FrOW LOGO Fruits on The Web

Documentation

Authors

Sorodoc Tudor Cosmin

Galan Andrei Iulian

Ignat Gabriel Andrei

1. Abstract

This document presents the development and evolution of a web application called "FrOW - Fruits On the Web". The application is designed to provide an engaging and educational gaming experience for users, especially primary school students, to learn about different fruits and vegetables, or even to practice their math.

2. Introduction

2.1 Purpose

The motivation behind selecting this subject is to provide an engaging educational experience for primary school students by introducing them to new fruits and integrating math into the gameplay.

2.2 Intended Audience and Reading Suggestions

The game is designed to be accessible to children of different backgrounds and abilities, and efforts will be made to ensure that the game is inclusive and easy to use. Additionally, teachers and parents who are interested in supplementing their children's education with a game-based learning tool may also find the FrOW project useful.

3. Overall Description

3.1 Product Functions

- An user can register to the application.
- An user can log in to the application.
- An user can log out.
- An user can view and edit his profile.
- An user can view the leaderboard, the top 10 best players.
- An user can play a game.
- An user can contact the team in case they encounter some difficulties using the app.
- An user can view the "About Us" section, where he can reach out to the team contributors.

4. External Interface Requirements

4.1 User Interfaces

4.1.1 Home Page

The home page is the first page that the user will see when he opens the application. It contains a navigation bar.

1. If the user is not logged in, the navigation bar will contain the following options:
    - Home
    - Contact
    - About
    - Login

Also, in the bottom of the home page, there will be a button that will redirect the user to the login page.

2. If the user is logged in, the navigation bar will contain the following options:
    - Home
    - Play
    - Ranking
    - Contact
    - About
    - Your Profile
    - Logout

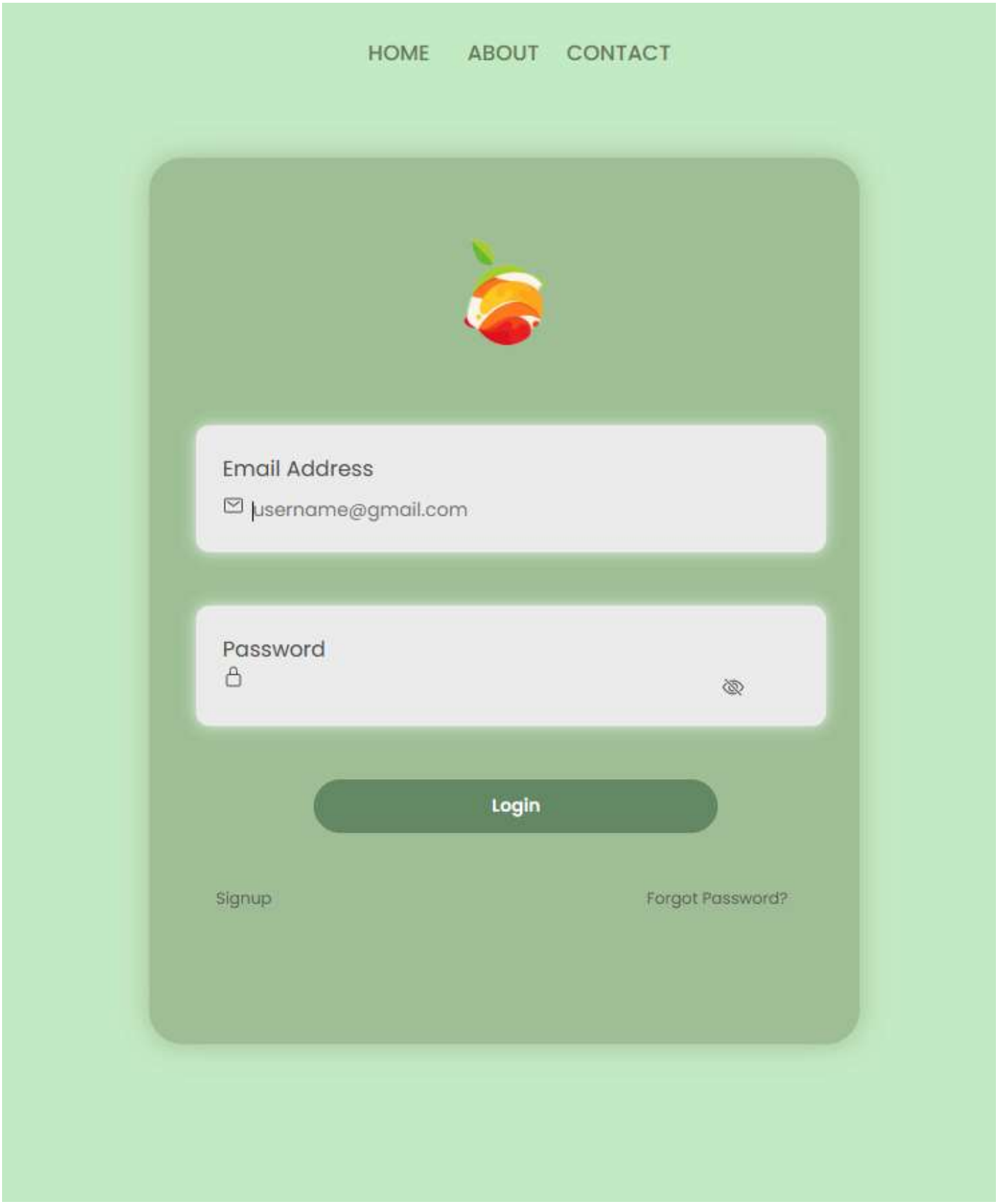
    Also, in the middle of the home page, there will be a welcoming message with the user first and last name and a button that will redirect the user to the play page.

