



A ■ P ■ U

**ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION**

Further Web Design & Development

(CT117-3-2)

Individual Assignment

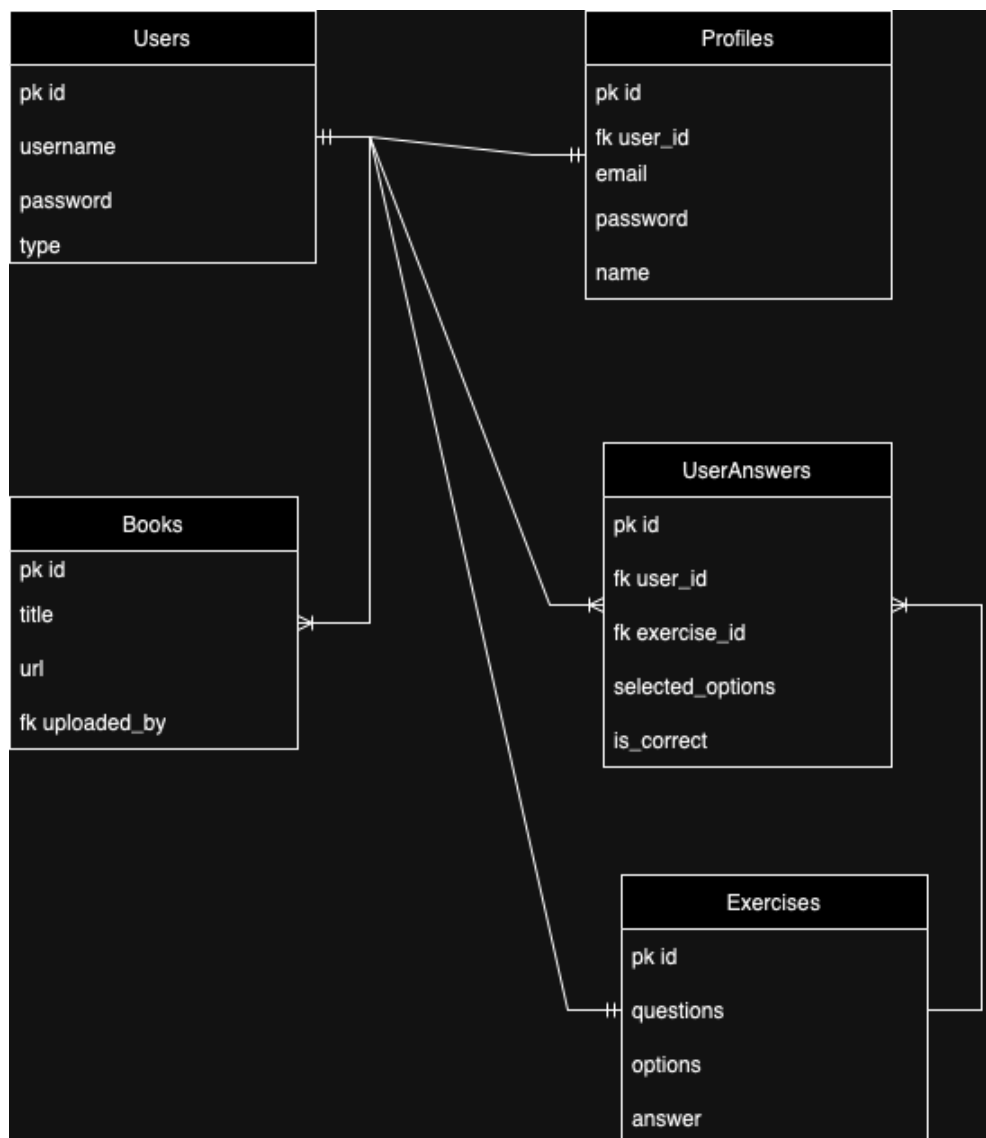
Name:	Omar Mahmoud Elkady
ID:	TP068703
Weightage:	100%

Pyledge

1. Introduction

Welcome to Pyledge, a complete educational platform committed to effective learning and user management. Pyledge is intended to deliver a stimulating and user-friendly experience for students and teachers, resulting in a balanced and effective learning environment.

