",
            "<html><head/><body><p
            align=\"center\">Zeta<br/>Approximation</p></body></html>"))
        self.PrevButton.setText(_translate("PolarGraphScreen", "Prev"))
        self.NextButton.setText(_translate("PolarGraphScreen", "Next"))
        self.GraphButton.setText(_translate("PolarGraphScreen", "Graph"))
        self.GraphInput.setToolTip(_translate("PolarGraphScreen",
            "<html><head/><body><p align=\"center\">
            Re(s)</body></html>"))
        self.GraphInput.setPlaceholderText(_translate("PolarGraphScreen",
            "   Re(s)"))
        self.SubTitleText.setText(_translate("PolarGraphScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Polar Graph of the Riemann Zeta
            Function</span></p></body></html>"))
        self.MainText.setText(_translate("PolarGraphScreen",
            "<html><head/><body><p>The polar graph of the Riemann Zeta
            Function, is very well known due to it\'s mesmerising shape,
            and mathematical beauty.</p><p>The zeta function has two
            inputs and two outputs, that is a real and imaginary input,
            and a real and imaginary output.</p><p>The graph takes 1
            input, this corresponds to the real input to the zeta
            function. The imaginary input is determined by time. Time
            and the imaginary input are directly proportional, such that
            as time increases, the imaginary input increases by a linear
            amount.</p><p>Then at any given time, the graph will plot
            the point that is the output of the zeta function at that
            moment, and all of the previous outputs. Because the graph
            depends on time, it will be constantly changing, and
            plotting new points.</p><p><br/></p><p>Try it out below!
            Enter 0.5 for the best results.</p></body></html>"))
        self.ErrorLabel.setText(_translate("PolarGraphScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/investigation_ui/prime_counting_function.py

```
"""
prime_coutning_function.py
==========================
A GUI for the prime counting function page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_PrimeCountingFunctionScreen(object):
    def setupUi(self, PrimeCountingFunctionScreen):
        PrimeCountingFunctionScreen.setObjectName("PrimeCountingFunctionScreen")
        PrimeCountingFunctionScreen.resize(1340, 720)
        PrimeCountingFunctionScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(PrimeCountingFunctionScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(550, 20, 251, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.ZetaZeroesPlotTab = QtWidgets.QPushButton(self.TabBar)
        self.ZetaZeroesPlotTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.ZetaZeroesPlotTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaZeroesPlotTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZetaZeroesPlotTab.setObjectName("ZetaZeroesPlotTab")
        self.PolarTab = QtWidgets.QPushButton(self.TabBar)
        self.PolarTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.PolarTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PolarTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PolarTab.setObjectName("PolarTab")
        self.PrimeTab = QtWidgets.QPushButton(self.TabBar)
        self.PrimeTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.PrimeTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrimeTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.PrimeTab.setObjectName("PrimeTab")
        self.ZetaApproximationTab = QtWidgets.QPushButton(self.TabBar)
        self.ZetaApproximationTab.setGeometry(QtCore.QRect(640, 5, 200,
            70))
        self.ZetaApproximationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaApproximationTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ZetaApproximationTab.setText("")
        self.ZetaApproximationTab.setObjectName("ZetaApproximationTab")
        self.ZetaApproximationLabel = QtWidgets.QLabel(self.TabBar)
        self.ZetaApproximationLabel.setGeometry(QtCore.QRect(640, 5,
            200, 70))
        self.ZetaApproximationLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZetaApproximationLabel.setObjectName("ZetaApproximationLabel")
        self.ZetaZeroesPlotTab.raise_()
        self.PolarTab.raise_()
        self.PrimeTab.raise_()
        self.ZetaApproximationLabel.raise_()
        self.ZetaApproximationTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.GraphButton = QtWidgets.QPushButton(self.MainWidget)
        self.GraphButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.GraphButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.GraphButton.setObjectName("GraphButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 681, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 80, 1251, 361))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")

        self.retranslateUi(PrimeCountingFunctionScreen)
        QtCore.QMetaObject.connectSlotsByName(PrimeCountingFunctionScreen)

    def retranslateUi(self, PrimeCountingFunctionScreen):
        _translate = QtCore.QCoreApplication.translate
        PrimeCountingFunctionScreen.setWindowTitle(_translate("PrimeCountingFunctionScreen",
            "Visualising the Riemann Hypothesis - Investigation"))
        self.Title.setText(_translate("PrimeCountingFunctionScreen",
            "Graph Plots"))
        self.ZetaZeroesPlotTab.setText(_translate("PrimeCountingFunctionScreen",
            "Zeroes"))
        self.PolarTab.setText(_translate("PrimeCountingFunctionScreen",
            "Polar"))
        self.PrimeTab.setText(_translate("PrimeCountingFunctionScreen",
            "Prime"))
        self.ZetaApproximationLabel.setText(_translate("PrimeCountingFunctionScreen",
            "<html><head/><body><p
            align=\"center\">Zeta<br/>Approximation</p></body></html>"))
        self.PrevButton.setText(_translate("PrimeCountingFunctionScreen",
            "Prev"))
        self.NextButton.setText(_translate("PrimeCountingFunctionScreen",
            "Next"))
        self.GraphButton.setText(_translate("PrimeCountingFunctionScreen",
            "Graph"))
        self.SubTitleText.setText(_translate("PrimeCountingFunctionScreen",
            "<html><head/><body><p><span style=\" font-weight:600;\">The
            Prime Counting Function</span></p></body></html>"))
        self.MainText.setText(_translate("PrimeCountingFunctionScreen",
            "<html><head/><body><p>The prime number theorem was a
            theorem thought of by Carl Friedrich Gauss near the end of
            the 18th century. This theorem describes the distribution of
            the prime numbers. It formalises the intuitive idea that as
            numbers get larger, the prime numbers are less common, by
            precisely quantifying the rate at which this occurs. One way
            this theorem was modelled was through the prime counting
            function (denoted (N)). Where (N) gives the number of primes
            that are less than or equal to N . Given this we can say
            that as N   then (N)/log(N)  1, where log(N) is the natural
            logarithm of N This, therefore, means that: (N)
            N/log(N).</p><p>This means we can approximate the numbers of
            primes less than or equal to N, by calculating N/log(N).
            However, Peter Dirichlet and Carl Friedrich Gauss came up
            this a much better approximation for (N). They said that:
            (N)  Li(N), where Li(N) is the logarithmic integral of N.
            Using Li(N) is much more accurate than using N/log(N). As
            well as the prime counting function, we also have a similar
            function called the prime power function (denoted by (N)).
            In the prime counting function, you would get 1 point per
            prime number (less than or equal to N ). But in the prime
            power function, you get 1 point per prime + 1/2 point per
            prime squared +1/3 point per prime cubed and so
            on.</p><p>But how is this related to the Riemann Hypothesis?
            It turns out that the prime number theorem was proved by
            using the Riemann Zeta Function. If the Riemann Hypothesis
            were true, then the difference between the prime counting
            function, and prime power function was be as small as
            possible. This is significant as it would lead to many other
            conjectures involving prime numbers to also be proven
            true.</p><p>Press the graph button to see the distribution
            of prime numbers, and how this is approximated using other
            functions.</p></body></html>"))
```

program/user_interface/investigation_ui/prime_numbers.py

```
"""
prime_numbers.py
================
A GUI for the prime numbers page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_PrimeNumbersScreen(object):

    def setupUi(self, PrimeNumbersScreen):
        PrimeNumbersScreen.setObjectName("PrimeNumbersScreen")
        PrimeNumbersScreen.resize(1340, 720)
        PrimeNumbersScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(PrimeNumbersScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(540, 20, 291, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.GraphsTab = QtWidgets.QPushButton(self.TabBar)
        self.GraphsTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.GraphsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphsTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.GraphsTab.setObjectName("GraphsTab")
        self.PrimesTab = QtWidgets.QPushButton(self.TabBar)
        self.PrimesTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.PrimesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrimesTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.PrimesTab.setObjectName("PrimesTab")
        self.CalculatorTab = QtWidgets.QPushButton(self.TabBar)
        self.CalculatorTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.CalculatorTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.CalculatorTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.CalculatorTab.setObjectName("CalculatorTab")
        self.ZeroesTab = QtWidgets.QPushButton(self.TabBar)
        self.ZeroesTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ZeroesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZeroesTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZeroesTab.setObjectName("ZeroesTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 681, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 80, 1251, 341))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")

        self.retranslateUi(PrimeNumbersScreen)
        QtCore.QMetaObject.connectSlotsByName(PrimeNumbersScreen)

    def retranslateUi(self, PrimeNumbersScreen):
        _translate = QtCore.QCoreApplication.translate
        PrimeNumbersScreen.setWindowTitle(_translate("PrimeNumbersScreen",
            "Visualising the Riemann Hypothesis - Investigation"))
        self.Title.setText(_translate("PrimeNumbersScreen",
            "Investigation"))
        self.GraphsTab.setText(_translate("PrimeNumbersScreen",
            "Graphs"))
        self.PrimesTab.setText(_translate("PrimeNumbersScreen",
            "Primes"))
        self.CalculatorTab.setText(_translate("PrimeNumbersScreen",
            "Calculator"))
        self.ZeroesTab.setText(_translate("PrimeNumbersScreen",
            "Zeroes"))
        self.PrevButton.setText(_translate("PrimeNumbersScreen", "Prev"))
        self.NextButton.setText(_translate("PrimeNumbersScreen", "Next"))
        self.SubTitleText.setText(_translate("PrimeNumbersScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Prime Numbers</span></p></body></html>"))
        self.MainText.setText(_translate("PrimeNumbersScreen",
            "<html><head/><body><p>The Riemann Hypothesis and Prime
            Numbers go hand in hand. A prime number is a number larger
            than 1, such that it is only divisible by 1 and the number
            itself. </p><p>Just like with the non-trivial zeroes, there
            is not real way to calculate prime numbers apart from trial
            and error. However, we are able to use the Riemann
            Hypothesis in order to understand the distribution of prime
            numbers, and if the Riemann Hypothesis was proven to be
            true, we would be able to calculate certain values of prime
            numbers using a formula. </p><p>Another connection between
            the Riemann Hypothesis and the prime numbers, is that
            Leonhard Euler proved that the zeta function could be
            written as an infite product over all of the primes such
            that (s) = (1/(1-p^-s)), where p is a prime
            number.</p><p>Furthermore if the Riemann Hypothesis was
            proven to be true, it would mean that the Weak Goldbach
            Conjecture is also true, which states that \'All odd
            integers greater than 5 are the sum of three primes\'.
            Furthermore, it would also mean that between consecutive
            cube numbers, there would always be at least 1 prime
            number.</p><p>All of these theorems and conjectures
            highlight the importance of the Riemann hypothesis, and that
            if it was to be proven, it would lead to several major
            break- throughs, not only in mathematics but quantum physics
            and computer science - fields that heavily use the prime
            numbers.</p></body></html>"))
        self.NotesButton.setText(_translate("PrimeNumbersScreen",
            "Notes"))
```

program/user_interface/investigation_ui/single_calculator.py

```
"""
single_calculator.py
====================
A GUI for the single calculator page of the investigation section
"""
```

```
from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_SingleCalculatorScreen(object):

    def setupUi(self, SingleCalculatorScreen):
        SingleCalculatorScreen.setObjectName("SingleCalculatorScreen")
        SingleCalculatorScreen.resize(1340, 720)
        SingleCalculatorScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(SingleCalculatorScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(560, 20, 221, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.SingleTab = QtWidgets.QPushButton(self.TabBar)
        self.SingleTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.SingleTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SingleTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239,239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.SingleTab.setObjectName("SingleTab")
        self.TableTab = QtWidgets.QPushButton(self.TabBar)
        self.TableTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.TableTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TableTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.TableTab.setObjectName("TableTab")
        self.LeaderboardTab = QtWidgets.QPushButton(self.TabBar)
        self.LeaderboardTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.LeaderboardTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LeaderboardTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.LeaderboardTab.setObjectName("LeaderboardTab")
       self.MainWidget = QtWidgets.QWidget(self.widget)
       self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
       self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
"border-radius: 20px;")
       self.MainWidget.setObjectName("MainWidget")
       self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
       self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
       self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.PrevButton.setObjectName("PrevButton")
       self.NextButton = QtWidgets.QPushButton(self.MainWidget)
       self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
       self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.NextButton.setObjectName("NextButton")
       self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
       self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 421, 51))
       self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
           color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
       self.SubTitleText.setObjectName("SubTitleText")
       self.EquationImage = QtWidgets.QLabel(self.MainWidget)
       self.EquationImage.setGeometry(QtCore.QRect(155, 70, 1031, 131))
       self.EquationImage.setText("")
       self.EquationImage.setPixmap(QtGui.QPixmap("ui/investigation_screens/../../media/riemanns_funct
       self.EquationImage.setObjectName("EquationImage")
       self.ZetaInput = QtWidgets.QLineEdit(self.MainWidget)
       self.ZetaInput.setGeometry(QtCore.QRect(170, 340, 371, 81))
       self.ZetaInput.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
"color: rgb(69, 69, 69);\n"
"font: 36pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
```

```
        self.ZetaInput.setText("")
        self.ZetaInput.setObjectName("ZetaInput")
        self.ZetaOutput = QtWidgets.QLabel(self.MainWidget)
        self.ZetaOutput.setGeometry(QtCore.QRect(800, 340, 431, 81))
        self.ZetaOutput.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 36pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.ZetaOutput.setText("")
        self.ZetaOutput.setTextInteractionFlags(QtCore.Qt.LinksAccessibleByMouse|QtCore.Qt.TextSelectab
        self.ZetaOutput.setObjectName("ZetaOutput")
        self.InputText = QtWidgets.QLabel(self.MainWidget)
        self.InputText.setGeometry(QtCore.QRect(230, 270, 281, 51))
        self.InputText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"text-decoration: underline;")
        self.InputText.setObjectName("InputText")
        self.OutputText = QtWidgets.QLabel(self.MainWidget)
        self.OutputText.setGeometry(QtCore.QRect(860, 250, 321, 61))
        self.OutputText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"text-decoration: underline;")
        self.OutputText.setObjectName("OutputText")
        self.DatabaseButton = QtWidgets.QPushButton(self.MainWidget)
        self.DatabaseButton.setGeometry(QtCore.QRect(429, 460, 211, 70))
        self.DatabaseButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.DatabaseButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.DatabaseButton.setObjectName("DatabaseButton")
        self.FileButton = QtWidgets.QPushButton(self.MainWidget)
        self.FileButton.setGeometry(QtCore.QRect(700, 460, 200, 70))
        self.FileButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.FileButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.FileButton.setObjectName("FileButton")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(370, 210, 631, 41))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")
        self.CalculateButton = QtWidgets.QPushButton(self.MainWidget)
        self.CalculateButton.setGeometry(QtCore.QRect(570, 350, 200, 70))
        self.CalculateButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.CalculateButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.CalculateButton.setObjectName("CalculateButton")
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.EquationImage.raise_()
        self.ZetaInput.raise_()
        self.SubTitleText.raise_()
        self.ZetaOutput.raise_()
        self.InputText.raise_()
        self.OutputText.raise_()
        self.DatabaseButton.raise_()
        self.FileButton.raise_()
        self.CalculateButton.raise_()
        self.ErrorLabel.raise_()

        self.retranslateUi(SingleCalculatorScreen)
        QtCore.QMetaObject.connectSlotsByName(SingleCalculatorScreen)

    def retranslateUi(self, SingleCalculatorScreen):
        _translate = QtCore.QCoreApplication.translate
        SingleCalculatorScreen.setWindowTitle(_translate("SingleCalculatorScreen",
            "Visualising the Riemann Hypothesis - Calculator"))
        self.Title.setText(_translate("SingleCalculatorScreen",
            "Calculator"))
        self.SingleTab.setText(_translate("SingleCalculatorScreen",
            "Single"))
        self.TableTab.setText(_translate("SingleCalculatorScreen",
            "Table"))
        self.LeaderboardTab.setText(_translate("SingleCalculatorScreen",
            "Leaderboard"))
        self.PrevButton.setText(_translate("SingleCalculatorScreen",
            "Prev"))
        self.NextButton.setText(_translate("SingleCalculatorScreen",
            "Next"))
        self.SubTitleText.setText(_translate("SingleCalculatorScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Single Zeta
            Calculator</span></p></body></html>"))
        self.InputText.setText(_translate("SingleCalculatorScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Input Value
            (s):</span></p></body></html>"))
        self.OutputText.setText(_translate("SingleCalculatorScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Output Value
            (s):</span></p></body></html>"))
        self.DatabaseButton.setText(_translate("SingleCalculatorScreen",
            "Save to database"))
        self.FileButton.setText(_translate("SingleCalculatorScreen",
            "Save to file"))
        self.ErrorLabel.setText(_translate("SingleCalculatorScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        self.CalculateButton.setText(_translate("SingleCalculatorScreen",
            "Calculate "))
```