There is also a footer containing copyrights reserved, in all the pages excepting Play Page.

4.1.2 Login Page

The login page is the page where the user can log in to the application. It contains a navigation bar.

1. The navigation bar will contain the following options:
    - Home
    - Contact
    - About
2. The login page will contain a form with the following fields:
    - Email Address
    - Password
3. If the user is not registered, he can click on the "Sign Up" button, which will redirect him to the signup page.
4. If the user forgot his password, he can click on the "Forgot Password?" button, which will redirect him to the forgot password page.



4.1.3 Sign Up Page

The sign up page is the page where the user can register to the application. It contains a navigation bar, with the same as fields as the Login Page.

1. The sign up page will contain a form with the following fields:
    - First Name
    - Last Name
    - Email Address
    - Username
    - Password
    - Repeat Password
    - Genre

    After he fills the form, the user can click on the "Register" button, which will redirect him to the Login Page.

4.1.4 Forgot Password Page

The forgot password page is the page where the user can reset his password. It contains a navigation bar, with the same as fields as the Login Page.

1. The forgot password page will contain a form with the following fields:
    - Email Address

    After he fills the form, the user can click on the "Send Code" button, which will redirect him to the "Verify Page" where he will enter the code that he received on the email.

4.1.5 Verify Page

The verify page is the page where the user can enter the code that he received on the email. It contains a navigation bar, with the same as fields as the Login Page.

1. The verify page will contain a form with the following fields:
    - Code: 4 digits

    After he fills the form, the user can click on the "Submit" button, which will redirect him to the "Reset Password Page" where he will enter the new password. Or he can click on the "Resend Code" button.

4.1.6 Reset Password Page

The reset password page is the page where the user can enter the new password. It contains a navigation bar, with the same as fields as the Login Page.

1. The reset password page will contain a form with the following fields:
    - Password
    - Repeat Password

After he fills the form, the user can click on the "Reset Password" button, which will redirect him to the Login Page.

4.1.7 About Page

The about page contains information about the developers team, their social accounts,how they chose the project and finally how to play the game. Like in the Home Page, the header will have different buttons depending on the log in status.

4.1.8 Contact Page

The contact page contains a form with the following fields:

- Enter your name
- Enter your email
- Enter the description of the encountered problems or questions, curiosities.

Then he will likely click the "Send Now" button which will send us the message.