2. ERD



3. Data Dictionary

3.1 Users

Column	Data Type	Constraints	Description
id	INT	PK, Auto Increment	Unique identifier for each user
username	VARCHAR(50)	NOT NULL	Username of the user
email	VARCHAR(100)	NOT NULL	Email address of the user
password	VARCHAR(255)	NOT NULL	Password for authentication
type	ENUM('student', 'instructir')	NOT NULL	Type of user

3.2 Profile

Column	Data Type	Constraints	Description
id	INT	PK, Auto Increment	Unique identifier for each profile
User_id	INT	FK, NOT NULL	
name	VARCHAR(100)	NOT NULL	Name of user
email	VARCHAR(100)	NOT NULL, Unique	Email of user
password	VARCHAR(255)	NOT NULL	Password for authentication

3.3 Books

Column	Data Type	Constraints	Description
id	INT	PK, Auto Increment	Unique identifier for each book
title	VARCHAR(255)	NOT NULL	Title of the book
url	VARCHAR(255)	NOT NULL	URI to access book
Uploaded_by	INT	FK, NOT NULL	

3.4 Exercise

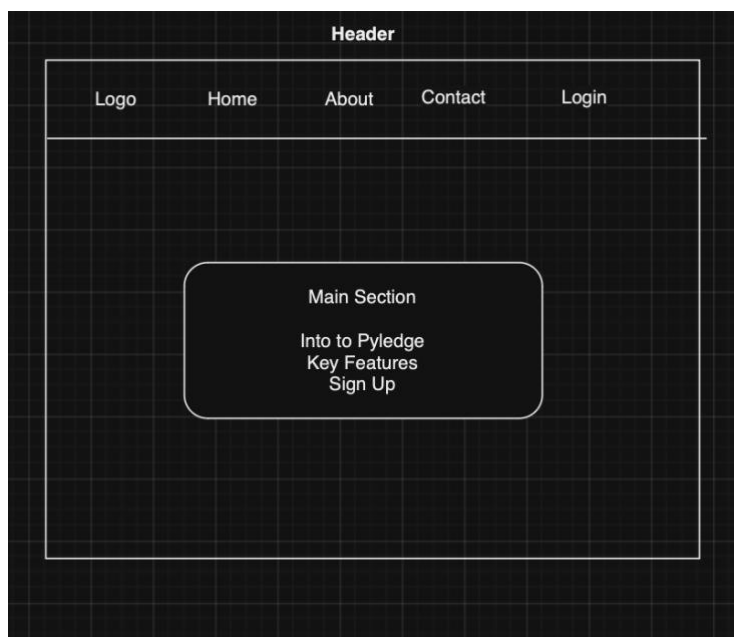
Column	Data Type	Constraints	Description
id	INT	PK, Auto Increment	Unique identifier for exercise
question	TEXT	NOT NULL	Question text
options	JSON	NOT NULL	JSON array
answer	VARCHAR(255)	NOT NULL	Correct Answer

3.5 UserAnswers

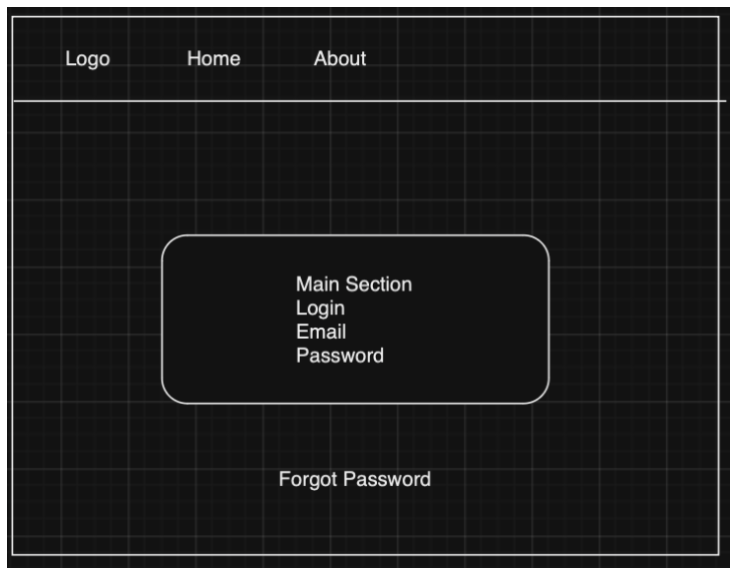
Column	Data Type	Constraints	Description
id	INT	PK, Auto Increment	Unique identifier for answer
User_id	INT	FK, NOT NULL	
exercise_id	INT	FK, NOT NULL	
selected_options	VARCHAR(255)	NOT NULL	The options selected by user
is_correct	BOOLEAN	NOT NULL	Indicates if selected option is correct