program/user_interface/investigation_ui/table_calculator.py

```
"""
table_calculator.py
===================
A GUI for the table calculator page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_TableCalculatorScreen(object):

    def setupUi(self, TableCalculatorScreen):
        TableCalculatorScreen.setObjectName("TableCalculatorScreen")
        TableCalculatorScreen.resize(1340, 720)
        TableCalculatorScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(TableCalculatorScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(560, 20, 221, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
```

```
       self.SingleTab = QtWidgets.QPushButton(self.TabBar)
       self.SingleTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
       self.SingleTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.SingleTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69,69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.SingleTab.setObjectName("SingleTab")
       self.TableTab = QtWidgets.QPushButton(self.TabBar)
       self.TableTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
       self.TableTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.TableTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
       self.TableTab.setObjectName("TableTab")
       self.LeaderboardTab = QtWidgets.QPushButton(self.TabBar)
       self.LeaderboardTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
       self.LeaderboardTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.LeaderboardTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69,69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.LeaderboardTab.setObjectName("LeaderboardTab")
       self.MainWidget = QtWidgets.QWidget(self.widget)
       self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
       self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
"border-radius: 20px;")
       self.MainWidget.setObjectName("MainWidget")
       self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
       self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
       self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.PrevButton.setObjectName("PrevButton")
       self.NextButton = QtWidgets.QPushButton(self.MainWidget)
       self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
       self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.NextButton.setObjectName("NextButton")
       self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
       self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 381, 41))
       self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
           color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
       self.SubTitleText.setObjectName("SubTitleText")
       self.StartInput = QtWidgets.QLineEdit(self.MainWidget)
       self.StartInput.setGeometry(QtCore.QRect(485, 110, 421, 81))
       self.StartInput.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
       self.StartInput.setText("")
       self.StartInput.setObjectName("StartInput")
       self.StartValueText = QtWidgets.QLabel(self.MainWidget)
       self.StartValueText.setGeometry(QtCore.QRect(100, 120, 291, 61))
       self.StartValueText.setStyleSheet("font: 36pt \"Sans Serif\";
           color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"text-decoration: underline;")
       self.StartValueText.setObjectName("StartValueText")
       self.CalculateButton = QtWidgets.QPushButton(self.MainWidget)
       self.CalculateButton.setGeometry(QtCore.QRect(980, 215, 200, 70))
       self.CalculateButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.CalculateButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.CalculateButton.setObjectName("CalculateButton")
       self.StepText = QtWidgets.QLabel(self.MainWidget)
       self.StepText.setGeometry(QtCore.QRect(100, 220, 291, 61))
       self.StepText.setStyleSheet("font: 36pt \"Sans Serif\";
           color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"text-decoration: underline;")
       self.StepText.setObjectName("StepText")
       self.StepInput = QtWidgets.QLineEdit(self.MainWidget)
       self.StepInput.setGeometry(QtCore.QRect(485, 210, 421, 81))
       self.StepInput.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.StepInput.setText("")
        self.StepInput.setObjectName("StepInput")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(400, 410, 701, 61))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")
        self.NoOfValuesText = QtWidgets.QLabel(self.MainWidget)
        self.NoOfValuesText.setGeometry(QtCore.QRect(10, 320, 381, 61))
        self.NoOfValuesText.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"text-decoration: underline;")
        self.NoOfValuesText.setObjectName("NoOfValuesText")
        self.NoOfValuesInput = QtWidgets.QLineEdit(self.MainWidget)
        self.NoOfValuesInput.setGeometry(QtCore.QRect(485, 310, 421, 81))
        self.NoOfValuesInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.NoOfValuesInput.setText("")
        self.NoOfValuesInput.setObjectName("NoOfValuesInput")
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.StartInput.raise_()
        self.SubTitleText.raise_()
        self.StartValueText.raise_()
        self.CalculateButton.raise_()
        self.StepText.raise_()
        self.StepInput.raise_()
        self.ErrorLabel.raise_()
        self.NoOfValuesText.raise_()
        self.NoOfValuesInput.raise_()

        self.retranslateUi(TableCalculatorScreen)
        QtCore.QMetaObject.connectSlotsByName(TableCalculatorScreen)

    def retranslateUi(self, TableCalculatorScreen):
        _translate = QtCore.QCoreApplication.translate
        TableCalculatorScreen.setWindowTitle(_translate("TableCalculatorScreen",
            "Visualising the Riemann Hypothesis - Calculator"))
        self.Title.setText(_translate("TableCalculatorScreen",
            "Calculator"))
        self.SingleTab.setText(_translate("TableCalculatorScreen",
            "Single"))
        self.TableTab.setText(_translate("TableCalculatorScreen",
            "Table"))
        self.LeaderboardTab.setText(_translate("TableCalculatorScreen",
            "Leaderboard"))
        self.PrevButton.setText(_translate("TableCalculatorScreen",
            "Prev"))
        self.NextButton.setText(_translate("TableCalculatorScreen",
            "Next"))
        self.SubTitleText.setText(_translate("TableCalculatorScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Table Zeta
            Calculation</span></p></body></html>"))
        self.StartInput.setPlaceholderText(_translate("TableCalculatorScreen",
            " Enter Start Value"))
        self.StartValueText.setText(_translate("TableCalculatorScreen",
            "<html><head/><body><p align=\"right\"><span style=\"
            font-weight:600;\">Start Value:</span></p></body></html>"))
        self.CalculateButton.setText(_translate("TableCalculatorScreen",
            "Calculate "))
        self.StepText.setText(_translate("TableCalculatorScreen",
            "<html><head/><body><p align=\"right\"><span style=\"
            font-weight:600;\">Step:</span></p></body></html>"))
        self.StepInput.setPlaceholderText(_translate("TableCalculatorScreen",
            " Enter Step Value"))
        self.ErrorLabel.setText(_translate("TableCalculatorScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        self.NoOfValuesText.setText(_translate("TableCalculatorScreen",
            "<html><head/><body><p align=\"right\"><span style=\"
            font-weight:600;\">No. of Values:</span></p></body></html>"))
        self.NoOfValuesInput.setPlaceholderText(_translate("TableCalculatorScreen",
            " Enter No. of Values"))
```

program/user_interface/investigation_ui/table_calculator_2.py

```
"""
table_calculator_2.py
=====================
A GUI for the table calculator 2 page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets
```

```
class Ui_TableCalculator2Screen(object):

    def setupUi(self, TableCalculator2Screen):
        TableCalculator2Screen.setObjectName("TableCalculator2Screen")
        TableCalculator2Screen.resize(1340, 720)
        TableCalculator2Screen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(TableCalculator2Screen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(560, 20, 221, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.SingleTab = QtWidgets.QPushButton(self.TabBar)
        self.SingleTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.SingleTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SingleTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69,69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SingleTab.setObjectName("SingleTab")
        self.TableTab = QtWidgets.QPushButton(self.TabBar)
        self.TableTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.TableTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TableTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.TableTab.setObjectName("TableTab")
        self.LeaderboardTab = QtWidgets.QPushButton(self.TabBar)
        self.LeaderboardTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.LeaderboardTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LeaderboardTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.LeaderboardTab.setObjectName("LeaderboardTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239, 239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 381, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(110, 395, 1120, 61))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")
        self.ZetaTable = QtWidgets.QTableWidget(self.MainWidget)
        self.ZetaTable.setGeometry(QtCore.QRect(510, 30, 300, 351))
        self.ZetaTable.setObjectName("ZetaTable")
        self.ZetaTable.setColumnCount(2)
        self.ZetaTable.setRowCount(0)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.ZetaTable.setHorizontalHeaderItem(0, item)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.ZetaTable.setHorizontalHeaderItem(1, item)
        self.DatabaseButton = QtWidgets.QPushButton(self.MainWidget)
        self.DatabaseButton.setGeometry(QtCore.QRect(420, 460, 200, 70))
        self.DatabaseButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.DatabaseButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.DatabaseButton.setObjectName("DatabaseButton")
        self.FileButton = QtWidgets.QPushButton(self.MainWidget)
        self.FileButton.setGeometry(QtCore.QRect(720, 460, 200, 70))
        self.FileButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.FileButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.FileButton.setObjectName("FileButton")

        self.retranslateUi(TableCalculator2Screen)
        QtCore.QMetaObject.connectSlotsByName(TableCalculator2Screen)

    def retranslateUi(self, TableCalculator2Screen):
        _translate = QtCore.QCoreApplication.translate
        TableCalculator2Screen.setWindowTitle(_translate("TableCalculator2Screen",
            "Visualising the Riemann Hypothesis - Calculator"))
        self.Title.setText(_translate("TableCalculator2Screen",
            "Calculator"))
        self.SingleTab.setText(_translate("TableCalculator2Screen",
            "Single"))
        self.TableTab.setText(_translate("TableCalculator2Screen",
            "Table"))
        self.LeaderboardTab.setText(_translate("TableCalculator2Screen",
            "Leaderboard"))
        self.PrevButton.setText(_translate("TableCalculator2Screen",
            "Prev"))
        self.NextButton.setText(_translate("TableCalculator2Screen",
            "Next"))
        self.SubTitleText.setText(_translate("TableCalculator2Screen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Table Zeta
            Calculation</span></p></body></html>"))
        self.ErrorLabel.setText(_translate("TableCalculator2Screen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        item = self.ZetaTable.horizontalHeaderItem(0)
        item.setText(_translate("TableCalculator2Screen", "Input (s)"))
        item = self.ZetaTable.horizontalHeaderItem(1)
        item.setText(_translate("TableCalculator2Screen", "Output (s)"))
        self.DatabaseButton.setText(_translate("TableCalculator2Screen",
            "Save to database"))
        self.FileButton.setText(_translate("TableCalculator2Screen",
            "Save to file"))
```

program/user_interface/investigation_ui/zeroes.py

```
"""
zeroes.py
=========
A GUI for the zeroes page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ZeroesScreen(object):

    def setupUi(self, ZeroesScreen):
        ZeroesScreen.setObjectName("ZeroesScreen")
        ZeroesScreen.resize(1340, 720)
        ZeroesScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ZeroesScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(530, 20, 291, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.GraphsTab = QtWidgets.QPushButton(self.TabBar)
        self.GraphsTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.GraphsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphsTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69,69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.GraphsTab.setObjectName("GraphsTab")
        self.PrimesTab = QtWidgets.QPushButton(self.TabBar)
        self.PrimesTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.PrimesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrimesTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrimesTab.setObjectName("PrimesTab")
        self.CalculatorTab = QtWidgets.QPushButton(self.TabBar)
        self.CalculatorTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.CalculatorTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.CalculatorTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.CalculatorTab.setObjectName("CalculatorTab")
        self.ZeroesTab = QtWidgets.QPushButton(self.TabBar)
        self.ZeroesTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ZeroesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZeroesTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.ZeroesTab.setObjectName("ZeroesTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
```

```
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 901, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(30, 70, 1251, 211))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.CalculateButton = QtWidgets.QPushButton(self.MainWidget)
        self.CalculateButton.setGeometry(QtCore.QRect(440, 460, 200, 70))
        self.CalculateButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.CalculateButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.CalculateButton.setObjectName("CalculateButton")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(700, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")
        self.QuestionText = QtWidgets.QLabel(self.MainWidget)
        self.QuestionText.setGeometry(QtCore.QRect(420, 270, 501, 71))
        self.QuestionText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.QuestionText.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.QuestionText.setWordWrap(True)
        self.QuestionText.setObjectName("QuestionText")
        self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
        self.QuestionInput.setGeometry(QtCore.QRect(390, 350, 230, 60))
        self.QuestionInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(720, 350, 121, 61))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(410, 410, 530, 41))
        self.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.MessageLabel.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.MessageLabel.setObjectName("MessageLabel")
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.MainText.raise_()
        self.CalculateButton.raise_()
        self.NotesButton.raise_()
        self.QuestionText.raise_()
        self.QuestionInput.raise_()
        self.MessageLabel.raise_()
        self.SubTitleText.raise_()
        self.SubmitButton.raise_()

        self.retranslateUi(ZeroesScreen)
        QtCore.QMetaObject.connectSlotsByName(ZeroesScreen)

    def retranslateUi(self, ZeroesScreen):
        _translate = QtCore.QCoreApplication.translate
        ZeroesScreen.setWindowTitle(_translate("ZeroesScreen",
            "Visualising the Riemann Hypothesis - Investigation"))
        self.Title.setText(_translate("ZeroesScreen", "Investigation"))
        self.GraphsTab.setText(_translate("ZeroesScreen", "Graphs"))
        self.PrimesTab.setText(_translate("ZeroesScreen", "Primes"))
        self.CalculatorTab.setText(_translate("ZeroesScreen",
            "Calculator"))
        self.ZeroesTab.setText(_translate("ZeroesScreen", "Zeroes"))
        self.PrevButton.setText(_translate("ZeroesScreen", "Prev"))
        self.NextButton.setText(_translate("ZeroesScreen", "Next"))
        self.SubTitleText.setText(_translate("ZeroesScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Calculating the Zeroes of the Riemann
            Zeta Function</span></p></body></html>"))
        self.MainText.setText(_translate("ZeroesScreen",
            "<html><head/><body><p>The zeroes of the Riemann Zeta
            Function is where the mystery of the Riemann Hypothesis
            lies.</p><p>A zero, also called a root, of the function f(x)
            is the x value such that f(x) = 0. For the zeta function,
            these roots are in two categories, trival, and non-trivial
            zeroes. The trivial zeroes, are much simply to understand.
            These occur when the input to the zeta function, is a
            negative even integer. There is solid proof for this.
            However, the non-trivial zeroes are mcuh more complex. The
            Riemann Hypothesis states that these non-trivial zeroes
            occur when the real part of the input to the zeta function
            is equal to 1/2. Although it has been proven that the zeroes
            must occur when the real part of the input is between 0 and
            1, it is no more specific than that. Although all
            non-trivial zeroes every calculated have had real part
            1/2.</p><p>There is no proof for why the non-trivial zeroes
            occur at 1/2, but we can try to calculate the non-trivial
            zeroes by setting the real part of our input to the zeta
            function to be 1/2, and varying the imaginary part. Click
            Calculate Zeroes to try and find the values of some of the
            non-trivial zeroes of the zeta function.</p></body></html>"))
        self.CalculateButton.setText(_translate("ZeroesScreen",
            "Calculate Zeroes"))
        self.NotesButton.setText(_translate("ZeroesScreen", "Notes"))
        self.QuestionText.setText(_translate("ZeroesScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:16pt;
            font-weight:600;\">Question</span></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("ZeroesScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("ZeroesScreen", "Submit"))
        self.MessageLabel.setText(_translate("ZeroesScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/investigation_ui/zeta_approximation.py

```
"""
zeta_approximation.py
=====================
A GUI for the zeta approximation page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ZetaApproximationScreen(object):

    def setupUi(self, ZetaApproximationScreen):
        ZetaApproximationScreen.setObjectName("ZetaApproximationScreen")
        ZetaApproximationScreen.resize(1340, 720)
        ZetaApproximationScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ZetaApproximationScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(550, 20, 281, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.PolarTab = QtWidgets.QPushButton(self.TabBar)
        self.PolarTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.PolarTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PolarTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PolarTab.setObjectName("PolarTab")
        self.ZetaZeroesPlotTab = QtWidgets.QPushButton(self.TabBar)
        self.ZetaZeroesPlotTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.ZetaZeroesPlotTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaZeroesPlotTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZetaZeroesPlotTab.setObjectName("ZetaZeroesPlotTab")
        self.PrimeTab = QtWidgets.QPushButton(self.TabBar)
        self.PrimeTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.PrimeTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrimeTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrimeTab.setObjectName("PrimeTab")
        self.ZetaApproximationLabel = QtWidgets.QLabel(self.TabBar)
        self.ZetaApproximationLabel.setGeometry(QtCore.QRect(640, 5,
            200, 70))
        self.ZetaApproximationLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.ZetaApproximationLabel.setObjectName("ZetaApproximationLabel")
        self.ZetaApproximationTab = QtWidgets.QPushButton(self.TabBar)
        self.ZetaApproximationTab.setGeometry(QtCore.QRect(640, 5, 200,
            70))
        self.ZetaApproximationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaApproximationTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ZetaApproximationTab.setText("")
        self.ZetaApproximationTab.setObjectName("ZetaApproximationTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.GraphButton = QtWidgets.QPushButton(self.MainWidget)
        self.GraphButton.setGeometry(QtCore.QRect(680, 460, 200, 70))
        self.GraphButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.GraphButton.setObjectName("GraphButton")
        self.GraphInput = QtWidgets.QLineEdit(self.MainWidget)
        self.GraphInput.setGeometry(QtCore.QRect(460, 460, 200, 70))
        self.GraphInput.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"text-align: center;")
        self.GraphInput.setText("")
        self.GraphInput.setCursorPosition(0)
        self.GraphInput.setAlignment(QtCore.Qt.AlignCenter)
        self.GraphInput.setObjectName("GraphInput")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 821, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(50, 80, 1251, 201))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(310, 400, 721, 41))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 12pt \"Sans Serif\";")
```

```
        self.ErrorLabel.setObjectName("ErrorLabel")

        self.retranslateUi(ZetaApproximationScreen)
        QtCore.QMetaObject.connectSlotsByName(ZetaApproximationScreen)