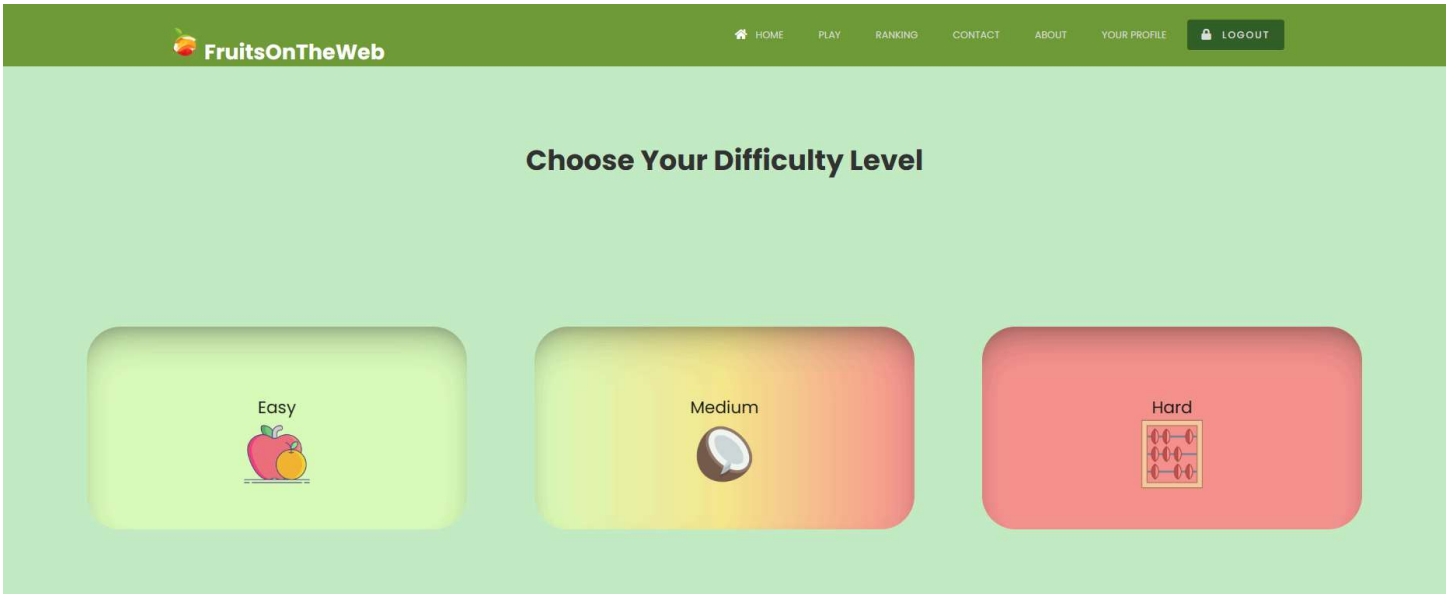
Also there are some other information about the emails/phone-number where the members can be contacted.

4.1.9 Play Page

The play page is the page where the user can play the game. It contains a navigation bar, with the same as fields as the Home Page.

At this page you can select the level:

- Easy
- Medium
- Hard



After the player chose the level, he will be redirected to the actual questions.

4.1.10 Question Page

The question page is the page where the user can answer the question. It contains an image/mathematical equation, a question and 4 possible answers, with only one correct answer. There is a next button, which will redirect the player to the next question.

Depending on the answer, the player will receive a certain amount of points or he will lose a life. When reaching the last question(the 10th), there will be a button called "Finish" redirecting him to the Congratulation Page. If the player loses all his lives, he will be redirected to the Game Over Page.

4.1.11 Ranking Page

This page will display the top 10 players with the highest score. The leaderboard will be updated in real-time.

There will be displayed the following:

- Position
- Username
- Score
- Playing Since

In the case the player is not in the leaderboard he can view his position in the profile page.