4. Wireframes

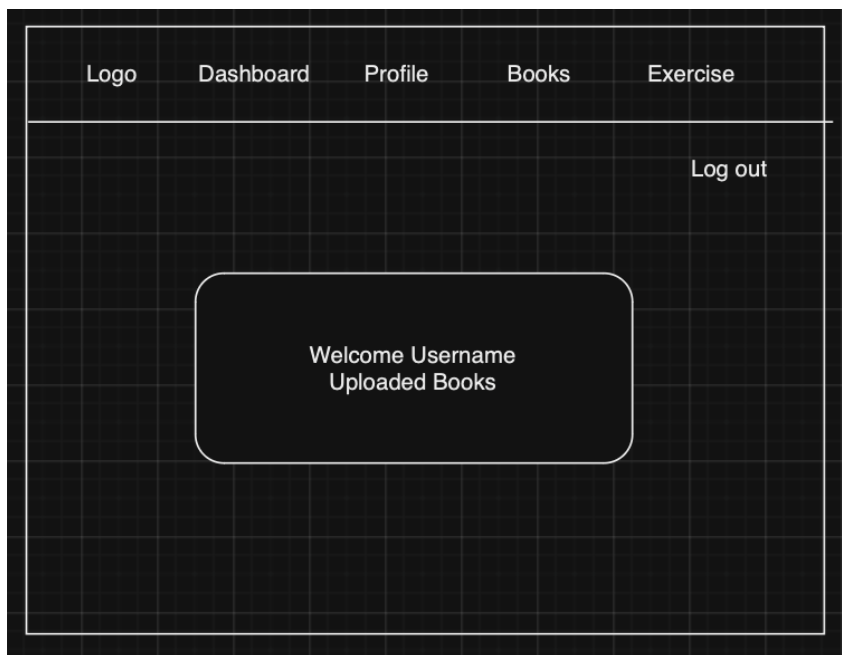
4.1 Header



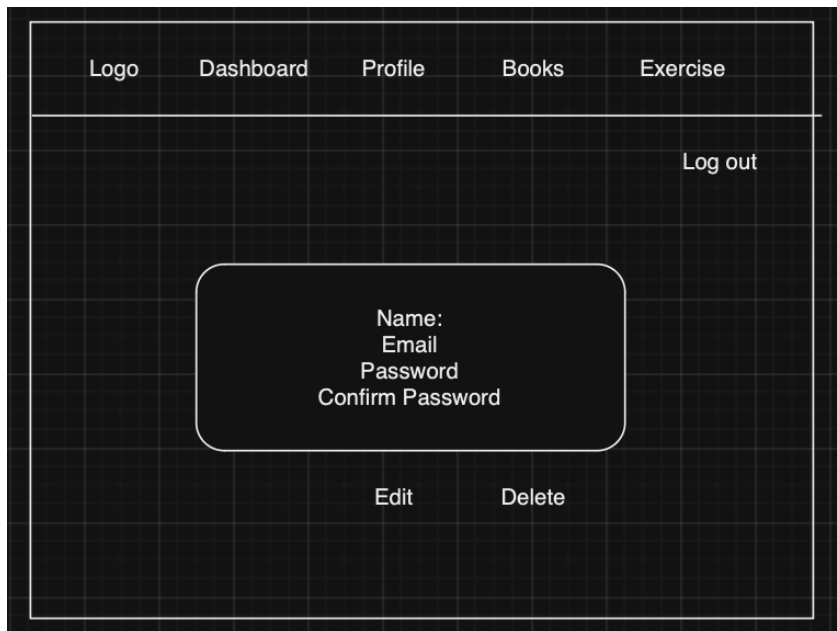
4.2 Login Page



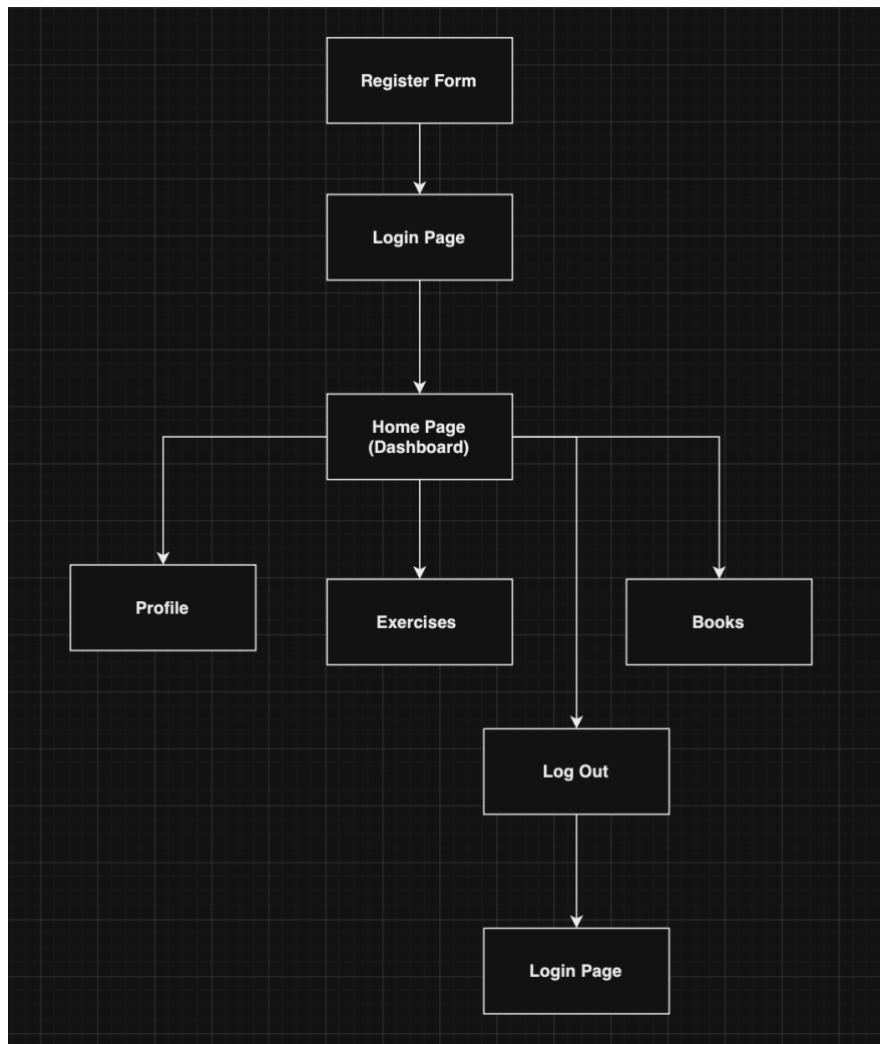
4.3 Dashboard



4.4 Profile Page



5. Website Navigational Structure



6. Code Snippet Explanation

1. User Authentication

a. Login

```

doctype html
html
  head
    title Login Page
    meta(charset="UTF-8")
    meta(name="viewport" content="width=device-width, initial-scale=1.0")
    link(rel="stylesheet" href="/stylesheets/login.css")
  body
    .container
      h1 Login
      form(action="/login", method="post")
        .form-group
          label(for="email") Email:
          input#email.form-control(type="email", name="email", required)
        .form-group
          label(for="password") Password:
          input#password.form-control(type="password", name="password", required)
        .form-group
          input.btn.btn-primary(type="submit", value="Login")
      .register-link
        p Not registered?
        a(href="/registerViews") Register