    def retranslateUi(self, ZetaApproximationScreen):
        _translate = QtCore.QCoreApplication.translate
        ZetaApproximationScreen.setWindowTitle(_translate("ZetaApproximationScreen",
            "Visualising the Riemann Hypothesis - Investigation"))
        self.Title.setText(_translate("ZetaApproximationScreen", "Graph
            Plots"))
        self.PolarTab.setText(_translate("ZetaApproximationScreen",
            "Polar"))
        self.ZetaZeroesPlotTab.setText(_translate("ZetaApproximationScreen",
            "Zeroes"))
        self.PrimeTab.setText(_translate("ZetaApproximationScreen",
            "Prime"))
        self.ZetaApproximationLabel.setText(_translate("ZetaApproximationScreen",
            "<html><head/><body><p
            align=\"center\">Zeta<br/>Approximation</p></body></html>"))
        self.PrevButton.setText(_translate("ZetaApproximationScreen",
            "Prev"))
        self.NextButton.setText(_translate("ZetaApproximationScreen",
            "Next"))
        self.GraphButton.setText(_translate("ZetaApproximationScreen",
            "Graph"))
        self.GraphInput.setToolTip(_translate("ZetaApproximationScreen",
            "<html><head/><body><p align=\"center\">
            Re(s)</body></html>"))
        self.GraphInput.setPlaceholderText(_translate("ZetaApproximationScreen",
            "Input"))
        self.SubTitleText.setText(_translate("ZetaApproximationScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Approximation of the Riemann Zeta
            Function</span></p></body></html>"))
        self.MainText.setText(_translate("ZetaApproximationScreen",
            "<html><head/><body><p>Due to the infinite and recursive
            nature of the zeta function, it would be impossbile to
            calculate any exact values for the zeta function using a
            computer program.</p><p><br/></p><p>The aim of this graph,
            is to demonstrate how the program uses sums with increasing
            amounts of accuracy to find values of the zeta
            function.</p><p><br/></p><p>Type any complex number into the
            box, and click graph!</p></body></html>"))
        self.ErrorLabel.setText(_translate("ZetaApproximationScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/investigation_ui/zeta_zeroes_plot.py

```
"""
zeta_zeroes_plot.py
===================
A GUI for the zeta zeroes plot page of the investigation section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ZetaZeroesPlotScreen(object):

    def setupUi(self, ZetaZeroesPlotScreen):
        ZetaZeroesPlotScreen.setObjectName("ZetaZeroesPlotScreen")
        ZetaZeroesPlotScreen.resize(1340, 722)
        ZetaZeroesPlotScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ZetaZeroesPlotScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(550, 20, 271, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.PolarTab = QtWidgets.QPushButton(self.TabBar)
        self.PolarTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.PolarTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PolarTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PolarTab.setObjectName("PolarTab")
        self.ZetaZeroesPlotTab = QtWidgets.QPushButton(self.TabBar)
        self.ZetaZeroesPlotTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.ZetaZeroesPlotTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaZeroesPlotTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
```

```
        self.ZetaZeroesPlotTab.setObjectName("ZetaZeroesPlotTab")
        self.PrimeTab = QtWidgets.QPushButton(self.TabBar)
        self.PrimeTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.PrimeTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrimeTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrimeTab.setObjectName("PrimeTab")
        self.ZetaApproximationTab = QtWidgets.QPushButton(self.TabBar)
        self.ZetaApproximationTab.setGeometry(QtCore.QRect(640, 5, 200,
            70))
        self.ZetaApproximationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZetaApproximationTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ZetaApproximationTab.setText("")
        self.ZetaApproximationTab.setObjectName("ZetaApproximationTab")
        self.ZetaApproximationLabel = QtWidgets.QLabel(self.TabBar)
        self.ZetaApproximationLabel.setGeometry(QtCore.QRect(640, 5,
            200, 70))
        self.ZetaApproximationLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZetaApproximationLabel.setObjectName("ZetaApproximationLabel")
        self.PolarTab.raise_()
        self.ZetaZeroesPlotTab.raise_()
        self.PrimeTab.raise_()
        self.ZetaApproximationLabel.raise_()
        self.ZetaApproximationTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
```

```
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.GraphButton = QtWidgets.QPushButton(self.MainWidget)
        self.GraphButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.GraphButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.GraphButton.setObjectName("GraphButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 681, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 80, 1251, 141))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.ZeroesTab = QtWidgets.QPushButton(self.MainWidget)
        self.ZeroesTab.setGeometry(QtCore.QRect(640, 70, 171, 51))
        self.ZeroesTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ZeroesTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 12pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ZeroesTab.setObjectName("ZeroesTab")
        self.QuestionText = QtWidgets.QLabel(self.MainWidget)
        self.QuestionText.setGeometry(QtCore.QRect(420, 230, 501, 101))
        self.QuestionText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.QuestionText.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.QuestionText.setWordWrap(True)
        self.QuestionText.setObjectName("QuestionText")
        self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
        self.QuestionInput.setGeometry(QtCore.QRect(390, 350, 230, 60))
        self.QuestionInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(720, 350, 121, 61))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(410, 410, 530, 41))
        self.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.MessageLabel.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.MessageLabel.setObjectName("MessageLabel")
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.GraphButton.raise_()
        self.SubTitleText.raise_()
        self.MainText.raise_()
        self.ZeroesTab.raise_()
        self.QuestionText.raise_()
        self.QuestionInput.raise_()
        self.MessageLabel.raise_()
        self.SubmitButton.raise_()

        self.retranslateUi(ZetaZeroesPlotScreen)
        QtCore.QMetaObject.connectSlotsByName(ZetaZeroesPlotScreen)

    def retranslateUi(self, ZetaZeroesPlotScreen):
        _translate = QtCore.QCoreApplication.translate
        ZetaZeroesPlotScreen.setWindowTitle(_translate("ZetaZeroesPlotScreen",
            "Visualising the Riemann Hypothesis - Investigation"))
        self.Title.setText(_translate("ZetaZeroesPlotScreen", "Graph
            Plots"))
        self.PolarTab.setText(_translate("ZetaZeroesPlotScreen",
            "Polar"))
        self.ZetaZeroesPlotTab.setText(_translate("ZetaZeroesPlotScreen",
            "Zeroes"))
        self.PrimeTab.setText(_translate("ZetaZeroesPlotScreen",
            "Prime"))
        self.ZetaApproximationLabel.setText(_translate("ZetaZeroesPlotScreen",
            "<html><head/><body><p
            align=\"center\">Zeta<br/>Approximation</p></body></html>"))
        self.PrevButton.setText(_translate("ZetaZeroesPlotScreen",
            "Prev"))
        self.NextButton.setText(_translate("ZetaZeroesPlotScreen",
            "Next"))
        self.GraphButton.setText(_translate("ZetaZeroesPlotScreen",
            "Graph"))
        self.SubTitleText.setText(_translate("ZetaZeroesPlotScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Zeroes of the Riemann Zeta
            Function</span></p></body></html>"))
        self.MainText.setText(_translate("ZetaZeroesPlotScreen",
            "<html><head/><body><p>As also mentioned in the zeroes
            calculator section of the investigation
            --&gt;<br/></p><p>The non-trivial zeroes of the riemann zeta
            function are input values between 0 and 1, for which the
            output of the function is zero.</p><p>This graph will aim to
            calculate each zeta zero, and then plot them on a graph;
            allowing you to see the distribution of the non-trivial zeta
            zeroes along the critical line.</p></body></html>"))
        self.ZeroesTab.setText(_translate("ZetaZeroesPlotScreen",
            "Zeroes Calculator"))
        self.QuestionText.setText(_translate("ZetaZeroesPlotScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:16pt;
            font-weight:600;\">Question</span></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("ZetaZeroesPlotScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("ZetaZeroesPlotScreen",
            "Submit"))
        self.MessageLabel.setText(_translate("ZetaZeroesPlotScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/login_ui/__init__.py

```
"""
__init__.py
```

```
===========
Imports for the login_ui
"""


from .forgotten_password import Ui_ForgottenPasswordScreen
from .forgotten_password2 import Ui_ForgottenPassword2Screen
from .login import Ui_LoginScreen
from .reset_password import Ui_ResetPasswordScreen
from .reset_password2 import Ui_ResetPassword2Screen
from .sign_up import Ui_SignUpScreen
```

program/user_interface/login_ui/forgotten_password.py

```
"""
forgotten_password.py
=====================
A GUI for the forgotten password page of the login section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ForgottenPasswordScreen(object):

    def setupUi(self, ForgottenPasswordScreen):
        ForgottenPasswordScreen.setObjectName("ForgottenPasswordScreen")
        ForgottenPasswordScreen.resize(1340, 720)
        ForgottenPasswordScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ForgottenPasswordScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(612, 20, 116, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
```

```
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.LoginTab.setObjectName("LoginTab")
       self.SignUpTab = QtWidgets.QPushButton(self.TabBar)
       self.SignUpTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
       self.SignUpTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.SignUpTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.SignUpTab.setObjectName("SignUpTab")
       self.ResetPasswordTab = QtWidgets.QPushButton(self.TabBar)
       self.ResetPasswordTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
       self.ResetPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.ResetPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.ResetPasswordTab.setObjectName("ResetPasswordTab")
       self.ForgottenPasswordLabel = QtWidgets.QLabel(self.TabBar)
       self.ForgottenPasswordLabel.setGeometry(QtCore.QRect(430, 5,
           200, 70))
       self.ForgottenPasswordLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"padding: 7px;\n"
"")
       self.ForgottenPasswordLabel.setObjectName("ForgottenPasswordLabel")
       self.ForgottenPasswordTab = QtWidgets.QPushButton(self.TabBar)
       self.ForgottenPasswordTab.setGeometry(QtCore.QRect(430, 5, 200,
           70))
       self.ForgottenPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.ForgottenPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"border-color: rgba(0, 0, 0, 0);")
       self.ForgottenPasswordTab.setText("")
       self.ForgottenPasswordTab.setObjectName("ForgottenPasswordTab")
       self.MainWidget = QtWidgets.QWidget(self.widget)
       self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
       self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
```

```
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.EmailText = QtWidgets.QLabel(self.MainWidget)
        self.EmailText.setGeometry(QtCore.QRect(300, 130, 301, 61))
        self.EmailText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.EmailText.setObjectName("EmailText")
        self.EmailInput = QtWidgets.QLineEdit(self.MainWidget)
        self.EmailInput.setGeometry(QtCore.QRect(720, 130, 361, 60))
        self.EmailInput.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.EmailInput.setText("")
        self.EmailInput.setCursorPosition(0)
        self.EmailInput.setObjectName("EmailInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(570, 440, 200, 70))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(440, 320, 461, 71))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")

        self.retranslateUi(ForgottenPasswordScreen)
        QtCore.QMetaObject.connectSlotsByName(ForgottenPasswordScreen)

    def retranslateUi(self, ForgottenPasswordScreen):
        _translate = QtCore.QCoreApplication.translate
        ForgottenPasswordScreen.setWindowTitle(_translate("ForgottenPasswordScreen",
            "Visualising the Riemann Hypothesis - Forgotten Password"))
        self.Title.setText(_translate("ForgottenPasswordScreen",
            "Login"))
        self.LoginTab.setText(_translate("ForgottenPasswordScreen",
            "Login"))
        self.SignUpTab.setText(_translate("ForgottenPasswordScreen",
            "Sign Up"))
        self.ResetPasswordTab.setText(_translate("ForgottenPasswordScreen",
            "Reset Password"))
        self.ForgottenPasswordLabel.setText(_translate("ForgottenPasswordScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:18pt;\">Forgotten<br/>Password</span></p></body></html>"))
        self.EmailText.setText(_translate("ForgottenPasswordScreen",
            "<html><head/><body><p
            align=\"right\">Email:</p></body></html>"))
        self.EmailInput.setPlaceholderText(_translate("ForgottenPasswordScreen",
            "Enter Email"))
        self.SubmitButton.setText(_translate("ForgottenPasswordScreen",
            "Submit"))
        self.ErrorLabel.setText(_translate("ForgottenPasswordScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/login_ui/forgotten_password2.py

```
"""
forgotten_password2.py
=====================
A GUI for the forgotten password 2 page of the login section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ForgottenPassword2Screen(object):

    def setupUi(self, ForgottenPassword2Screen):
        ForgottenPassword2Screen.setObjectName("ForgottenPassword2Screen")
        ForgottenPassword2Screen.resize(1340, 720)
        ForgottenPassword2Screen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ForgottenPassword2Screen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(612, 20, 116, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.LoginTab.setObjectName("LoginTab")
        self.SignUpTab = QtWidgets.QPushButton(self.TabBar)
        self.SignUpTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.SignUpTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SignUpTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SignUpTab.setObjectName("SignUpTab")
        self.ResetPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ResetPasswordTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ResetPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ResetPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ResetPasswordTab.setObjectName("ResetPasswordTab")
        self.ForgottenPasswordLabel = QtWidgets.QLabel(self.TabBar)
        self.ForgottenPasswordLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.ForgottenPasswordLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"padding: 7px;\n"
"")
        self.ForgottenPasswordLabel.setObjectName("ForgottenPasswordLabel")
        self.ForgottenPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ForgottenPasswordTab.setGeometry(QtCore.QRect(430, 5, 200,
            70))
        self.ForgottenPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ForgottenPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"border-color: rgba(0, 0, 0, 0);")
        self.ForgottenPasswordTab.setText("")
        self.ForgottenPasswordTab.setObjectName("ForgottenPasswordTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.VerificationCodeText = QtWidgets.QLabel(self.MainWidget)
        self.VerificationCodeText.setGeometry(QtCore.QRect(300, 200,
            301, 61))
        self.VerificationCodeText.setStyleSheet("font: 25pt \"Sans
            Serif\"; color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.VerificationCodeText.setObjectName("VerificationCodeText")
        self.VerificationCodeInput = QtWidgets.QLineEdit(self.MainWidget)
        self.VerificationCodeInput.setGeometry(QtCore.QRect(720, 200,
            261, 60))
        self.VerificationCodeInput.setStyleSheet("background-color:
            rgb(239, 239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.VerificationCodeInput.setText("")
        self.VerificationCodeInput.setCursorPosition(0)
        self.VerificationCodeInput.setObjectName("VerificationCodeInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(570, 440, 200, 70))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.VerificationText = QtWidgets.QLabel(self.MainWidget)
        self.VerificationText.setGeometry(QtCore.QRect(305, 60, 730, 61))
        self.VerificationText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.VerificationText.setObjectName("VerificationText")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(440, 350, 461, 71))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")

        self.retranslateUi(ForgottenPassword2Screen)
        QtCore.QMetaObject.connectSlotsByName(ForgottenPassword2Screen)
```

```
    def retranslateUi(self, ForgottenPassword2Screen):
        _translate = QtCore.QCoreApplication.translate
        ForgottenPassword2Screen.setWindowTitle(_translate("ForgottenPassword2Screen",
            "Visualising the Riemann Hypothesis - Forgotten Password"))
        self.Title.setText(_translate("ForgottenPassword2Screen",
            "Login"))
        self.LoginTab.setText(_translate("ForgottenPassword2Screen",
            "Login"))
        self.SignUpTab.setText(_translate("ForgottenPassword2Screen",
            "Sign Up"))
        self.ResetPasswordTab.setText(_translate("ForgottenPassword2Screen",
            "Reset Password"))
        self.ForgottenPasswordLabel.setText(_translate("ForgottenPassword2Screen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:18pt;\">Forgotten<br/>Password</span></p></body></html>"))
        self.VerificationCodeText.setText(_translate("ForgottenPassword2Screen",
            "<html><head/><body><p align=\"right\">Verification
            Code:</p></body></html>"))
        self.VerificationCodeInput.setPlaceholderText(_translate("ForgottenPassword2Screen",
            "Enter Verification Code"))
        self.SubmitButton.setText(_translate("ForgottenPassword2Screen",
            "Submit"))
        self.VerificationText.setText(_translate("ForgottenPassword2Screen",
            "<html><head/><body><p align=\"center\">A Verification Code
            has been sent to your email</p></body></html>"))
        self.ErrorLabel.setText(_translate("ForgottenPassword2Screen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/login_ui/login.py

```
"""
login.py
========
A GUI for the login page of the login section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_LoginScreen(object):

    def setupUi(self, LoginScreen):
        LoginScreen.setObjectName("LoginScreen")
        LoginScreen.resize(1340, 720)
        LoginScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(LoginScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(612, 20, 116, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.LoginTab.setObjectName("LoginTab")
        self.SignUpTab = QtWidgets.QPushButton(self.TabBar)
        self.SignUpTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.SignUpTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SignUpTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"padding: 7px;\n"
"")
        self.SignUpTab.setObjectName("SignUpTab")
        self.ResetPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ResetPasswordTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ResetPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ResetPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ResetPasswordTab.setObjectName("ResetPasswordTab")
        self.ForgottenPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ForgottenPasswordTab.setGeometry(QtCore.QRect(430, 5, 200,
            70))
        self.ForgottenPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ForgottenPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"border-color: rgba(0, 0, 0, 0);")
        self.ForgottenPasswordTab.setText("")
        self.ForgottenPasswordTab.setObjectName("ForgottenPasswordTab")
        self.ForgottenPasswordLabel = QtWidgets.QLabel(self.TabBar)
        self.ForgottenPasswordLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.ForgottenPasswordLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"padding: 7px;\n"
"")
        self.ForgottenPasswordLabel.setObjectName("ForgottenPasswordLabel")
        self.LoginTab.raise_()
        self.SignUpTab.raise_()
        self.ResetPasswordTab.raise_()
        self.ForgottenPasswordLabel.raise_()
        self.ForgottenPasswordTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.UsernameOrEmailText = QtWidgets.QLabel(self.MainWidget)
        self.UsernameOrEmailText.setGeometry(QtCore.QRect(300, 130, 301,
            61))
        self.UsernameOrEmailText.setStyleSheet("font: 25pt \"Sans
            Serif\"; color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.UsernameOrEmailText.setObjectName("UsernameOrEmailText")
        self.PasswordText = QtWidgets.QLabel(self.MainWidget)
        self.PasswordText.setGeometry(QtCore.QRect(320, 280, 281, 61))
        self.PasswordText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.PasswordText.setObjectName("PasswordText")
        self.UsernameInput = QtWidgets.QLineEdit(self.MainWidget)
        self.UsernameInput.setGeometry(QtCore.QRect(720, 130, 231, 60))
        self.UsernameInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.UsernameInput.setText("")
        self.UsernameInput.setCursorPosition(0)
        self.UsernameInput.setObjectName("UsernameInput")
        self.PasswordInput = QtWidgets.QLineEdit(self.MainWidget)
        self.PasswordInput.setGeometry(QtCore.QRect(720, 280, 231, 60))
        self.PasswordInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.PasswordInput.setText("")
        self.PasswordInput.setEchoMode(QtWidgets.QLineEdit.Password)
        self.PasswordInput.setCursorPosition(0)
        self.PasswordInput.setObjectName("PasswordInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(570, 440, 200, 70))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(440, 350, 461, 71))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")
        self.ShowHideButton = QtWidgets.QPushButton(self.MainWidget)
        self.ShowHideButton.setGeometry(QtCore.QRect(980, 290, 111, 41))
        self.ShowHideButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ShowHideButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ShowHideButton.setObjectName("ShowHideButton")
        self.BackButton = QtWidgets.QPushButton(self.MainWidget)
        self.BackButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.BackButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.BackButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.BackButton.setObjectName("BackButton")

        self.retranslateUi(LoginScreen)
        QtCore.QMetaObject.connectSlotsByName(LoginScreen)

    def retranslateUi(self, LoginScreen):
        _translate = QtCore.QCoreApplication.translate
        LoginScreen.setWindowTitle(_translate("LoginScreen",
            "Visualising the Riemann Hypothesis - Log In"))
        self.Title.setText(_translate("LoginScreen", "Login"))
        self.LoginTab.setText(_translate("LoginScreen", "Login"))
        self.SignUpTab.setText(_translate("LoginScreen", "Sign Up"))
        self.ResetPasswordTab.setText(_translate("LoginScreen", "Reset
            Password"))
        self.ForgottenPasswordLabel.setText(_translate("LoginScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:18pt;\">Forgotten<br/>Password</span></p></body></html>"))
        self.UsernameOrEmailText.setText(_translate("LoginScreen",
            "<html><head/><body><p
            align=\"right\">Username:</p></body></html>"))
        self.PasswordText.setText(_translate("LoginScreen",
            "<html><head/><body><p
            align=\"right\">Password:</p></body></html>"))
        self.UsernameInput.setPlaceholderText(_translate("LoginScreen",
            "Enter Username"))
        self.PasswordInput.setPlaceholderText(_translate("LoginScreen",
            "Enter Password"))
        self.SubmitButton.setText(_translate("LoginScreen", "Submit"))
        self.ErrorLabel.setText(_translate("LoginScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        self.ShowHideButton.setText(_translate("LoginScreen", "Show"))
        self.BackButton.setText(_translate("LoginScreen", "Back"))
```

program/user_interface/login_ui/resset_password.py

```
"""
reset_password.py
=================
A GUI for the reset password page of the login section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ResetPasswordScreen(object):

    def setupUi(self, ResetPasswordScreen):
        ResetPasswordScreen.setObjectName("ResetPasswordScreen")
        ResetPasswordScreen.resize(1340, 720)
        ResetPasswordScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ResetPasswordScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(612, 20, 116, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.LoginTab.setObjectName("LoginTab")
        self.SignUpTab = QtWidgets.QPushButton(self.TabBar)
        self.SignUpTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.SignUpTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SignUpTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SignUpTab.setObjectName("SignUpTab")
        self.ResetPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ResetPasswordTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ResetPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ResetPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.ResetPasswordTab.setObjectName("ResetPasswordTab")
        self.ForgottenPasswordLabel = QtWidgets.QLabel(self.TabBar)
        self.ForgottenPasswordLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.ForgottenPasswordLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"padding: 7px;\n"
"")
        self.ForgottenPasswordLabel.setObjectName("ForgottenPasswordLabel")
        self.ForgottenPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ForgottenPasswordTab.setGeometry(QtCore.QRect(430, 5, 200,
            70))
        self.ForgottenPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ForgottenPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"border-color: rgba(0, 0, 0, 0);")
        self.ForgottenPasswordTab.setText("")
        self.ForgottenPasswordTab.setObjectName("ForgottenPasswordTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.UsernameOrEmailText = QtWidgets.QLabel(self.MainWidget)
        self.UsernameOrEmailText.setGeometry(QtCore.QRect(300, 130, 301,
            61))
        self.UsernameOrEmailText.setStyleSheet("font: 25pt \"Sans
            Serif\"; color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.UsernameOrEmailText.setObjectName("UsernameOrEmailText")
        self.PasswordText = QtWidgets.QLabel(self.MainWidget)
        self.PasswordText.setGeometry(QtCore.QRect(320, 280, 281, 61))
        self.PasswordText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.PasswordText.setObjectName("PasswordText")
        self.UsernameInput = QtWidgets.QLineEdit(self.MainWidget)
        self.UsernameInput.setGeometry(QtCore.QRect(720, 130, 261, 60))
        self.UsernameInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.UsernameInput.setText("")
        self.UsernameInput.setCursorPosition(0)
        self.UsernameInput.setObjectName("UsernameInput")
        self.PasswordInput = QtWidgets.QLineEdit(self.MainWidget)
        self.PasswordInput.setGeometry(QtCore.QRect(720, 280, 261, 60))
        self.PasswordInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
```

```
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.PasswordInput.setText("")
        self.PasswordInput.setEchoMode(QtWidgets.QLineEdit.Password)
        self.PasswordInput.setCursorPosition(0)
        self.PasswordInput.setObjectName("PasswordInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(570, 440, 200, 70))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.ShowHideButton = QtWidgets.QPushButton(self.MainWidget)
        self.ShowHideButton.setGeometry(QtCore.QRect(1010, 290, 111, 41))
        self.ShowHideButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ShowHideButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ShowHideButton.setObjectName("ShowHideButton")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(440, 350, 461, 71))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")

        self.retranslateUi(ResetPasswordScreen)
        QtCore.QMetaObject.connectSlotsByName(ResetPasswordScreen)

    def retranslateUi(self, ResetPasswordScreen):
        _translate = QtCore.QCoreApplication.translate
        ResetPasswordScreen.setWindowTitle(_translate("ResetPasswordScreen",
            "Visualising the Riemann Hypothesis - Reset Password"))
        self.Title.setText(_translate("ResetPasswordScreen", "Login"))
        self.LoginTab.setText(_translate("ResetPasswordScreen", "Login"))
        self.SignUpTab.setText(_translate("ResetPasswordScreen", "Sign
            Up"))
        self.ResetPasswordTab.setText(_translate("ResetPasswordScreen",
            "Reset Password"))
        self.ForgottenPasswordLabel.setText(_translate("ResetPasswordScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:18pt;\">Forgotten<br/>Password</span></p></body></html>"))
        self.UsernameOrEmailText.setText(_translate("ResetPasswordScreen",
            "<html><head/><body><p
            align=\"right\">Username:</p></body></html>"))
        self.PasswordText.setText(_translate("ResetPasswordScreen",
            "<html><head/><body><p
            align=\"right\">Password:</p></body></html>"))
        self.UsernameInput.setPlaceholderText(_translate("ResetPasswordScreen",
            "Enter Username"))
        self.PasswordInput.setPlaceholderText(_translate("ResetPasswordScreen",
            "Enter Password"))
        self.SubmitButton.setText(_translate("ResetPasswordScreen",
            "Submit"))
        self.ShowHideButton.setText(_translate("ResetPasswordScreen",
            "Show"))
        self.ErrorLabel.setText(_translate("ResetPasswordScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/login_ui/resset_password2.py

```
"""
reset_password2.py
=================
A GUI for the reset password 2 page of the login section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ResetPassword2Screen(object):

    def setupUi(self, ResetPassword2Screen):
        ResetPassword2Screen.setObjectName("ResetPassword2Screen")
        ResetPassword2Screen.resize(1340, 720)
        ResetPassword2Screen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ResetPassword2Screen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(612, 20, 116, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.LoginTab.setObjectName("LoginTab")
        self.SignUpTab = QtWidgets.QPushButton(self.TabBar)
        self.SignUpTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.SignUpTab.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
        self.SignUpTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SignUpTab.setObjectName("SignUpTab")
        self.ResetPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ResetPasswordTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ResetPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.ArrowCursor))
        self.ResetPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.ResetPasswordTab.setObjectName("ResetPasswordTab")
        self.ForgottenPasswordLabel = QtWidgets.QLabel(self.TabBar)
        self.ForgottenPasswordLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.ForgottenPasswordLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"padding: 7px;\n"
"")
        self.ForgottenPasswordLabel.setObjectName("ForgottenPasswordLabel")
        self.ForgottenPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ForgottenPasswordTab.setGeometry(QtCore.QRect(430, 5, 200,
            70))
        self.ForgottenPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ForgottenPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
```

```
"border-color: rgba(0, 0, 0, 0);")
        self.ForgottenPasswordTab.setText("")
        self.ForgottenPasswordTab.setObjectName("ForgottenPasswordTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PasswordText = QtWidgets.QLabel(self.MainWidget)
        self.PasswordText.setGeometry(QtCore.QRect(260, 130, 341, 61))
        self.PasswordText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.PasswordText.setObjectName("PasswordText")
        self.ConfirmPasswordText = QtWidgets.QLabel(self.MainWidget)
        self.ConfirmPasswordText.setGeometry(QtCore.QRect(210, 280, 391,
            61))
        self.ConfirmPasswordText.setStyleSheet("font: 25pt \"Sans
            Serif\"; color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.ConfirmPasswordText.setObjectName("ConfirmPasswordText")
        self.PasswordInput = QtWidgets.QLineEdit(self.MainWidget)
        self.PasswordInput.setGeometry(QtCore.QRect(720, 130, 280, 60))
        self.PasswordInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.PasswordInput.setText("")
        self.PasswordInput.setEchoMode(QtWidgets.QLineEdit.Password)
        self.PasswordInput.setCursorPosition(0)
        self.PasswordInput.setObjectName("PasswordInput")
        self.ConfirmPasswordInput = QtWidgets.QLineEdit(self.MainWidget)
        self.ConfirmPasswordInput.setGeometry(QtCore.QRect(720, 280,
            280, 60))
        self.ConfirmPasswordInput.setStyleSheet("background-color:
            rgb(239, 239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.ConfirmPasswordInput.setText("")
        self.ConfirmPasswordInput.setEchoMode(QtWidgets.QLineEdit.Password)
        self.ConfirmPasswordInput.setCursorPosition(0)
        self.ConfirmPasswordInput.setObjectName("ConfirmPasswordInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(570, 440, 200, 70))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.ShowHideButton = QtWidgets.QPushButton(self.MainWidget)
        self.ShowHideButton.setGeometry(QtCore.QRect(1030, 290, 111, 41))
        self.ShowHideButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ShowHideButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ShowHideButton.setObjectName("ShowHideButton")

        self.retranslateUi(ResetPassword2Screen)
        QtCore.QMetaObject.connectSlotsByName(ResetPassword2Screen)

    def retranslateUi(self, ResetPassword2Screen):
        _translate = QtCore.QCoreApplication.translate
        ResetPassword2Screen.setWindowTitle(_translate("ResetPassword2Screen",
            "Visualising the Riemann Hypothesis - Reset Password"))
        self.Title.setText(_translate("ResetPassword2Screen", "Login"))
        self.LoginTab.setText(_translate("ResetPassword2Screen",
            "Login"))
        self.SignUpTab.setText(_translate("ResetPassword2Screen", "Sign
            Up"))
        self.ResetPasswordTab.setText(_translate("ResetPassword2Screen",
            "Reset Password"))
        self.ForgottenPasswordLabel.setText(_translate("ResetPassword2Screen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:18pt;\">Forgotten<br/>Password</span></p></body></html>"))
        self.PasswordText.setText(_translate("ResetPassword2Screen",
            "<html><head/><body><p align=\"right\">Enter New
            Password:</p></body></html>"))
        self.ConfirmPasswordText.setText(_translate("ResetPassword2Screen",
            "<html><head/><body><p align=\"right\">Confirm New
            Password:</p></body></html>"))
        self.PasswordInput.setPlaceholderText(_translate("ResetPassword2Screen",
            "Enter New Password"))
        self.ConfirmPasswordInput.setPlaceholderText(_translate("ResetPassword2Screen",
            "Re-enter New Password"))
        self.SubmitButton.setText(_translate("ResetPassword2Screen",
            "Submit"))
        self.ShowHideButton.setText(_translate("ResetPassword2Screen",
            "Show"))
```

program/user_interface/login_ui/sign_up.py

```
"""
sign_up.py
==========
A GUI for the sign up page of the login section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_SignUpScreen(object):

    def setupUi(self, SignUpScreen):
        SignUpScreen.setObjectName("SignUpScreen")
        SignUpScreen.resize(1340, 720)
        SignUpScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(SignUpScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(612, 20, 121, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.LoginTab.setObjectName("LoginTab")
        self.SignUpTab = QtWidgets.QPushButton(self.TabBar)
        self.SignUpTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.SignUpTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SignUpTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.SignUpTab.setObjectName("SignUpTab")
        self.ResetPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ResetPasswordTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ResetPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ResetPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ResetPasswordTab.setObjectName("ResetPasswordTab")
        self.ForgottenPasswordLabel = QtWidgets.QLabel(self.TabBar)
        self.ForgottenPasswordLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.ForgottenPasswordLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"padding: 7px;\n"
"")
        self.ForgottenPasswordLabel.setObjectName("ForgottenPasswordLabel")
        self.ForgottenPasswordTab = QtWidgets.QPushButton(self.TabBar)
        self.ForgottenPasswordTab.setGeometry(QtCore.QRect(430, 5, 200,
            70))
        self.ForgottenPasswordTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ForgottenPasswordTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"border-color: rgba(0, 0, 0, 0);")
        self.ForgottenPasswordTab.setText("")
        self.ForgottenPasswordTab.setObjectName("ForgottenPasswordTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.UsernameOrEmailText = QtWidgets.QLabel(self.MainWidget)
        self.UsernameOrEmailText.setGeometry(QtCore.QRect(270, 40, 301,
            61))
        self.UsernameOrEmailText.setStyleSheet("font: 25pt \"Sans
            Serif\"; color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.UsernameOrEmailText.setObjectName("UsernameOrEmailText")
        self.EmailText = QtWidgets.QLabel(self.MainWidget)
        self.EmailText.setGeometry(QtCore.QRect(290, 120, 281, 61))
        self.EmailText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.EmailText.setObjectName("EmailText")
        self.UsernameInput = QtWidgets.QLineEdit(self.MainWidget)
        self.UsernameInput.setGeometry(QtCore.QRect(680, 40, 231, 60))
        self.UsernameInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.UsernameInput.setText("")
        self.UsernameInput.setCursorPosition(0)
        self.UsernameInput.setObjectName("UsernameInput")
        self.EmailInput = QtWidgets.QLineEdit(self.MainWidget)
        self.EmailInput.setGeometry(QtCore.QRect(680, 120, 231, 60))
        self.EmailInput.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.EmailInput.setText("")
        self.EmailInput.setEchoMode(QtWidgets.QLineEdit.Normal)
        self.EmailInput.setCursorPosition(0)
        self.EmailInput.setObjectName("EmailInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(570, 440, 200, 70))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.PasswordText = QtWidgets.QLabel(self.MainWidget)
        self.PasswordText.setGeometry(QtCore.QRect(290, 200, 281, 61))
        self.PasswordText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.PasswordText.setObjectName("PasswordText")
        self.PasswordInput = QtWidgets.QLineEdit(self.MainWidget)
        self.PasswordInput.setGeometry(QtCore.QRect(680, 200, 231, 60))
        self.PasswordInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
```

```
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.PasswordInput.setText("")
        self.PasswordInput.setEchoMode(QtWidgets.QLineEdit.Password)
        self.PasswordInput.setCursorPosition(0)
        self.PasswordInput.setObjectName("PasswordInput")
        self.PasswordText_2 = QtWidgets.QLabel(self.MainWidget)
        self.PasswordText_2.setGeometry(QtCore.QRect(260, 280, 311, 61))
        self.PasswordText_2.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.PasswordText_2.setObjectName("PasswordText_2")
        self.PasswordInput_2 = QtWidgets.QLineEdit(self.MainWidget)
        self.PasswordInput_2.setGeometry(QtCore.QRect(680, 280, 231, 60))
        self.PasswordInput_2.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.PasswordInput_2.setText("")
        self.PasswordInput_2.setEchoMode(QtWidgets.QLineEdit.Password)
        self.PasswordInput_2.setCursorPosition(0)
        self.PasswordInput_2.setObjectName("PasswordInput_2")
        self.ErrorLabel = QtWidgets.QLabel(self.MainWidget)
        self.ErrorLabel.setGeometry(QtCore.QRect(365, 350, 611, 71))
        self.ErrorLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.ErrorLabel.setObjectName("ErrorLabel")
        self.ShowHideButton_2 = QtWidgets.QPushButton(self.MainWidget)
        self.ShowHideButton_2.setGeometry(QtCore.QRect(940, 290, 111,
            41))
        self.ShowHideButton_2.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ShowHideButton_2.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ShowHideButton_2.setObjectName("ShowHideButton_2")
        self.ShowHideButton = QtWidgets.QPushButton(self.MainWidget)
        self.ShowHideButton.setGeometry(QtCore.QRect(940, 210, 111, 41))
        self.ShowHideButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ShowHideButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ShowHideButton.setObjectName("ShowHideButton")

        self.retranslateUi(SignUpScreen)
        QtCore.QMetaObject.connectSlotsByName(SignUpScreen)

    def retranslateUi(self, SignUpScreen):
        _translate = QtCore.QCoreApplication.translate
        SignUpScreen.setWindowTitle(_translate("SignUpScreen",
            "Visualising the Riemann Hypothesis - Sign Up"))
        self.Title.setText(_translate("SignUpScreen", "Login"))
        self.LoginTab.setText(_translate("SignUpScreen", "Login"))
        self.SignUpTab.setText(_translate("SignUpScreen", "Sign Up"))
        self.ResetPasswordTab.setText(_translate("SignUpScreen", "Reset
            Password"))
        self.ForgottenPasswordLabel.setText(_translate("SignUpScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:18pt;\">Forgotten<br/>Password</span></p></body></html>"))
        self.UsernameOrEmailText.setText(_translate("SignUpScreen",
            "<html><head/><body><p
            align=\"right\">Username:</p></body></html>"))
        self.EmailText.setText(_translate("SignUpScreen",
            "<html><head/><body><p
            align=\"right\">Email:</p></body></html>"))
        self.UsernameInput.setPlaceholderText(_translate("SignUpScreen",
            "Enter Username"))
        self.EmailInput.setPlaceholderText(_translate("SignUpScreen",
            "Enter Email"))
        self.SubmitButton.setText(_translate("SignUpScreen", "Submit"))
        self.PasswordText.setText(_translate("SignUpScreen",
            "<html><head/><body><p
            align=\"right\">Password:</p></body></html>"))
        self.PasswordInput.setPlaceholderText(_translate("SignUpScreen",
            "Enter Password"))
        self.PasswordText_2.setText(_translate("SignUpScreen",
            "<html><head/><body><p align=\"right\">Confirm
            Password:</p></body></html>"))
        self.PasswordInput_2.setPlaceholderText(_translate("SignUpScreen",
            "Re-enter Password"))
        self.ErrorLabel.setText(_translate("SignUpScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        self.ShowHideButton_2.setText(_translate("SignUpScreen", "Show"))
        self.ShowHideButton.setText(_translate("SignUpScreen", "Show"))
```

program/user_interface/notes_ui/__init__.py

```
"""
__init__.py
===========
Imports for the notes_ui
"""


from .tutorial_notes import Ui_TutorialNotesScreen
from .introduction_notes import Ui_IntroductionNotesScreen
from .investigation_notes import Ui_InvestigationNotesScreen
from .summary_notes import Ui_SummaryNotesScreen
```

program/user_interface/notes_ui/introduction_notes.py

```
"""
introduction_notes.py
=====================
A GUI for the introduction notes page of the notes section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_IntroductionNotesScreen(object):

    def setupUi(self, IntroductionNotesScreen):
        IntroductionNotesScreen.setObjectName("IntroductionNotesScreen")
        IntroductionNotesScreen.resize(1340, 723)
        IntroductionNotesScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(IntroductionNotesScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(605, 20, 131, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(69, 69, 69);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
 "")
        self.TutorialTab.setObjectName("TutorialTab")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(69, 69, 69);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
 "")
        self.SummaryTab.setObjectName("SummaryTab")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(239, 239, 239);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
 "")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(69, 69, 69);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
 "")
        self.InvestigationTab.setObjectName("InvestigationTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
 "border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.BackButton = QtWidgets.QPushButton(self.MainWidget)
        self.BackButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.BackButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.BackButton.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(69, 69, 69);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
 "")
```

```
        self.BackButton.setObjectName("BackButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 231, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.NotesText = QtWidgets.QTextEdit(self.MainWidget)
        self.NotesText.setGeometry(QtCore.QRect(50, 80, 1211, 371))
        self.NotesText.viewport().setProperty("cursor",
            QtGui.QCursor(QtCore.Qt.IBeamCursor))
        self.NotesText.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-color:rgb(69, 69, 69);\n"
"border-radius: 0px;")
        self.NotesText.setReadOnly(False)
        self.NotesText.setObjectName("NotesText")
        self.SaveButton = QtWidgets.QPushButton(self.MainWidget)
        self.SaveButton.setGeometry(QtCore.QRect(470, 460, 200, 70))
        self.SaveButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SaveButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SaveButton.setObjectName("SaveButton")
        self.SavedText = QtWidgets.QLabel(self.MainWidget)
        self.SavedText.setGeometry(QtCore.QRect(690, 465, 191, 61))
        self.SavedText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"border-radius: 0px")
        self.SavedText.setObjectName("SavedText")

        self.retranslateUi(IntroductionNotesScreen)
        QtCore.QMetaObject.connectSlotsByName(IntroductionNotesScreen)
```

```
    def retranslateUi(self, IntroductionNotesScreen):
        _translate = QtCore.QCoreApplication.translate
        IntroductionNotesScreen.setWindowTitle(_translate("IntroductionNotesScreen",
            "Visualising the Riemann Hypothesis - Notes"))
        self.Title.setText(_translate("IntroductionNotesScreen",
            "Notes"))
        self.TutorialTab.setText(_translate("IntroductionNotesScreen",
            "Tutorial"))
        self.SummaryTab.setText(_translate("IntroductionNotesScreen",
            "Summary"))
        self.IntroductionTab.setText(_translate("IntroductionNotesScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("IntroductionNotesScreen",
            "Investigation"))
        self.BackButton.setText(_translate("IntroductionNotesScreen",
            "Back"))
        self.NextButton.setText(_translate("IntroductionNotesScreen",
            "Next"))
        self.SubTitleText.setText(_translate("IntroductionNotesScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Introduction</span></p></body></html>"))
        self.NotesText.setHtml(_translate("IntroductionNotesScreen",
            "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\"
            \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
    type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Sans Serif\';
    font-size:18pt; font-weight:400; font-style:normal;\">\n"
"<p style=\"-qt-paragraph-type:empty; margin-top:0px; margin-bottom:0px;
    margin-left:0px; margin-right:0px; -qt-block-indent:0;
    text-indent:0px;\"><br /></p></body></html>"))
        self.SaveButton.setText(_translate("IntroductionNotesScreen",
            "Save"))
        self.SavedText.setText(_translate("IntroductionNotesScreen",
            "<html><head/><body><p><br/></p></body></html>"))
```