```

LoginViews renders html to view the login form

```

1
2 module.exports = function(db) {
3   const express = require('express');
4   const router = express.Router();
5
6   router.get('/logout', (req, res) => {
7     req.session.destroy(err => {
8       if (err) {
9         console.log('Error during logout:', err);
10        res.send('Error occurred during logout');
11      } else {
12        res.redirect('/loginViews');
13      }
14    });
15  });
16
17  return router;

```

Handles user log out functionality

b. Registration

```

2. const express = require('express');
3. const router = express.Router();
4.
5. module.exports = (db) => {
6.
7.   router.post('/register', (req, res) => {
8.     const { email, username, password, type } = req.body;

```



```

9.
10.   if (password.length < 6) {
11.       return res.status(400).json({ error: 'Password must be at least 6 characters long' });
12.   }
13.   const query = 'INSERT INTO users (email, username, password, type) VALUES (?, ?, ?, ?)';
14.
15.   db.query(query, [email, username, password, type], (err, results) => {
16.       if (err) throw err;
17.       res.redirect('/login')
18.   });
19. });
20.
21. return router;
22. };
23.

```

Registers password if requirements are met as well as email user name, and type of user and stores them in the database

```

$(document).ready(function() {
    $('#email').on('input', function() {
        var email = $('#email').val();
        console.log('Checking email:', email);

        $.ajax({
            url: '/checkEmail',
            data: { email: email },
            success: function(data) {
                console.log('Response received:', data);
                if (data.exists) {
                    $('#register-button').prop('disabled', true);
                    $('#register-button').css({
                        'background-color': '#ff7f6e',
                        'color': 'black'
                    });
                    $('#email').css('background-color', '#ffc0b8');
                    $('#email-error').text('Email already registered!').css('color', 'red');
                } else {
                    $('#register-button').prop('disabled', false);
                    $('#email').css('background-color', '#d7ffb8');
                    $('#register-button').css({
                        'background-color': '',
                        'color': ''
                    });
                    $('#email-error').text('Email is available!').css('color', 'green');
                }
            },
            error: function(err) {
                console.log('AJAX error:', err);
            }
        });
    });
});

```

The script send AJAX request to the server to validate the email is already registered or is not in the correct format.

24. Profile Management

```
doctype html
html
  head
    title Profile
    meta(charset="UTF-8")
    meta(name="viewport" content="width=device-width, initial-scale=1.0")
    link(rel="stylesheet" href="/stylesheets/profile.css")
    script(src="https://code.jquery.com/jquery-3.6.0.min.js")
  script.
    $(document).ready(function() {
      // Delete confirmation
      $('.delete-button').on('click', function(e) {
        if (!confirm('Are you sure you want to delete your profile?')) {
          e.preventDefault();
        }
      });

      // Session timeout
      let inactivityTime = function () {
        let time;
        window.onload = resetTimer;
        window.onmousemove = resetTimer;
        window.onmousedown = resetTimer;
        window.ontouchstart = resetTimer;
        window.ontouchmove = resetTimer;
        window.onclick = resetTimer;
        window.onkeypress = resetTimer;

        function logout() {
          window.location.href = '/logout';
        }

        function resetTimer() {

```

```

        clearTimeout(time);

        time = setTimeout(logout, 30000); // 30 seconds
    }
};

inactivityTime();
});
body
.container
    header
        h1 Profile
        a.back-button(href="/dashboard") Back to Dashboard
    main
        .profile-info
            p
                span.label Name:
                span.value #{user.name}
            p
                span.label Type:
                span.value #{user.type}
            p
                span.label Email:
                span.value #{user.email}
        .profile-actions
            a.edit-button(href="/profile/edit") Edit
            form(action="/profile/delete" method="POST" style="display:inline;")
                button.delete-button(type="submit") Delete

```

The Profile Views sets a delete and edit button where the edit redirects to another editView and route, the delete deletes user from db. It also has a 30 second user session time out that redirects back to the login page.

```

module.exports = function(db) {
    const express = require('express');
    const router = express.Router();

    // Profile page
    router.get('/profile', (req, res) => {
        if (!req.session.user) {