program/user_interface/notes_ui/investigation_notes.py

```
"""
investigation_notes.py
======================
A GUI for the investigation notes page of the notes section
"""


from PyQt5 import QtCore, QtGui, QtWidgets
```

```
class Ui_InvestigationNotesScreen(object):

    def setupUi(self, InvestigationNotesScreen):
        InvestigationNotesScreen.setObjectName("InvestigationNotesScreen")
        InvestigationNotesScreen.resize(1340, 723)
        InvestigationNotesScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(InvestigationNotesScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(605, 20, 131, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.TutorialTab.setObjectName("TutorialTab")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SummaryTab.setObjectName("SummaryTab")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
```

```
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.InvestigationTab.setObjectName("InvestigationTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.BackButton = QtWidgets.QPushButton(self.MainWidget)
        self.BackButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.BackButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.BackButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.BackButton.setObjectName("BackButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 241, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.NotesText = QtWidgets.QTextEdit(self.MainWidget)
        self.NotesText.setGeometry(QtCore.QRect(50, 80, 1211, 371))
        self.NotesText.viewport().setProperty("cursor",
            QtGui.QCursor(QtCore.Qt.IBeamCursor))
        self.NotesText.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
```

```
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-color:rgb(69, 69, 69);\n"
"border-radius: 0px;")
        self.NotesText.setReadOnly(False)
        self.NotesText.setObjectName("NotesText")
        self.SaveButton = QtWidgets.QPushButton(self.MainWidget)
        self.SaveButton.setGeometry(QtCore.QRect(470, 460, 200, 70))
        self.SaveButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SaveButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SaveButton.setObjectName("SaveButton")
        self.SavedText = QtWidgets.QLabel(self.MainWidget)
        self.SavedText.setGeometry(QtCore.QRect(690, 465, 191, 61))
        self.SavedText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"border-radius: 0px")
        self.SavedText.setObjectName("SavedText")

        self.retranslateUi(InvestigationNotesScreen)
        QtCore.QMetaObject.connectSlotsByName(InvestigationNotesScreen)

    def retranslateUi(self, InvestigationNotesScreen):
        _translate = QtCore.QCoreApplication.translate
        InvestigationNotesScreen.setWindowTitle(_translate("InvestigationNotesScreen",
            "Visualising the Riemann Hypothesis - Notes"))
        self.Title.setText(_translate("InvestigationNotesScreen",
            "Notes"))
        self.TutorialTab.setText(_translate("InvestigationNotesScreen",
            "Tutorial"))
        self.SummaryTab.setText(_translate("InvestigationNotesScreen",
            "Summary"))
        self.IntroductionTab.setText(_translate("InvestigationNotesScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("InvestigationNotesScreen",
            "Investigation"))
        self.BackButton.setText(_translate("InvestigationNotesScreen",
            "Back"))
        self.NextButton.setText(_translate("InvestigationNotesScreen",
            "Next"))
        self.SubTitleText.setText(_translate("InvestigationNotesScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Investigation</span></p></body></html>"))
        self.NotesText.setHtml(_translate("InvestigationNotesScreen",
            "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\"
```

```
           \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
    type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Sans Serif\';
    font-size:18pt; font-weight:400; font-style:normal;\">\n"
"<p style=\"-qt-paragraph-type:empty; margin-top:0px; margin-bottom:0px;
    margin-left:0px; margin-right:0px; -qt-block-indent:0;
    text-indent:0px;\"><br /></p></body></html>"))
       self.SaveButton.setText(_translate("InvestigationNotesScreen",
           "Save"))
       self.SavedText.setText(_translate("InvestigationNotesScreen",
           "<html><head/><body><p><br/></p></body></html>"))
```

program/user_interface/notes_ui/summary_notes.py

```
"""
summary_notes.py
================
A GUI for the summary notes page of the notes section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_SummaryNotesScreen(object):

    def setupUi(self, SummaryNotesScreen):
        SummaryNotesScreen.setObjectName("SummaryNotesScreen")
        SummaryNotesScreen.resize(1340, 723)
        SummaryNotesScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(SummaryNotesScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(605, 20, 131, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
```

```
       self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"\n"
"")
       self.TutorialTab.setObjectName("TutorialTab")
       self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
       self.SummaryTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
       self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
       self.SummaryTab.setObjectName("SummaryTab")
       self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
       self.IntroductionTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
       self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.IntroductionTab.setObjectName("IntroductionTab")
       self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
       self.InvestigationTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
       self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.InvestigationTab.setObjectName("InvestigationTab")
       self.MainWidget = QtWidgets.QWidget(self.widget)
       self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
       self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
"border-radius: 20px;")
       self.MainWidget.setObjectName("MainWidget")
       self.BackButton = QtWidgets.QPushButton(self.MainWidget)
       self.BackButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
       self.BackButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.BackButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.BackButton.setObjectName("BackButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 181, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.NotesText = QtWidgets.QTextEdit(self.MainWidget)
        self.NotesText.setGeometry(QtCore.QRect(50, 80, 1211, 371))
        self.NotesText.viewport().setProperty("cursor",
            QtGui.QCursor(QtCore.Qt.IBeamCursor))
        self.NotesText.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-color:rgb(69, 69, 69);\n"
"border-radius: 0px;")
        self.NotesText.setReadOnly(False)
        self.NotesText.setObjectName("NotesText")
        self.SaveButton = QtWidgets.QPushButton(self.MainWidget)
        self.SaveButton.setGeometry(QtCore.QRect(470, 460, 200, 70))
        self.SaveButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SaveButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SaveButton.setObjectName("SaveButton")
        self.SavedText = QtWidgets.QLabel(self.MainWidget)
        self.SavedText.setGeometry(QtCore.QRect(690, 465, 191, 61))
        self.SavedText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"border-radius: 0px")
        self.SavedText.setObjectName("SavedText")

        self.retranslateUi(SummaryNotesScreen)
        QtCore.QMetaObject.connectSlotsByName(SummaryNotesScreen)

    def retranslateUi(self, SummaryNotesScreen):
        _translate = QtCore.QCoreApplication.translate
        SummaryNotesScreen.setWindowTitle(_translate("SummaryNotesScreen",
            "Visualising the Riemann Hypothesis - Notes"))
        self.Title.setText(_translate("SummaryNotesScreen", "Notes"))
        self.TutorialTab.setText(_translate("SummaryNotesScreen",
            "Tutorial"))
        self.SummaryTab.setText(_translate("SummaryNotesScreen",
            "Summary"))
        self.IntroductionTab.setText(_translate("SummaryNotesScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("SummaryNotesScreen",
            "Investigation"))
        self.BackButton.setText(_translate("SummaryNotesScreen", "Back"))
        self.SubTitleText.setText(_translate("SummaryNotesScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Summary</span></p><p><br/></p></body></html>"))
        self.NotesText.setHtml(_translate("SummaryNotesScreen",
            "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\"
            \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
    type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Sans Serif\';
    font-size:18pt; font-weight:400; font-style:normal;\">\n"
"<p style=\"-qt-paragraph-type:empty; margin-top:0px; margin-bottom:0px;
    margin-left:0px; margin-right:0px; -qt-block-indent:0;
    text-indent:0px;\"><br /></p></body></html>"))
        self.SaveButton.setText(_translate("SummaryNotesScreen", "Save"))
        self.SavedText.setText(_translate("SummaryNotesScreen",
            "<html><head/><body><p><br/></p></body></html>"))
```

program/user_interface/notes_ui/tutorial_notes.py

```
"""
tutorial_notes.py
=================
A GUI for the tutorial notes page of the notes section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_TutorialNotesScreen(object):

    def setupUi(self, TutorialNotesScreen):
        TutorialNotesScreen.setObjectName("TutorialNotesScreen")
        TutorialNotesScreen.resize(1340, 723)
        TutorialNotesScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(TutorialNotesScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(605, 20, 131, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.TutorialTab.setObjectName("TutorialTab")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SummaryTab.setObjectName("SummaryTab")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationTab.setGeometry(QtCore.QRect(430, 5, 200, 70))
        self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.InvestigationTab.setObjectName("InvestigationTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.BackButton = QtWidgets.QPushButton(self.MainWidget)
        self.BackButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.BackButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.BackButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.BackButton.setObjectName("BackButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 151, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.NotesText = QtWidgets.QTextEdit(self.MainWidget)
        self.NotesText.setGeometry(QtCore.QRect(50, 80, 1211, 371))
        self.NotesText.viewport().setProperty("cursor",
            QtGui.QCursor(QtCore.Qt.IBeamCursor))
        self.NotesText.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-color:rgb(69, 69, 69);\n"
"border-radius: 0px;")
        self.NotesText.setReadOnly(False)
        self.NotesText.setObjectName("NotesText")
        self.SaveButton = QtWidgets.QPushButton(self.MainWidget)
        self.SaveButton.setGeometry(QtCore.QRect(470, 460, 200, 70))
        self.SaveButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SaveButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SaveButton.setObjectName("SaveButton")
        self.SavedText = QtWidgets.QLabel(self.MainWidget)
        self.SavedText.setGeometry(QtCore.QRect(690, 465, 191, 61))
        self.SavedText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;\n"
"border-radius: 0px")
        self.SavedText.setObjectName("SavedText")

        self.retranslateUi(TutorialNotesScreen)
        QtCore.QMetaObject.connectSlotsByName(TutorialNotesScreen)

    def retranslateUi(self, TutorialNotesScreen):
        _translate = QtCore.QCoreApplication.translate
        TutorialNotesScreen.setWindowTitle(_translate("TutorialNotesScreen",
            "Visualising the Riemann Hypothesis - Notes"))
        self.Title.setText(_translate("TutorialNotesScreen", "Notes"))
        self.TutorialTab.setText(_translate("TutorialNotesScreen",
            "Tutorial"))
        self.SummaryTab.setText(_translate("TutorialNotesScreen",
            "Summary"))
        self.IntroductionTab.setText(_translate("TutorialNotesScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("TutorialNotesScreen",
            "Investigation"))
        self.BackButton.setText(_translate("TutorialNotesScreen",
            "Back"))
        self.NextButton.setText(_translate("TutorialNotesScreen",
            "Next"))
        self.SubTitleText.setText(_translate("TutorialNotesScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Tutorial</span></p></body></html>"))
        self.NotesText.setHtml(_translate("TutorialNotesScreen",
            "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0//EN\"
            \"http://www.w3.org/TR/REC-html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
    type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'Sans Serif\';
    font-size:18pt; font-weight:400; font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px;
    margin-right:0px; -qt-block-indent:0;
    text-indent:0px;\">Text1</p></body></html>"))
        self.SaveButton.setText(_translate("TutorialNotesScreen",
            "Save"))
        self.SavedText.setText(_translate("TutorialNotesScreen",
            "<html><head/><body><p><br/></p></body></html>"))
```

program/user_interface/summary_ui/__init__.py

```
"""
__init__.py
===========
Imports for the summary_ui
"""


from .summary import Ui_SummaryScreen
from .theory_recap import Ui_TheoryRecapScreen
from .investigation_results import Ui_InvestigationResultsScreen
from .conclusion import Ui_ConclusionScreen
from .impact import Ui_ImpactScreen
```

program/user_interface/summary_ui/conclusion.py

```
"""
conclusion.py
=================
A GUI for the conclusion page of the summary section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ConclusionScreen(object):

    def setupUi(self, ConclusionScreen):
        ConclusionScreen.setObjectName("ConclusionScreen")
        ConclusionScreen.resize(1340, 723)
        ConclusionScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ConclusionScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(565, 20, 211, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SummaryTab.setObjectName("SummaryTab")
        self.TheoryRecapTab = QtWidgets.QPushButton(self.TabBar)
        self.TheoryRecapTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.TheoryRecapTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TheoryRecapTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.TheoryRecapTab.setObjectName("TheoryRecapTab")
        self.InvestigationResultsLabel = QtWidgets.QLabel(self.TabBar)
        self.InvestigationResultsLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsLabel.setStyleSheet("border: 2px
            solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.InvestigationResultsLabel.setObjectName("InvestigationResultsLabel")
        self.InvestigationResultsTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationResultsTab.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationResultsTab.setStyleSheet("border-radius:
            20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.InvestigationResultsTab.setText("")
        self.InvestigationResultsTab.setObjectName("InvestigationResultsTab")
        self.ConclusionLabel = QtWidgets.QLabel(self.TabBar)
        self.ConclusionLabel.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ConclusionLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
```

```
        self.ConclusionLabel.setObjectName("ConclusionLabel")
        self.ImpactLabel = QtWidgets.QLabel(self.TabBar)
        self.ImpactLabel.setGeometry(QtCore.QRect(850, 5, 241, 70))
        self.ImpactLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ImpactLabel.setObjectName("ImpactLabel")
        self.ConclusionTab = QtWidgets.QPushButton(self.TabBar)
        self.ConclusionTab.setGeometry(QtCore.QRect(650, 5, 200, 70))
        self.ConclusionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ConclusionTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ConclusionTab.setText("")
        self.ConclusionTab.setObjectName("ConclusionTab")
        self.ImpactTab = QtWidgets.QPushButton(self.TabBar)
        self.ImpactTab.setGeometry(QtCore.QRect(850, 5, 241, 70))
        self.ImpactTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ImpactTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ImpactTab.setText("")
        self.ImpactTab.setObjectName("ImpactTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 421, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 90, 1251, 341))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")

        self.retranslateUi(ConclusionScreen)
        QtCore.QMetaObject.connectSlotsByName(ConclusionScreen)

    def retranslateUi(self, ConclusionScreen):
        _translate = QtCore.QCoreApplication.translate
        ConclusionScreen.setWindowTitle(_translate("ConclusionScreen",
            "Visualising the Riemann Hypothesis - Summary"))
        self.Title.setText(_translate("ConclusionScreen", "Summary"))
        self.SummaryTab.setText(_translate("ConclusionScreen",
            "Summary"))
        self.TheoryRecapTab.setText(_translate("ConclusionScreen",
            "Theory Recap"))
        self.InvestigationResultsLabel.setText(_translate("ConclusionScreen",
            "<html><head/><body><p
            align=\"center\">Investigation<br/>Results</p></body></html>"))
        self.ConclusionLabel.setText(_translate("ConclusionScreen",
            "<html><head/><body><p align=\"center\">Conclusion &
            <br/>Evaluation</p></body></html>"))
        self.ImpactLabel.setText(_translate("ConclusionScreen",
            "<html><head/><body><p align=\"center\">Impact of the
            <br/>Riemann Hypothesis</p></body></html>"))
        self.PrevButton.setText(_translate("ConclusionScreen", "Prev"))
        self.NextButton.setText(_translate("ConclusionScreen", "Next"))
        self.SubTitleText.setText(_translate("ConclusionScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Conclusion &amp;
            Evaluation</span></p></body></html>"))
        self.MainText.setText(_translate("ConclusionScreen",
            "<html><head/><body><p>Unfortunately, due to the fact that
            there are an infinite number of zeta zeroes, one could not
            prove the Riemann Hypothesis by simply trying to calculate
            every single zero. However, if a zero is calculated, where
            the real part of the input is not equal to 1/2, then this
            would instantly disprove the Riemann
            Hypothesis.</p><p><br/></p><p>However, disproving the
            Riemann Hypothesis would be quite the task, seeing as there
            are an infinite amount of numbers that you would need to try
            to possibly find a zero that does not comply with the
            hypothesis. <br/></p><p>Hopefully, by using this program,
            you have discovered zeta zeroes where the real part of the
            input is 1/2. This at least reinforces that idea that
            Riemann was correct with his conjecture, although it is by
            no means a solid proof.</p></body></html>"))
        self.NotesButton.setText(_translate("ConclusionScreen", "Notes"))
```

program/user_interface/summary_ui/impact.py

```
"""
impact.py
=========
A GUI for the impact page of the summary section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ImpactScreen(object):

    def setupUi(self, ImpactScreen):
        ImpactScreen.setObjectName("ImpactScreen")
        ImpactScreen.resize(1340, 723)
        ImpactScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ImpactScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(565, 20, 211, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SummaryTab.setObjectName("SummaryTab")
        self.TheoryRecapTab = QtWidgets.QPushButton(self.TabBar)
        self.TheoryRecapTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.TheoryRecapTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TheoryRecapTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.TheoryRecapTab.setObjectName("TheoryRecapTab")
        self.InvestigationResultsLabel = QtWidgets.QLabel(self.TabBar)
        self.InvestigationResultsLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsLabel.setStyleSheet("border: 2px
            solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.InvestigationResultsLabel.setObjectName("InvestigationResultsLabel")
        self.InvestigationResultsTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationResultsTab.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationResultsTab.setStyleSheet("border-radius:
            20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.InvestigationResultsTab.setText("")
        self.InvestigationResultsTab.setObjectName("InvestigationResultsTab")
        self.ConclusionLabel = QtWidgets.QLabel(self.TabBar)
        self.ConclusionLabel.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ConclusionLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ConclusionLabel.setObjectName("ConclusionLabel")
        self.ImpactLabel = QtWidgets.QLabel(self.TabBar)
        self.ImpactLabel.setGeometry(QtCore.QRect(850, 5, 241, 70))
        self.ImpactLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.ImpactLabel.setObjectName("ImpactLabel")
        self.ConclusionTab = QtWidgets.QPushButton(self.TabBar)
        self.ConclusionTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ConclusionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ConclusionTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ConclusionTab.setText("")
        self.ConclusionTab.setObjectName("ConclusionTab")
        self.ImpactTab = QtWidgets.QPushButton(self.TabBar)
        self.ImpactTab.setGeometry(QtCore.QRect(850, 5, 241, 70))
        self.ImpactTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ImpactTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ImpactTab.setText("")
        self.ImpactTab.setObjectName("ImpactTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
```

```
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 611, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 90, 1251, 191))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")
        self.QuestionText = QtWidgets.QLabel(self.MainWidget)
        self.QuestionText.setGeometry(QtCore.QRect(420, 240, 501, 81))
        self.QuestionText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.QuestionText.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionText.setWordWrap(True)
        self.QuestionText.setObjectName("QuestionText")
        self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
        self.QuestionInput.setGeometry(QtCore.QRect(410, 330, 230, 60))
        self.QuestionInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
```

```
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(700, 330, 121, 61))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(410, 400, 530, 41))
        self.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.MessageLabel.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.MessageLabel.setObjectName("MessageLabel")

        self.retranslateUi(ImpactScreen)
        QtCore.QMetaObject.connectSlotsByName(ImpactScreen)

    def retranslateUi(self, ImpactScreen):
        _translate = QtCore.QCoreApplication.translate
        ImpactScreen.setWindowTitle(_translate("ImpactScreen",
            "Visualising the Riemann Hypothesis - Summary"))
        self.Title.setText(_translate("ImpactScreen", "Summary"))
        self.SummaryTab.setText(_translate("ImpactScreen", "Summary"))
        self.TheoryRecapTab.setText(_translate("ImpactScreen", "Theory
            Recap"))
        self.InvestigationResultsLabel.setText(_translate("ImpactScreen",
            "<html><head/><body><p
            align=\"center\">Investigation<br/>Results</p></body></html>"))
        self.ConclusionLabel.setText(_translate("ImpactScreen",
            "<html><head/><body><p align=\"center\">Conclusion &
            <br/>Evaluation</p></body></html>"))
        self.ImpactLabel.setText(_translate("ImpactScreen",
            "<html><head/><body><p align=\"center\">Impact of the
            <br/>Riemann Hypothesis</p></body></html>"))
        self.PrevButton.setText(_translate("ImpactScreen", "Prev"))
        self.NextButton.setText(_translate("ImpactScreen", "Next"))
        self.SubTitleText.setText(_translate("ImpactScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Impact of the Riemann
            Hypothesis</span></p></body></html>"))
        self.MainText.setText(_translate("ImpactScreen",
            "<html><head/><body><p>The Riemann Hypothesis is fundamental
            to the way we think about prime numbers. Although studying a
            single function may seem futile and even pointless, if this
            conjecture was proven to be true, it would be one of the
            most significant mathematical events to occur.<br/></p><p>It
            would radically change how prime numbers can be calculated
            and significantly increase our understanding of how prime
            numbers are distributed.<br/></p><p>As previously mentioned,
            this would affect fields such as crypotgraphy, and even
            quantum physics, completely revolutionising the way we view
            prime numbers.</p></body></html>"))
        self.NotesButton.setText(_translate("ImpactScreen", "Notes"))
        self.QuestionText.setText(_translate("ImpactScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:16pt;
            font-weight:600;\">Question</span></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("ImpactScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("ImpactScreen", "Submit"))
        self.MessageLabel.setText(_translate("ImpactScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/summary_ui/investigation_results.py

```
"""
investigation_results.py
========================
A GUI for the investigation results page of the summary section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_InvestigationResultsScreen(object):
    def setupUi(self, InvestigationResultsScreen):
        InvestigationResultsScreen.setObjectName("InvestigationResultsScreen")
        InvestigationResultsScreen.resize(1340, 723)
        InvestigationResultsScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(InvestigationResultsScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(565, 20, 211, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SummaryTab.setObjectName("SummaryTab")
        self.TheoryRecapTab = QtWidgets.QPushButton(self.TabBar)
        self.TheoryRecapTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.TheoryRecapTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TheoryRecapTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.TheoryRecapTab.setObjectName("TheoryRecapTab")
        self.InvestigationResultsLabel = QtWidgets.QLabel(self.TabBar)
        self.InvestigationResultsLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsLabel.setStyleSheet("border: 2px
            solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.InvestigationResultsLabel.setObjectName("InvestigationResultsLabel")
        self.InvestigationResultsTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationResultsTab.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationResultsTab.setStyleSheet("border-radius:
            20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.InvestigationResultsTab.setText("")
        self.InvestigationResultsTab.setObjectName("InvestigationResultsTab")
        self.ConclusionLabel = QtWidgets.QLabel(self.TabBar)
        self.ConclusionLabel.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ConclusionLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.ConclusionLabel.setObjectName("ConclusionLabel")
       self.ImpactLabel = QtWidgets.QLabel(self.TabBar)
       self.ImpactLabel.setGeometry(QtCore.QRect(850, 5, 241, 70))
       self.ImpactLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.ImpactLabel.setObjectName("ImpactLabel")
       self.ConclusionTab = QtWidgets.QPushButton(self.TabBar)
       self.ConclusionTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
       self.ConclusionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.ConclusionTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
       self.ConclusionTab.setText("")
       self.ConclusionTab.setObjectName("ConclusionTab")
       self.ImpactTab = QtWidgets.QPushButton(self.TabBar)
       self.ImpactTab.setGeometry(QtCore.QRect(850, 5, 241, 70))
       self.ImpactTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.ImpactTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
       self.ImpactTab.setText("")
       self.ImpactTab.setObjectName("ImpactTab")
       self.MainWidget = QtWidgets.QWidget(self.widget)
       self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
       self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
"border-radius: 20px;")
       self.MainWidget.setObjectName("MainWidget")
       self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
       self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
       self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.PrevButton.setObjectName("PrevButton")
       self.NextButton = QtWidgets.QPushButton(self.MainWidget)
```

```
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 361, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 90, 711, 341))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.ZetaTable = QtWidgets.QTableWidget(self.MainWidget)
        self.ZetaTable.setGeometry(QtCore.QRect(900, 70, 300, 351))
        self.ZetaTable.setObjectName("ZetaTable")
        self.ZetaTable.setColumnCount(2)
        self.ZetaTable.setRowCount(0)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.ZetaTable.setHorizontalHeaderItem(0, item)
        item = QtWidgets.QTableWidgetItem()
        item.setTextAlignment(QtCore.Qt.AlignCenter)
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        item.setFont(font)
        self.ZetaTable.setHorizontalHeaderItem(1, item)
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")

        self.retranslateUi(InvestigationResultsScreen)
        QtCore.QMetaObject.connectSlotsByName(InvestigationResultsScreen)

    def retranslateUi(self, InvestigationResultsScreen):
        _translate = QtCore.QCoreApplication.translate
        InvestigationResultsScreen.setWindowTitle(_translate("InvestigationResultsScreen",
            "Visualising the Riemann Hypothesis - Summary"))
        self.Title.setText(_translate("InvestigationResultsScreen",
            "Summary"))
        self.SummaryTab.setText(_translate("InvestigationResultsScreen",
            "Summary"))
        self.TheoryRecapTab.setText(_translate("InvestigationResultsScreen",
            "Theory Recap"))
        self.InvestigationResultsLabel.setText(_translate("InvestigationResultsScreen",
            "<html><head/><body><p
            align=\"center\">Investigation<br/>Results</p></body></html>"))
        self.ConclusionLabel.setText(_translate("InvestigationResultsScreen",
            "<html><head/><body><p align=\"center\">Conclusion &
            <br/>Evaluation</p></body></html>"))
        self.ImpactLabel.setText(_translate("InvestigationResultsScreen",
            "<html><head/><body><p align=\"center\">Impact of the
            <br/>Riemann Hypothesis</p></body></html>"))
        self.PrevButton.setText(_translate("InvestigationResultsScreen",
            "Prev"))
        self.NextButton.setText(_translate("InvestigationResultsScreen",
            "Next"))
        self.SubTitleText.setText(_translate("InvestigationResultsScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Investigation
            Results</span></p></body></html>"))
        self.MainText.setText(_translate("InvestigationResultsScreen",
            "<html><head/><body><p>Hopefully, throuhout this program you
            have been able to gather and record results from
            investigating the Riemann Hypothesis.</p><p>You should
            notice, that the zeroes of the Riemann Zeta function occur
            only when the real part of the input is 1 half.
            </p><p>Furthermore, you sghould have noticed the connection
            between the prime power function and the prime counting
            function, and how these are able to be approximated using
            other functions.</p><p>See the table to the right to look at
            various values of the Zeta Function that have been
            calculated by users of this program.</p></body></html>"))
        item = self.ZetaTable.horizontalHeaderItem(0)
        item.setText(_translate("InvestigationResultsScreen", "Input
            (s)"))
        item = self.ZetaTable.horizontalHeaderItem(1)
        item.setText(_translate("InvestigationResultsScreen", "Output
            (s)"))
        self.NotesButton.setText(_translate("InvestigationResultsScreen",
            "Notes"))
```

program/user_interface/summary_ui/summary.py

```
"""
summary.py
==========
A GUI for the summary page of the summary section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_SummaryScreen(object):

    def setupUi(self, SummaryScreen):
        SummaryScreen.setObjectName("SummaryScreen")
        SummaryScreen.resize(1340, 723)
        SummaryScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(SummaryScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(565, 20, 211, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.SummaryTab.setObjectName("SummaryTab")
        self.TheoryRecapTab = QtWidgets.QPushButton(self.TabBar)
        self.TheoryRecapTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.TheoryRecapTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TheoryRecapTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.TheoryRecapTab.setObjectName("TheoryRecapTab")
        self.InvestigationResultsLabel = QtWidgets.QLabel(self.TabBar)
        self.InvestigationResultsLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsLabel.setStyleSheet("border: 2px
            solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.InvestigationResultsLabel.setObjectName("InvestigationResultsLabel")
        self.InvestigationResultsTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationResultsTab.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationResultsTab.setStyleSheet("border-radius:
            20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.InvestigationResultsTab.setText("")
        self.InvestigationResultsTab.setObjectName("InvestigationResultsTab")
        self.ConclusionLabel = QtWidgets.QLabel(self.TabBar)
        self.ConclusionLabel.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ConclusionLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ConclusionLabel.setObjectName("ConclusionLabel")
        self.ImpactLabel = QtWidgets.QLabel(self.TabBar)
        self.ImpactLabel.setGeometry(QtCore.QRect(850, 5, 241, 70))
        self.ImpactLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ImpactLabel.setObjectName("ImpactLabel")
        self.ConclusionTab = QtWidgets.QPushButton(self.TabBar)
        self.ConclusionTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ConclusionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ConclusionTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ConclusionTab.setText("")
        self.ConclusionTab.setObjectName("ConclusionTab")
        self.ImpactTab = QtWidgets.QPushButton(self.TabBar)
        self.ImpactTab.setGeometry(QtCore.QRect(850, 5, 241, 70))
        self.ImpactTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ImpactTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ImpactTab.setText("")
        self.ImpactTab.setObjectName("ImpactTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 171, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 80, 1251, 141))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 460, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")
        self.QuestionText = QtWidgets.QLabel(self.MainWidget)
        self.QuestionText.setGeometry(QtCore.QRect(420, 210, 501, 101))
        self.QuestionText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.QuestionText.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.QuestionText.setWordWrap(True)
        self.QuestionText.setObjectName("QuestionText")
        self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
        self.QuestionInput.setGeometry(QtCore.QRect(410, 330, 230, 60))
        self.QuestionInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(700, 330, 121, 61))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(410, 400, 530, 41))
        self.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.MessageLabel.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.MessageLabel.setObjectName("MessageLabel")

        self.retranslateUi(SummaryScreen)
        QtCore.QMetaObject.connectSlotsByName(SummaryScreen)

    def retranslateUi(self, SummaryScreen):
        _translate = QtCore.QCoreApplication.translate
        SummaryScreen.setWindowTitle(_translate("SummaryScreen",
            "Visualising the Riemann Hypothesis - Summary"))
        self.Title.setText(_translate("SummaryScreen", "Summary"))
        self.SummaryTab.setText(_translate("SummaryScreen", "Summary"))
        self.TheoryRecapTab.setText(_translate("SummaryScreen", "Theory
            Recap"))
        self.InvestigationResultsLabel.setText(_translate("SummaryScreen",
            "<html><head/><body><p
            align=\"center\">Investigation<br/>Results</p></body></html>"))
        self.ConclusionLabel.setText(_translate("SummaryScreen",
            "<html><head/><body><p align=\"center\">Conclusion &
            <br/>Evaluation</p></body></html>"))
        self.ImpactLabel.setText(_translate("SummaryScreen",
            "<html><head/><body><p align=\"center\">Impact of the
            <br/>Riemann Hypothesis</p></body></html>"))
        self.PrevButton.setText(_translate("SummaryScreen", "Prev"))
        self.NextButton.setText(_translate("SummaryScreen", "Next"))
        self.SubTitleText.setText(_translate("SummaryScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Summary</span></p></body></html>"))
        self.MainText.setText(_translate("SummaryScreen",
            "<html><head/><body><p>The summary section is the final part
            of this program.</p><p><br/></p><p>Use this section to
            compare your results to the expected results, make any notes
            you need to, learn about the significance of the results
            that you have obtained, and answer the remaining
            questions.</p></body></html>"))
        self.NotesButton.setText(_translate("SummaryScreen", "Notes"))
        self.QuestionText.setText(_translate("SummaryScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:16pt;
            font-weight:600;\">Question</span></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("SummaryScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("SummaryScreen", "Submit"))
        self.MessageLabel.setText(_translate("SummaryScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/summary_ui/theory_recap.py

```
"""
theory_recap.py
===============
A GUI for the theory recap page of the summary section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_TheoryRecapScreen(object):

    def setupUi(self, TheoryRecapScreen):
        TheoryRecapScreen.setObjectName("TheoryRecapScreen")
        TheoryRecapScreen.resize(1340, 723)
        TheoryRecapScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(TheoryRecapScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(565, 20, 211, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SummaryTab.setObjectName("SummaryTab")
        self.TheoryRecapTab = QtWidgets.QPushButton(self.TabBar)
        self.TheoryRecapTab.setGeometry(QtCore.QRect(220, 5, 200, 70))
        self.TheoryRecapTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TheoryRecapTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
```

```
        self.TheoryRecapTab.setObjectName("TheoryRecapTab")
        self.InvestigationResultsLabel = QtWidgets.QLabel(self.TabBar)
        self.InvestigationResultsLabel.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsLabel.setStyleSheet("border: 2px
            solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.InvestigationResultsLabel.setObjectName("InvestigationResultsLabel")
        self.InvestigationResultsTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationResultsTab.setGeometry(QtCore.QRect(430, 5,
            200, 70))
        self.InvestigationResultsTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationResultsTab.setStyleSheet("border-radius:
            20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.InvestigationResultsTab.setText("")
        self.InvestigationResultsTab.setObjectName("InvestigationResultsTab")
        self.ConclusionLabel = QtWidgets.QLabel(self.TabBar)
        self.ConclusionLabel.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ConclusionLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ConclusionLabel.setObjectName("ConclusionLabel")
        self.ImpactLabel = QtWidgets.QLabel(self.TabBar)
        self.ImpactLabel.setGeometry(QtCore.QRect(850, 5, 241, 70))
        self.ImpactLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ImpactLabel.setObjectName("ImpactLabel")
        self.ConclusionTab = QtWidgets.QPushButton(self.TabBar)
        self.ConclusionTab.setGeometry(QtCore.QRect(640, 5, 200, 70))
        self.ConclusionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ConclusionTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ConclusionTab.setText("")
        self.ConclusionTab.setObjectName("ConclusionTab")
        self.ImpactTab = QtWidgets.QPushButton(self.TabBar)
        self.ImpactTab.setGeometry(QtCore.QRect(850, 5, 241, 70))
        self.ImpactTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ImpactTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ImpactTab.setText("")
        self.ImpactTab.setObjectName("ImpactTab")
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 251, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 90, 1251, 341))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.QuestionText = QtWidgets.QLabel(self.MainWidget)
        self.QuestionText.setGeometry(QtCore.QRect(420, 280, 501, 101))
        self.QuestionText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.QuestionText.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.QuestionText.setWordWrap(True)
        self.QuestionText.setObjectName("QuestionText")
        self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
        self.QuestionInput.setGeometry(QtCore.QRect(410, 400, 230, 60))
        self.QuestionInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(700, 400, 121, 61))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(410, 470, 530, 41))
        self.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.MessageLabel.setAlignment(QtCore.Qt.AlignBottom|QtCore.Qt.AlignHCenter)
        self.MessageLabel.setObjectName("MessageLabel")

        self.retranslateUi(TheoryRecapScreen)
        QtCore.QMetaObject.connectSlotsByName(TheoryRecapScreen)

    def retranslateUi(self, TheoryRecapScreen):
        _translate = QtCore.QCoreApplication.translate
        TheoryRecapScreen.setWindowTitle(_translate("TheoryRecapScreen",
            "Visualising the Riemann Hypothesis - Summary"))
        self.Title.setText(_translate("TheoryRecapScreen", "Summary"))
        self.SummaryTab.setText(_translate("TheoryRecapScreen",
            "Summary"))
        self.TheoryRecapTab.setText(_translate("TheoryRecapScreen",
            "Theory Recap"))
        self.InvestigationResultsLabel.setText(_translate("TheoryRecapScreen",
            "<html><head/><body><p
            align=\"center\">Investigation<br/>Results</p></body></html>"))
        self.ConclusionLabel.setText(_translate("TheoryRecapScreen",
            "<html><head/><body><p align=\"center\">Conclusion &
            <br/>Evaluation</p></body></html>"))
        self.ImpactLabel.setText(_translate("TheoryRecapScreen",
            "<html><head/><body><p align=\"center\">Impact of the
            <br/>Riemann Hypothesis</p></body></html>"))
        self.PrevButton.setText(_translate("TheoryRecapScreen", "Prev"))
        self.NextButton.setText(_translate("TheoryRecapScreen", "Next"))
        self.SubTitleText.setText(_translate("TheoryRecapScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Theory Recap</span></p></body></html>"))
        self.MainText.setText(_translate("TheoryRecapScreen",
            "<html><head/><body><p>The Riemann Hypothesis, originating
            from Bernhard Riemann\'s 1859 paper \'On the Number Of
            Primes Less Than a Given Magnitude\', states that \'the real
            part of every nontrivial zero of the Riemann zeta function
            is 0.5\'. </p><p>Hopefully, by using this program, you have
            been able to investigate this conjecture.</p><p>The Riemann
            zeta function, is a more developed version of a function
            first studied by Leonhard Euler back in 1737. This function
            is the sum from n=1 to infinity of 1 dividid by n to the
            power s, where s is a complex number. A complex number is
            any number of the form a+bi, where a and b are real numbers,
            and i is the imaginary unit (equal to the square root of
            -1).</p><p>If proven to be true, the Riemann Zeta Function
            could be used to generate prime numbers and find their
            distribution, which would have profound effects in
            cryptography and even quantum
            physics.</p><p><br/></p></body></html>"))
        self.QuestionText.setText(_translate("TheoryRecapScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:16pt;
            font-weight:600;\">Question</span></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("TheoryRecapScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("TheoryRecapScreen",
            "Submit"))
        self.MessageLabel.setText(_translate("TheoryRecapScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
```