```

```

        return res.redirect('/login');
    }

    const user = {
        email: req.session.email,
        name: req.session.username,
        type: req.session.type
    };

    res.render('profileViews', { user });
});

// Edit profile page
router.get('/profile/edit', (req, res) => {
    if (!req.session.user) {
        return res.redirect('/login');
    }

    const user = {
        email: req.session.email,
        name: req.session.username,
        type: req.session.type
    };

    res.render('editViews', { user });
});

// Handle profile update
router.post('/profile/edit', (req, res) => {
    if (!req.session.user) {
        return res.redirect('/login');
    }

    const { email, name, password, confirmPassword } = req.body;

    if (password !== confirmPassword) {
        return res.send('Passwords do not match.');
```

```

    }

    const updateQuery = 'UPDATE users SET email = ?, username = ?, password = ? WHERE id = ?';
```

```

db.query(updateQuery, [email, name, password, req.session.user.id], (err, result) => {
  if (err) {
    return res.send('An error occurred.');
```

 }

 req.session.email = email;
 req.session.username = name;
 res.redirect('/profile');
});
});

// Handle profile deletion
router.post('/profile/delete', (req, res) => {
 if (!req.session.user) {
 return res.redirect('/login');
 }

 const deleteQuery = 'DELETE FROM users WHERE id = ?';
 db.query(deleteQuery, [req.session.user.id], (err, result) => {
 if (err) {
 return res.send('An error occurred.');
 }

 req.session.destroy((err) => {
 if (err) {
 return res.send('An error occurred while logging out.');
 }
 res.redirect('/login');
 });
 });
});

return router;
};

The profile also handles edit logic after redirecting to editViews the profile updates db username, email and password with a confirmation and a JQuery also that checks for password validity as provided in the following code

```

$(document).ready(function() {
    // Password validation
    $('#password, #confirmPassword').on('input', function() {
        var password = $('#password').val();
        var confirmPassword = $('#confirmPassword').val();

        if (password.length < 6) {
            $('#password').css('background-color', '#ffc0b8');
            $('#password-error').text('Password must be at least 6 characters!').css('color', 'red');
        } else {
            $('#password').css('background-color', '#d7ffb8');
            $('#password-error').text("");
        }

        if (password !== confirmPassword) {
            $('#confirmPassword').css('background-color', '#ffc0b8');
            $('#confirmPassword-error').text('Passwords do not match!').css('color', 'red');
        } else if (confirmPassword.length >= 6) {
            $('#confirmPassword').css('background-color', '#d7ffb8');
            $('#confirmPassword-error').text('Passwords match!').css('color', 'green');
        } else {
            $('#confirmPassword').css('background-color', "");
            $('#confirmPassword-error').text("");
        }
    });
});

```

25. Content Access

3.1 Books

```

doctype html
html
head
    title Books
    meta(charset="UTF-8")
    meta(name="viewport" content="width=device-width, initial-scale=1.0")
    link(rel="stylesheet" href="/stylesheets/book.css")

```

```

script(src="https://code.jquery.com/jquery-3.6.0.min.js")
script.
$(document).ready(function() {
  $('.dropdown').on('click', function() {
    $(this).toggleClass('open');
  });
});
body
.container
.header
h1 Books
.nav
ul.menu
li
  a(href="/profile") Profile
li.dropdown
  a(href="#") Menu
ul.dropdown-menu
li
  a(href="/exercises") Exercises
li
  a(href="/logout") Logout
main
a.back-button(href="/dashboard") Back to Dashboard
section
ul.books-list
each book in books
li
  a(href=book.url target="_blank")= book.title
span
  | (
  a(href=book.url download) Download
  | )

```

The booksViews structures a set of pdfs

```

module.exports = function(db) {
  const express = require('express');

```

```

const router = express.Router();

router.get('/books', (req, res) => {
  if (!req.session.user) {
    res.redirect('/login');
  } else {
    const books = [
      { title: 'Intro to Python Web', url: '/pdfs/Intro_Python_web.pdf' },
      { title: 'Learning Python', url: '/pdfs/Learning_Python.pdf' },
      { title: 'Python Basics', url: '/pdfs/python_basics.pdf' },
      { title: 'Python', url: '/pdfs/python.pdf' },
    ];

    res.render('booksViews', { books, username: req.session.user.username });
  }
});

return router;
};

```

While the route also has a user session timeout as well as the url of the pdfs giving them a link to download them.

3.2 Exercises

```

const express = require('express');
const router = express.Router();

const questions = [
  {
    question: 'What is the output of print(2 ** 3)?',
    options: ['5', '6', '7', '8'],
    answer: '8'
  },
  {
    question: 'What does the following code do: list1 = [1, 2, 3]; list2 = list1; list1.append(4)?',
    options: ['list2 is [1, 2, 3, 4]', 'list1 is [1, 2, 3]', 'list2 is [4]', 'list1 is [4]'],
    answer: 'list2 is [1, 2, 3, 4]'
  },
  {

```



```
question: "What is the output of print('Hello World'[::-1])?",
options: ["dlroW olleH", "Hello World", "World Hello", "dlroW olleH "],
answer: "dlroW olleH"
},
{
question: "Which of the following is a mutable data type in Python?",
options: ["tuple", "str", "list", "int"],
answer: "list"
},
{
question: "Which keyword is used for function in Python?",
options: ["func", "def", "function", "define"],
answer: "def"
},
{
question: "What does the // operator do in Python?",
options: ["Addition", "Division", "Integer Division", "Exponentiation"],
answer: "Integer Division"
},
{
question: "Which function is used to find the length of a string in Python?",
options: ["len()", "length()", "size()", "strlen()"],
answer: "len()"
},
{
question: "What is the output of print(type(10))?",
options: ["<class 'float'>", "<class 'str'>", "<class 'int'>", "<class 'list'>"],
answer: "<class 'int'>"
},
{
question: "Which keyword is used to create a class in Python?",
options: ["class", "def", "function", "object"],
answer: "class"
},
{
question: "What is the correct file extension for Python files?",
options: [".python", ".pyth", ".py", ".pt"],
answer: ".py"
}
];
```

```

router.get('/exercises', (req, res) => {
  if (!req.session.user) {
    return res.redirect('/loginViews');
  }

  // Shuffle and select 5 random questions
  const shuffledQuestions = questions.sort(() => 0.5 - Math.random());
  const selectedQuestions = shuffledQuestions.slice(0, 5);

  res.render('exerciseViews', { questions: selectedQuestions });
});

router.post('/exercise/submit', (req, res) => {
  const userAnswers = req.body;
  let score = 0;

  const selectedQuestions = req.session.selectedQuestions || [];

  selectedQuestions.forEach((question, index) => {
    const userAnswer = userAnswers[`question-${index}`];
    console.log(`Question ${index + 1}: ${question.question}`);
    console.log(`User Answer: ${userAnswer}`);
    console.log(`Correct Answer: ${question.answer}`);
    if (userAnswer && userAnswer === question.answer) {
      score += 2; // Each question is worth 2 marks
    }
  });

  console.log(`Final Score: ${score} out of 10`);
  res.json({ score, total: 10 });
});

module.exports = router;

```

The exercises sets 10 questions and views randomly 5 of them and renders the ExerciseViews the users answering the question add up points which are revealed in a pop out window out of 10 after submitting the exercise.

```

doctype html
html

```

```

head
  title Exercises
  meta(charset="UTF-8")
  meta(name="viewport" content="width=device-width, initial-scale=1.0")
  link(rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css")
  script(src="https://code.jquery.com/jquery-3.6.0.min.js")
  script(src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js")
body
  .container
    header
      h1 Exercises
      a.back-button(href="/dashboard") Back to Dashboard
    main
      form#exercise-form
        each question, index in questions
          .question
            p Question #{index + 1}: #{question.question}
            ul
              each option, optionIndex in question.options
                li
                  input(type="radio" id=`question-${index}-${optionIndex}` name=`question-${index}` value=option)
                  label(for=`question-${index}-${optionIndex}`) #{option}
            button.btn.btn-primary(type="submit") Submit

// Include the modal for displaying the score
.modal.fade#scoreModal(tabindex="-1" role="dialog")
.modal-dialog(role="document")
.modal-content
.modal-header
  h5.modal-title Score
  button.close(type="button" data-dismiss="modal" aria-label="Close")
  span(aria-hidden="true") &times;
.modal-body
  p#score-text You scored 0 out of 10
.modal-footer
  button.btn.btn-secondary(type="button" data-dismiss="modal") Close

script.
$(document).ready(function() {
  $('#exercise-form').on('submit', function(event) {