program/user_interface/tutorial_ui/__init__.py

```
"""
__init__.py
===========
Imports for the tutorial_ui
```

```
"""


from .tutorial import Ui_TutorialScreen
from .program_structure_tutorial import Ui_ProgramStructureTutorialScreen
from .introduction_tutorial import Ui_IntroductionTutorialScreen
from .investigation_tutorial import Ui_InvestigationTutorialScreen
from .login_tutorial import Ui_LoginTutorialScreen
from .summary_tutorial import Ui_SummaryTutorialScreen
```

program/user_interface/tutorial_ui/introduction_tutorial.py

```
"""
introduction_tutorial.py
========================
A GUI for the introduction tutorial page of the tutorial section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_IntroductionTutorialScreen(object):

    def setupUi(self, IntroductionTutorialScreen):
        IntroductionTutorialScreen.setObjectName("IntroductionTutorialScreen")
        IntroductionTutorialScreen.resize(1340, 723)
        IntroductionTutorialScreen.setToolTipDuration(0)
        IntroductionTutorialScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(IntroductionTutorialScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(590, 20, 161, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.TutorialTab.setObjectName("TutorialTab")
       self.LoginTab = QtWidgets.QPushButton(self.TabBar)
       self.LoginTab.setGeometry(QtCore.QRect(430, 5, 221, 70))
       self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.LoginTab.setObjectName("LoginTab")
       self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
       self.IntroductionTab.setGeometry(QtCore.QRect(660, 5, 200, 70))
       self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
       self.IntroductionTab.setObjectName("IntroductionTab")
       self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
       self.InvestigationTab.setGeometry(QtCore.QRect(870, 5, 200, 70))
       self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.InvestigationTab.setObjectName("InvestigationTab")
       self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
       self.SummaryTab.setGeometry(QtCore.QRect(1080, 5, 200, 70))
       self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.SummaryTab.setObjectName("SummaryTab")
       self.ProgramStructureTab = QtWidgets.QPushButton(self.TabBar)
       self.ProgramStructureTab.setGeometry(QtCore.QRect(220, 5, 200,
           70))
       self.ProgramStructureTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.ProgramStructureTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
```

```
"font: 18pt \"Sans Serif\";\n"
"")
        self.ProgramStructureTab.setText("")
        self.ProgramStructureTab.setObjectName("ProgramStructureTab")
        self.ProgramStructureLabel = QtWidgets.QLabel(self.TabBar)
        self.ProgramStructureLabel.setGeometry(QtCore.QRect(220, 5, 200,
            70))
        self.ProgramStructureLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ProgramStructureLabel.setObjectName("ProgramStructureLabel")
        self.TutorialTab.raise_()
        self.LoginTab.raise_()
        self.IntroductionTab.raise_()
        self.InvestigationTab.raise_()
        self.SummaryTab.raise_()
        self.ProgramStructureLabel.raise_()
        self.ProgramStructureTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 311, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 60, 1251, 391))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
        self.QuestionInput.setGeometry(QtCore.QRect(555, 410, 230, 60))
        self.QuestionInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.QuestionText = QtWidgets.QLabel(self.MainWidget)
        self.QuestionText.setGeometry(QtCore.QRect(470, 330, 391, 71))
        self.QuestionText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.QuestionText.setWordWrap(True)
        self.QuestionText.setObjectName("QuestionText")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(405, 480, 530, 41))
        self.MessageLabel.setStyleSheet("color: rgb(0, 140, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.MessageLabel.setObjectName("MessageLabel")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(820, 415, 121, 51))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.SubTitleText.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.MainText.raise_()
        self.QuestionText.raise_()
        self.QuestionInput.raise_()
        self.MessageLabel.raise_()
        self.SubmitButton.raise_()

        self.retranslateUi(IntroductionTutorialScreen)
        QtCore.QMetaObject.connectSlotsByName(IntroductionTutorialScreen)

    def retranslateUi(self, IntroductionTutorialScreen):
        _translate = QtCore.QCoreApplication.translate
        IntroductionTutorialScreen.setWindowTitle(_translate("IntroductionTutorialScreen",
            "Visualising the Riemann Hypothesis - Tutorial"))
        self.Title.setText(_translate("IntroductionTutorialScreen",
            "Tutorial"))
        self.TutorialTab.setText(_translate("IntroductionTutorialScreen",
            "Tutorial"))
        self.LoginTab.setText(_translate("IntroductionTutorialScreen",
            "Login"))
        self.IntroductionTab.setText(_translate("IntroductionTutorialScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("IntroductionTutorialScreen",
            "Investigation"))
        self.SummaryTab.setText(_translate("IntroductionTutorialScreen",
            "Summary"))
        self.ProgramStructureLabel.setText(_translate("IntroductionTutorialScreen",
            "<html><head/><body><p
            align=\"center\">Program<br/>Structure</p></body></html>"))
        self.PrevButton.setText(_translate("IntroductionTutorialScreen",
            "Prev"))
        self.NextButton.setText(_translate("IntroductionTutorialScreen",
            "Next"))
        self.SubTitleText.setText(_translate("IntroductionTutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Introduction</span></p></body></html>"))
        self.MainText.setText(_translate("IntroductionTutorialScreen",
            "<html><head/><body><p>The introduction section of this
            program is designed to give you a sufficient amount of
            knowledge about the Riemann Hypothesis, such that you will
            be able to understand the complicated mathematics behind
            this program so that you will be able to fully utilise the
            functionality of this program.</p><p>This section will give
            you some Historical Background on the Riemann Hypothesis, it
            will explain what the Riemann Hypothesis actually is, and
            detail some Practical Applications of the Riemann
            Hypothesis.</p><p><br/>The introduction section will give
            you the basic knowledge you need to be able to understand
            this program, so it is strongly recommended to read this
            before started to use the program.</p><p>Throughout the
            Introduction Section, and the rest of the program, will be
            varioues questions. Answer these questions correctly to be
            able to learn mroe about the Riemann Hypothesis! When you
            answer the question correctly, it will say that you have it
            correct. Otherwise, keep on trying to get the right
            answer.</p><p>Here is an example
            below:</p><p><br/></p><p><br/></p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("IntroductionTutorialScreen",
            "Answer"))
        self.QuestionText.setText(_translate("IntroductionTutorialScreen",
            "<html><head/><body><p align=\"center\"><span style=\"
            font-size:16pt;
            font-weight:600;\">Question</span></p></body></html>"))
        self.MessageLabel.setText(_translate("IntroductionTutorialScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        self.SubmitButton.setText(_translate("IntroductionTutorialScreen",
            "Submit"))
```

program/user_interface/tutorial_ui/investigation_tutorial.py

```
"""
investigation_tutorial.py
=========================
A GUI for the investigation tutorial page of the tutorial section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_InvestigationTutorialScreen(object):

    def setupUi(self, InvestigationTutorialScreen):
        InvestigationTutorialScreen.setObjectName("InvestigationTutorialScreen")
        InvestigationTutorialScreen.resize(1340, 723)
        InvestigationTutorialScreen.setToolTipDuration(0)
        InvestigationTutorialScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(InvestigationTutorialScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(590, 20, 161, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.TutorialTab.setObjectName("TutorialTab")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(430, 5, 221, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.LoginTab.setObjectName("LoginTab")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(660, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationTab.setGeometry(QtCore.QRect(870, 5, 200, 70))
        self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.InvestigationTab.setObjectName("InvestigationTab")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(1080, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
```

```
        self.SummaryTab.setObjectName("SummaryTab")
        self.ProgramStructureTab = QtWidgets.QPushButton(self.TabBar)
        self.ProgramStructureTab.setGeometry(QtCore.QRect(220, 5, 200,
            70))
        self.ProgramStructureTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ProgramStructureTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ProgramStructureTab.setText("")
        self.ProgramStructureTab.setObjectName("ProgramStructureTab")
        self.ProgramStructureLabel = QtWidgets.QLabel(self.TabBar)
        self.ProgramStructureLabel.setGeometry(QtCore.QRect(220, 5, 200,
            70))
        self.ProgramStructureLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ProgramStructureLabel.setObjectName("ProgramStructureLabel")
        self.TutorialTab.raise_()
        self.LoginTab.raise_()
        self.IntroductionTab.raise_()
        self.InvestigationTab.raise_()
        self.SummaryTab.raise_()
        self.ProgramStructureLabel.raise_()
        self.ProgramStructureTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
```

```
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 311, 51))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MSlider = QtWidgets.QSlider(self.MainWidget)
        self.MSlider.setGeometry(QtCore.QRect(400, 250, 181, 31))
        self.MSlider.setMinimum(-10)
        self.MSlider.setMaximum(10)
        self.MSlider.setOrientation(QtCore.Qt.Horizontal)
        self.MSlider.setInvertedAppearance(False)
        self.MSlider.setInvertedControls(False)
        self.MSlider.setTickPosition(QtWidgets.QSlider.NoTicks)
        self.MSlider.setObjectName("MSlider")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 70, 1251, 111))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.GraphText = QtWidgets.QLabel(self.MainWidget)
        self.GraphText.setGeometry(QtCore.QRect(600, 170, 151, 61))
        self.GraphText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.GraphText.setObjectName("GraphText")
        self.CSlider = QtWidgets.QSlider(self.MainWidget)
        self.CSlider.setGeometry(QtCore.QRect(760, 250, 181, 31))
        self.CSlider.setMinimum(-10)
        self.CSlider.setMaximum(10)
        self.CSlider.setOrientation(QtCore.Qt.Horizontal)
        self.CSlider.setInvertedAppearance(False)
        self.CSlider.setInvertedControls(False)
        self.CSlider.setTickPosition(QtWidgets.QSlider.NoTicks)
        self.CSlider.setObjectName("CSlider")
        self.MText = QtWidgets.QLabel(self.MainWidget)
        self.MText.setGeometry(QtCore.QRect(330, 220, 61, 61))
        self.MText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MText.setObjectName("MText")
        self.CText = QtWidgets.QLabel(self.MainWidget)
        self.CText.setGeometry(QtCore.QRect(690, 220, 61, 61))
        self.CText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.CText.setObjectName("CText")
        self.GraphButton = QtWidgets.QPushButton(self.MainWidget)
        self.GraphButton.setGeometry(QtCore.QRect(610, 280, 131, 51))
        self.GraphButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.GraphButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.GraphButton.setObjectName("GraphButton")
        self.MainText_2 = QtWidgets.QLabel(self.MainWidget)
        self.MainText_2.setGeometry(QtCore.QRect(20, 350, 861, 31))
        self.MainText_2.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText_2.setWordWrap(True)
        self.MainText_2.setObjectName("MainText_2")
        self.QuestionInput = QtWidgets.QLineEdit(self.MainWidget)
        self.QuestionInput.setGeometry(QtCore.QRect(540, 420, 111, 60))
        self.QuestionInput.setStyleSheet("background-color: rgb(239,
            239, 239);\n"
"color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\";\n"
"border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);")
        self.QuestionInput.setText("")
        self.QuestionInput.setEchoMode(QtWidgets.QLineEdit.Normal)
        self.QuestionInput.setCursorPosition(0)
        self.QuestionInput.setAlignment(QtCore.Qt.AlignCenter)
        self.QuestionInput.setObjectName("QuestionInput")
        self.SubmitButton = QtWidgets.QPushButton(self.MainWidget)
        self.SubmitButton.setGeometry(QtCore.QRect(690, 420, 131, 60))
        self.SubmitButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SubmitButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SubmitButton.setObjectName("SubmitButton")
        self.MessageLabel = QtWidgets.QLabel(self.MainWidget)
        self.MessageLabel.setGeometry(QtCore.QRect(440, 480, 461, 41))
        self.MessageLabel.setStyleSheet("color: rgb(255, 0, 0);\n"
"font: 18pt \"Sans Serif\";")
        self.MessageLabel.setObjectName("MessageLabel")
        self.MDisplay = QtWidgets.QLabel(self.MainWidget)
        self.MDisplay.setGeometry(QtCore.QRect(430, 200, 120, 51))
        self.MDisplay.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.MDisplay.setObjectName("MDisplay")
        self.CDisplay = QtWidgets.QLabel(self.MainWidget)
        self.CDisplay.setGeometry(QtCore.QRect(790, 200, 120, 51))
        self.CDisplay.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.CDisplay.setObjectName("CDisplay")
        self.QuestionText = QtWidgets.QLabel(self.MainWidget)
        self.QuestionText.setGeometry(QtCore.QRect(590, 380, 161, 31))
        self.QuestionText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.QuestionText.setWordWrap(True)
        self.QuestionText.setObjectName("QuestionText")
        self.SubTitleText.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.MSlider.raise_()
        self.MainText.raise_()
        self.GraphText.raise_()
        self.CSlider.raise_()
        self.MText.raise_()
        self.CText.raise_()
        self.GraphButton.raise_()
        self.MainText_2.raise_()
        self.QuestionInput.raise_()
        self.SubmitButton.raise_()
        self.MessageLabel.raise_()
        self.MDisplay.raise_()
        self.CDisplay.raise_()
        self.QuestionText.raise_()

        self.retranslateUi(InvestigationTutorialScreen)
        QtCore.QMetaObject.connectSlotsByName(InvestigationTutorialScreen)

    def retranslateUi(self, InvestigationTutorialScreen):
        _translate = QtCore.QCoreApplication.translate
        InvestigationTutorialScreen.setWindowTitle(_translate("InvestigationTutorialScreen",
            "Visualising the Riemann Hypothesis - Tutorial"))
        self.Title.setText(_translate("InvestigationTutorialScreen",
            "Tutorial"))
        self.TutorialTab.setText(_translate("InvestigationTutorialScreen",
            "Tutorial"))
        self.LoginTab.setText(_translate("InvestigationTutorialScreen",
            "Login"))
        self.IntroductionTab.setText(_translate("InvestigationTutorialScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("InvestigationTutorialScreen",
            "Investigation"))
        self.SummaryTab.setText(_translate("InvestigationTutorialScreen",
            "Summary"))
        self.ProgramStructureLabel.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p
            align=\"center\">Program<br/>Structure</p></body></html>"))
        self.PrevButton.setText(_translate("InvestigationTutorialScreen",
            "Prev"))
        self.NextButton.setText(_translate("InvestigationTutorialScreen",
            "Next"))
        self.SubTitleText.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Investigation</span></p></body></html>"))
        self.MainText.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p>The Investigation Section is the main
            part of this program. It will allow you to conduct your own
            investigation into the Riemann Hypothesis and let you record
            your results, while asking you questions along the
            way</p><p>Lets get some practice with how this program will
            work. Slide the slides to adjust the equation, then press
            graph to display the graph. Change these values to see what
            happens.</p></body></html>"))
        self.GraphText.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">y=mx+c</span></p></body></html>"))
        self.MText.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">M:</span></p></body></html>"))
        self.CText.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">C:</span></p></body></html>"))
        self.GraphButton.setText(_translate("InvestigationTutorialScreen",
            "Graph"))
        self.MainText_2.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p>There will also be many opportunities
            to answer questions during this section. Have a go at the
            one below!</p></body></html>"))
        self.QuestionInput.setPlaceholderText(_translate("InvestigationTutorialScreen",
            "Answer"))
        self.SubmitButton.setText(_translate("InvestigationTutorialScreen",
            "Submit"))
        self.MessageLabel.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p
            align=\"center\"><br/></p></body></html>"))
        self.MDisplay.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p
            align=\"center\">0</p></body></html>"))
        self.CDisplay.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p
            align=\"center\">0</p></body></html>"))
        self.QuestionText.setText(_translate("InvestigationTutorialScreen",
            "<html><head/><body><p><span style=\" font-size:16pt;
            font-weight:600;\">Question</span></p></body></html>"))
```

program/user_interface/tutorial_ui/login$_{t}utorial.py$

```
"""
login_tutorial.py
=================
A GUI for the login tutorial page of the tutorial section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_LoginTutorialScreen(object):

    def setupUi(self, LoginTutorialScreen):
        LoginTutorialScreen.setObjectName("LoginTutorialScreen")
        LoginTutorialScreen.resize(1340, 723)
        LoginTutorialScreen.setToolTipDuration(0)
        LoginTutorialScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(LoginTutorialScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(590, 20, 161, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(69, 69, 69);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
 "")
        self.TutorialTab.setObjectName("TutorialTab")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(430, 5, 221, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(239, 239, 239);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
 "")
        self.LoginTab.setObjectName("LoginTab")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(660, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(69, 69, 69);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
 "")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationTab.setGeometry(QtCore.QRect(870, 5, 200, 70))
        self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(69, 69, 69);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
 "")
        self.InvestigationTab.setObjectName("InvestigationTab")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(1080, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
 "border-radius: 20px;\n"
 "border-color:rgb(69, 69, 69);\n"
 "background-color: rgb(69, 69, 69);\n"
 "font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
 "")
        self.SummaryTab.setObjectName("SummaryTab")
        self.ProgramStructureTab = QtWidgets.QPushButton(self.TabBar)
        self.ProgramStructureTab.setGeometry(QtCore.QRect(220, 5, 200,
            70))
        self.ProgramStructureTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ProgramStructureTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ProgramStructureTab.setText("")
        self.ProgramStructureTab.setObjectName("ProgramStructureTab")
        self.ProgramStructureLabel = QtWidgets.QLabel(self.TabBar)
        self.ProgramStructureLabel.setGeometry(QtCore.QRect(220, 5, 200,
            70))
        self.ProgramStructureLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ProgramStructureLabel.setObjectName("ProgramStructureLabel")
        self.TutorialTab.raise_()
        self.LoginTab.raise_()
        self.IntroductionTab.raise_()
        self.InvestigationTab.raise_()
        self.SummaryTab.raise_()
        self.ProgramStructureLabel.raise_()
        self.ProgramStructureTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 311, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 60, 1251, 391))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.SubTitleText.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.MainText.raise_()

        self.retranslateUi(LoginTutorialScreen)
        QtCore.QMetaObject.connectSlotsByName(LoginTutorialScreen)

    def retranslateUi(self, LoginTutorialScreen):
        _translate = QtCore.QCoreApplication.translate
        LoginTutorialScreen.setWindowTitle(_translate("LoginTutorialScreen",
            "Visualising the Riemann Hypothesis - Tutorial"))
        self.Title.setText(_translate("LoginTutorialScreen", "Tutorial"))
        self.TutorialTab.setText(_translate("LoginTutorialScreen",
            "Tutorial"))
        self.LoginTab.setText(_translate("LoginTutorialScreen", "Login"))
        self.IntroductionTab.setText(_translate("LoginTutorialScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("LoginTutorialScreen",
            "Investigation"))
        self.SummaryTab.setText(_translate("LoginTutorialScreen",
            "Summary"))
        self.ProgramStructureLabel.setText(_translate("LoginTutorialScreen",
            "<html><head/><body><p
            align=\"center\">Program<br/>Structure</p></body></html>"))
        self.PrevButton.setText(_translate("LoginTutorialScreen",
            "Prev"))
        self.NextButton.setText(_translate("LoginTutorialScreen",
            "Next"))
        self.SubTitleText.setText(_translate("LoginTutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Login</span></p></body></html>"))
        self.MainText.setText(_translate("LoginTutorialScreen",
            "<html><head/><body><p>The Login section of this program
            allows you to sign in to an account.</p><p>The different
            options you have are: </p><p>Login In</p><p>Sign
            Up</p><p>Forgotten Password</p><p>Reset
            Password</p><p><br/></p><p>Although you can use this program
            without an account, once you create and sign in to an
            account you will be able to use this program to it\'s full
            extent. When signed into an account, you will be able to
            answer questions on the Riemann Hypothesis, make your own
            notes, and participate to the
            leaderboard</p></body></html>"))
```