```

```

event.preventDefault();

const formData = $(this).serialize();

$.ajax({
  type: 'POST',
  url: '/exercise/submit',
  data: formData,
  success: function(response) {
    $('#score-text').text(`You scored ${response.score} out of ${response.total}`);
    $('#scoreModal').modal('show');
  },
  error: function() {
    alert('An error occurred. Please try again.');
```

```

  }
});
});

// Close modal
$('.close').on('click', function() {
  $('#scoreModal').hide();
});
});

```

The exerciseViews gives a few main function where the main one is to be able to pick one answer only per question using radio buttons.

3.3 Dashboard

This is the main page that redirects to most pages which has a user session time out

```

const express = require('express');
const router = express.Router();

module.exports = (db) => {

  router.get('/dashboard', (req, res) => {
    if (!req.session.user) {
      res.redirect('/login');
```

```

    } else {
      res.render('dashboard', { username: req.session.username });
    }
  });

  router.get('/logout', (req, res) => {
    req.session.destroy(err => {
      if (err) {
        console.error(err);
        res.send('Error occurred during logout');
      } else {
        res.redirect('/login');
      }
    });
  });

  return router;
};

```

As well as a View that has an introduction and basic information about the website

```

doctype html
html
  head
    title Dashboard
    meta(charset="UTF-8")
    meta(name="viewport" content="width=device-width, initial-scale=1.0")
    link(rel="stylesheet" href="/stylesheets/dashboard.css")
    script(src="https://code.jquery.com/jquery-3.6.0.min.js")
    script.
      $(document).ready(function() {
        $('.dropdown').on('click', function() {
          $(this).toggleClass('open');
        });

        let inactivityTime = function () {
          let time;

          window.onload = resetTimer;
          window.onmousemove = resetTimer;
          window.onmousedown = resetTimer; // catches touchscreen presses as well

```

```
window.ontouchstart = resetTimer; // catches touchscreen swipes as well
window.ontouchmove = resetTimer; // required for touchscreen devices
window.onclick = resetTimer; // catches touchpad clicks as well
window.onkeypress = resetTimer;
```

```
function logout() {
  window.location.href = '/logout'; // redirects to logout route
}
```

```
function resetTimer() {
  clearTimeout(time);
  time = setTimeout(logout, 30000); // 30 seconds of inactivity
}
};
```

```
  inactivityTime();
});
```

body

.container

header

h1 Welcome, #{username}!

nav

ul.menu

li

[a\(href="/profile"\)](/profile) Profile

li.dropdown

[a\(href="#"\)](#) Menu

ul.dropdown-menu

li

[a\(href="/exercises"\)](/exercises) Exercises

li

[a\(href="/books"\)](/books) Books

li

[a\(href="/logout"\)](/logout) Logout

main

section

h2 Introduction to Python

p

| Python is a powerful and versatile programming language that is easy to learn and fun to use.

| Whether you are a beginner or an experienced programmer, Python has something to offer.
| With Python, you can develop web applications, automate tasks, analyze data,
| create machine learning models, and much more.

p

| Python's syntax is designed to be readable and straightforward,
| which makes it an excellent choice for people who are new to programming.
| The Python community is also very active and supportive, providing a wealth of resources
| and libraries that you can use to enhance your projects.

section

h2 About Our Website

p

| Our website is dedicated to helping you learn and master Python programming.
| We offer a variety of tutorials, exercises, and projects that are designed to teach you
| the fundamentals of Python and help you apply what you have learned in real-world scenarios.

p

| You can track your progress through your profile, complete exercises to reinforce your learning,
| and work on projects that will help you build a portfolio of Python applications.
| Join our community of learners and start your journey to becoming a Python expert today!

NOTE: Any kind of breaches to a page before the user session starts through a log in will be redirected back to the login page

```
outer.get('/dashboard', (req, res) => {  
  if (!req.session.user) {  
    res.redirect('/login');
```

Every Page has that.