program/user_interface/tutorial_ui/program_structure_tutorial.py

```
"""
program_structure_tutorial.py
=============================
A GUI for the program structure tutorial page of the tutorial section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_ProgramStructureTutorialScreen(object):

    def setupUi(self, ProgramStructureTutorialScreen):
        ProgramStructureTutorialScreen.setObjectName("ProgramStructureTutorialScreen")
        ProgramStructureTutorialScreen.resize(1340, 723)
        ProgramStructureTutorialScreen.setToolTipDuration(0)
        ProgramStructureTutorialScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(ProgramStructureTutorialScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(590, 20, 161, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
```

```
        self.TutorialTab.setObjectName("TutorialTab")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(430, 5, 221, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.LoginTab.setObjectName("LoginTab")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(660, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationTab.setGeometry(QtCore.QRect(870, 5, 200, 70))
        self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.InvestigationTab.setObjectName("InvestigationTab")
        self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
        self.SummaryTab.setGeometry(QtCore.QRect(1080, 5, 200, 70))
        self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.SummaryTab.setObjectName("SummaryTab")
        self.ProgramStructureTab = QtWidgets.QPushButton(self.TabBar)
        self.ProgramStructureTab.setGeometry(QtCore.QRect(220, 5, 200,
            70))
        self.ProgramStructureTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.ProgramStructureTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ProgramStructureTab.setText("")
```

```
       self.ProgramStructureTab.setObjectName("ProgramStructureTab")
       self.ProgramStructureLabel = QtWidgets.QLabel(self.TabBar)
       self.ProgramStructureLabel.setGeometry(QtCore.QRect(220, 5, 200,
           70))
       self.ProgramStructureLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
       self.ProgramStructureLabel.setObjectName("ProgramStructureLabel")
       self.TutorialTab.raise_()
       self.LoginTab.raise_()
       self.IntroductionTab.raise_()
       self.InvestigationTab.raise_()
       self.SummaryTab.raise_()
       self.ProgramStructureLabel.raise_()
       self.ProgramStructureTab.raise_()
       self.MainWidget = QtWidgets.QWidget(self.widget)
       self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
       self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
"border-radius: 20px;")
       self.MainWidget.setObjectName("MainWidget")
       self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
       self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
       self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.PrevButton.setObjectName("PrevButton")
       self.NextButton = QtWidgets.QPushButton(self.MainWidget)
       self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
       self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.NextButton.setObjectName("NextButton")
       self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
       self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 341, 41))
       self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
           color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
       self.SubTitleText.setObjectName("SubTitleText")
```

```
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 80, 1251, 61))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.TemplateImage = QtWidgets.QLabel(self.MainWidget)
        self.TemplateImage.setGeometry(QtCore.QRect(200, 150, 941, 361))
        sizePolicy =
            QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Preferred,
            QtWidgets.QSizePolicy.Preferred)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(self.TemplateImage.sizePolicy().hasHeightForWidth())
        self.TemplateImage.setSizePolicy(sizePolicy)
        self.TemplateImage.setText("")
        self.TemplateImage.setPixmap(QtGui.QPixmap("ui/tutorial_screens/../../media/annotated-template-
        self.TemplateImage.setObjectName("TemplateImage")
        self.SubTitleText.raise_()
        self.MainText.raise_()
        self.TemplateImage.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()

        self.retranslateUi(ProgramStructureTutorialScreen)
        QtCore.QMetaObject.connectSlotsByName(ProgramStructureTutorialScreen)

    def retranslateUi(self, ProgramStructureTutorialScreen):
        _translate = QtCore.QCoreApplication.translate
        ProgramStructureTutorialScreen.setWindowTitle(_translate("ProgramStructureTutorialScreen",
            "Visualising the Riemann Hypothesis - Tutorial"))
        self.Title.setText(_translate("ProgramStructureTutorialScreen",
            "Tutorial"))
        self.TutorialTab.setText(_translate("ProgramStructureTutorialScreen",
            "Tutorial"))
        self.LoginTab.setText(_translate("ProgramStructureTutorialScreen",
            "Login"))
        self.IntroductionTab.setText(_translate("ProgramStructureTutorialScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("ProgramStructureTutorialScreen",
            "Investigation"))
        self.SummaryTab.setText(_translate("ProgramStructureTutorialScreen",
            "Summary"))
        self.ProgramStructureLabel.setText(_translate("ProgramStructureTutorialScreen",
            "<html><head/><body><p
            align=\"center\">Program<br/>Structure</p></body></html>"))
        self.PrevButton.setText(_translate("ProgramStructureTutorialScreen",
            "Prev"))
        self.NextButton.setText(_translate("ProgramStructureTutorialScreen",
            "Next"))
        self.SubTitleText.setText(_translate("ProgramStructureTutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Program
            Structure</span></p></body></html>"))
        self.MainText.setText(_translate("ProgramStructureTutorialScreen",
            "<html><head/><body><p>This section of the tutorial aims to
            inform you how to navigate between each page in the
            program</p><p>Below is an image of an example screen design,
            and is labelled with how to navigate it</p></body></html>"))
```

program/user_interface/tutorial_ui/summary_tutorial.py

```
"""
summary_tutorial.py
===================
A GUI for the summary tutorial page of the tutorial section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_SummaryTutorialScreen(object):

    def setupUi(self, SummaryTutorialScreen):
        SummaryTutorialScreen.setObjectName("SummaryTutorialScreen")
        SummaryTutorialScreen.resize(1340, 723)
        SummaryTutorialScreen.setToolTipDuration(0)
        SummaryTutorialScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(SummaryTutorialScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(590, 20, 161, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
```

```
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.TutorialTab.setObjectName("TutorialTab")
       self.LoginTab = QtWidgets.QPushButton(self.TabBar)
       self.LoginTab.setGeometry(QtCore.QRect(430, 5, 221, 70))
       self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.LoginTab.setObjectName("LoginTab")
       self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
       self.IntroductionTab.setGeometry(QtCore.QRect(660, 5, 200, 70))
       self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.IntroductionTab.setObjectName("IntroductionTab")
       self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
       self.InvestigationTab.setGeometry(QtCore.QRect(870, 5, 200, 70))
       self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.InvestigationTab.setObjectName("InvestigationTab")
       self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
       self.SummaryTab.setGeometry(QtCore.QRect(1080, 5, 200, 70))
       self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
       self.SummaryTab.setObjectName("SummaryTab")
       self.ProgramStructureTab = QtWidgets.QPushButton(self.TabBar)
       self.ProgramStructureTab.setGeometry(QtCore.QRect(220, 5, 200,
           70))
       self.ProgramStructureTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.ProgramStructureTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
        self.ProgramStructureTab.setText("")
        self.ProgramStructureTab.setObjectName("ProgramStructureTab")
        self.ProgramStructureLabel = QtWidgets.QLabel(self.TabBar)
        self.ProgramStructureLabel.setGeometry(QtCore.QRect(220, 5, 200,
            70))
        self.ProgramStructureLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.ProgramStructureLabel.setObjectName("ProgramStructureLabel")
        self.TutorialTab.raise_()
        self.LoginTab.raise_()
        self.IntroductionTab.raise_()
        self.InvestigationTab.raise_()
        self.SummaryTab.raise_()
        self.ProgramStructureLabel.raise_()
        self.ProgramStructureTab.raise_()
        self.MainWidget = QtWidgets.QWidget(self.widget)
        self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
        self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
            239);\n"
"border-radius: 20px;")
        self.MainWidget.setObjectName("MainWidget")
        self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
        self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 311, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 60, 1251, 61))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")
        self.MainText_3 = QtWidgets.QLabel(self.MainWidget)
        self.MainText_2.setGeometry(QtCore.QRect(40, 130, 1251, 101))
        self.MainText_2.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText_2.setWordWrap(True)
        self.MainText_2.setObjectName("MainText_2")
        self.NotesButton = QtWidgets.QPushButton(self.MainWidget)
        self.NotesButton.setGeometry(QtCore.QRect(570, 240, 200, 70))
        self.NotesButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NotesButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NotesButton.setObjectName("NotesButton")
        self.SubTitleText.raise_()
        self.PrevButton.raise_()
        self.NextButton.raise_()
        self.MainText.raise_()
        self.MainText_2.raise_()
        self.NotesButton.raise_()

        self.retranslateUi(SummaryTutorialScreen)
        QtCore.QMetaObject.connectSlotsByName(SummaryTutorialScreen)

    def retranslateUi(self, SummaryTutorialScreen):
        _translate = QtCore.QCoreApplication.translate
        SummaryTutorialScreen.setWindowTitle(_translate("SummaryTutorialScreen",
            "Visualising the Riemann Hypothesis - Tutorial"))
        self.Title.setText(_translate("SummaryTutorialScreen",
            "Tutorial"))
        self.TutorialTab.setText(_translate("SummaryTutorialScreen",
            "Tutorial"))
        self.LoginTab.setText(_translate("SummaryTutorialScreen",
            "Login"))
        self.IntroductionTab.setText(_translate("SummaryTutorialScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("SummaryTutorialScreen",
            "Investigation"))
        self.SummaryTab.setText(_translate("SummaryTutorialScreen",
            "Summary"))
        self.ProgramStructureLabel.setText(_translate("SummaryTutorialScreen",
            "<html><head/><body><p
            align=\"center\">Program<br/>Structure</p></body></html>"))
        self.PrevButton.setText(_translate("SummaryTutorialScreen",
            "Prev"))
        self.NextButton.setText(_translate("SummaryTutorialScreen",
            "Next"))
        self.SubTitleText.setText(_translate("SummaryTutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Summary</span></p></body></html>"))
        self.MainText.setText(_translate("SummaryTutorialScreen",
            "<html><head/><body><p>The Summary Section will be the final
            section of this program. It will be a chance for you, as the
            user, to be able to reflect on your own results that you
            have found from this program, and comparing these results to
            what would have been expected for the Riemann Hypothesis to
            be true.</p></body></html>"))
        self.MainText_2.setText(_translate("SummaryTutorialScreen",
            "<html><head/><body><p>Throughout this section, and previous
            other sections in the program, there will be various buttons
            saying \'Notes\'. Clicking on these will allow you to write
            your own notes on what you have just learnt or found out
            from the program, for you to refer back to
            later.</p><p>Click on the notes button below to try it
            out!</p></body></html>"))
        self.NotesButton.setText(_translate("SummaryTutorialScreen",
            "Notes"))
```

program/user_interface/tutorial_ui/tutorial.py

```
"""
tutorial.py
===========
A GUI for the tutorial page of the tutorial section
"""


from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_TutorialScreen(object):

    def setupUi(self, TutorialScreen):
        TutorialScreen.setObjectName("TutorialScreen")
        TutorialScreen.resize(1340, 723)
        TutorialScreen.setSizeGripEnabled(False)
        self.widget = QtWidgets.QWidget(TutorialScreen)
        self.widget.setGeometry(QtCore.QRect(0, 0, 1340, 720))
        self.widget.setStyleSheet("background-color: rgb(69, 69, 69);")
        self.widget.setObjectName("widget")
        self.Title = QtWidgets.QLabel(self.widget)
        self.Title.setGeometry(QtCore.QRect(590, 20, 161, 51))
        self.Title.setStyleSheet("font: 36pt \"Sans Serif\";
            color:rgb(239, 239, 239)")
        self.Title.setObjectName("Title")
        self.TabBar = QtWidgets.QWidget(self.widget)
        self.TabBar.setGeometry(QtCore.QRect(0, 80, 1340, 80))
        self.TabBar.setStyleSheet("background-color: rgb(239, 239,
            239);")
        self.TabBar.setObjectName("TabBar")
        self.TutorialTab = QtWidgets.QPushButton(self.TabBar)
        self.TutorialTab.setGeometry(QtCore.QRect(10, 5, 200, 70))
        self.TutorialTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.TutorialTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239);\n"
"font: 18pt \"Sans Serif\"; color:rgb(69, 69, 69);\n"
"")
        self.TutorialTab.setObjectName("TutorialTab")
        self.LoginTab = QtWidgets.QPushButton(self.TabBar)
        self.LoginTab.setGeometry(QtCore.QRect(430, 5, 221, 70))
        self.LoginTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.LoginTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.LoginTab.setObjectName("LoginTab")
        self.IntroductionTab = QtWidgets.QPushButton(self.TabBar)
        self.IntroductionTab.setGeometry(QtCore.QRect(660, 5, 200, 70))
        self.IntroductionTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.IntroductionTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.IntroductionTab.setObjectName("IntroductionTab")
        self.InvestigationTab = QtWidgets.QPushButton(self.TabBar)
        self.InvestigationTab.setGeometry(QtCore.QRect(870, 5, 200, 70))
        self.InvestigationTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.InvestigationTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.InvestigationTab.setObjectName("InvestigationTab")
       self.SummaryTab = QtWidgets.QPushButton(self.TabBar)
       self.SummaryTab.setGeometry(QtCore.QRect(1080, 5, 200, 70))
       self.SummaryTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.SummaryTab.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.SummaryTab.setObjectName("SummaryTab")
       self.ProgramStructureTab = QtWidgets.QPushButton(self.TabBar)
       self.ProgramStructureTab.setGeometry(QtCore.QRect(220, 5, 200,
           70))
       self.ProgramStructureTab.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
       self.ProgramStructureTab.setStyleSheet("border-radius: 20px;\n"
"background-color: rgba(0, 0, 0, 0);\n"
"font: 18pt \"Sans Serif\";\n"
"")
       self.ProgramStructureTab.setText("")
       self.ProgramStructureTab.setObjectName("ProgramStructureTab")
       self.ProgramStructureLabel = QtWidgets.QLabel(self.TabBar)
       self.ProgramStructureLabel.setGeometry(QtCore.QRect(220, 5, 200,
           70))
       self.ProgramStructureLabel.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
       self.ProgramStructureLabel.setObjectName("ProgramStructureLabel")
       self.TutorialTab.raise_()
       self.LoginTab.raise_()
       self.IntroductionTab.raise_()
       self.InvestigationTab.raise_()
       self.SummaryTab.raise_()
       self.ProgramStructureLabel.raise_()
       self.ProgramStructureTab.raise_()
       self.MainWidget = QtWidgets.QWidget(self.widget)
       self.MainWidget.setGeometry(QtCore.QRect(10, 170, 1320, 540))
       self.MainWidget.setStyleSheet("background-color: rgb(239, 239,
           239);\n"
"border-radius: 20px;")
       self.MainWidget.setObjectName("MainWidget")
       self.PrevButton = QtWidgets.QPushButton(self.MainWidget)
       self.PrevButton.setGeometry(QtCore.QRect(10, 460, 200, 70))
```

```
        self.PrevButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.PrevButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.PrevButton.setObjectName("PrevButton")
        self.NextButton = QtWidgets.QPushButton(self.MainWidget)
        self.NextButton.setGeometry(QtCore.QRect(1110, 460, 200, 70))
        self.NextButton.setCursor(QtGui.QCursor(QtCore.Qt.PointingHandCursor))
        self.NextButton.setStyleSheet("border: 2px solid;\n"
"border-radius: 20px;\n"
"border-color:rgb(69, 69, 69);\n"
"background-color: rgb(69, 69, 69);\n"
"font: 18pt \"Sans Serif\"; color:rgb(239, 239, 239);\n"
"")
        self.NextButton.setObjectName("NextButton")
        self.SubTitleText = QtWidgets.QLabel(self.MainWidget)
        self.SubTitleText.setGeometry(QtCore.QRect(40, 20, 681, 41))
        self.SubTitleText.setStyleSheet("font: 25pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.SubTitleText.setObjectName("SubTitleText")
        self.MainText = QtWidgets.QLabel(self.MainWidget)
        self.MainText.setGeometry(QtCore.QRect(40, 80, 1251, 201))
        self.MainText.setStyleSheet("font: 13pt \"Sans Serif\";
            color:rgb(69, 69, 69);\n"
"background-color: rgb(239, 239, 239); padding: 5px;")
        self.MainText.setAlignment(QtCore.Qt.AlignLeading|QtCore.Qt.AlignLeft|QtCore.Qt.AlignTop)
        self.MainText.setWordWrap(True)
        self.MainText.setObjectName("MainText")

        self.retranslateUi(TutorialScreen)
        QtCore.QMetaObject.connectSlotsByName(TutorialScreen)

    def retranslateUi(self, TutorialScreen):
        _translate = QtCore.QCoreApplication.translate
        TutorialScreen.setWindowTitle(_translate("TutorialScreen",
            "Visualising the Riemann Hypothesis - Tutorial"))
        self.Title.setText(_translate("TutorialScreen", "Tutorial"))
        self.TutorialTab.setText(_translate("TutorialScreen",
            "Tutorial"))
        self.LoginTab.setText(_translate("TutorialScreen", "Login"))
        self.IntroductionTab.setText(_translate("TutorialScreen",
            "Introduction"))
        self.InvestigationTab.setText(_translate("TutorialScreen",
            "Investigation"))
        self.SummaryTab.setText(_translate("TutorialScreen", "Summary"))
        self.ProgramStructureLabel.setText(_translate("TutorialScreen",
            "<html><head/><body><p
            align=\"center\">Program<br/>Structure</p></body></html>"))
        self.PrevButton.setText(_translate("TutorialScreen", "Prev"))
        self.NextButton.setText(_translate("TutorialScreen", "Next"))
        self.SubTitleText.setText(_translate("TutorialScreen",
            "<html><head/><body><p><span style=\"
            font-weight:600;\">Tutorial</span></p></body></html>"))
        self.MainText.setText(_translate("TutorialScreen",
            "<html><head/><body><p>Welcome to the Tutorial Section of
            this program.</p><p>The aim of this is to teach you how to
            be able to use this program.</p><p>In terms of controls for
            the program, avoid pressing the enter key to submit any text
            input - this can be done but make sure you\'re selecting the
            correct button. To avoid any confusion it may be simplest to
            just click on the desired button. Furthermore, to quit any
            pages that do not have a back button, simply press
            escape.</p><p>Click on any of the five other tabs above, or
            on the next button, to find out how to use this program to
            it\'s full extent</p></body></html>"))
```