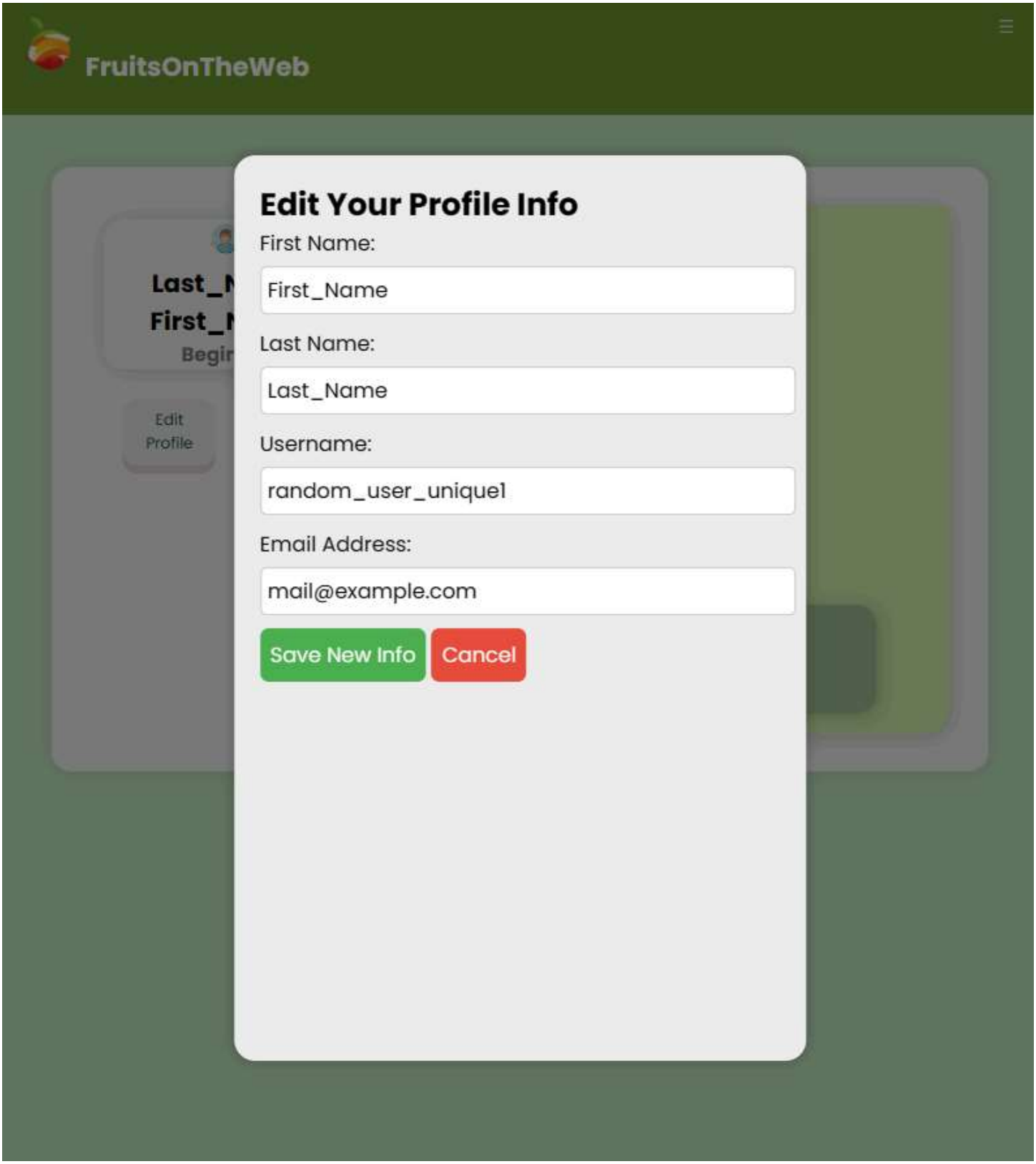
4.1.12 Profile Page

The profile page will provide information about the user:

- First Name
- Last Name
- Username
- Email
- Points
- Ranking

There are 2 buttons which will allow you to edit your information in the same page:

- Edit profile
- Change password



4.1.13 Logout

The logout button will redirect the user to the home page for logged out users.

5. Other Requirements

5.1 Responsiveness

rogue scientists All pages within the scope of this project have been developed with a responsive design approach, utilizing specific media queries for optimal display on a range of devices and screen sizes.

Example: play page for mobile devices

```
@media screen and (max-width: 768px) {

  .option-easy, .option-medium, .option-hard {
    width: 100%;
    max-width: 100%;
    margin: 0;
    padding: 2em;
  }
  .choose-text h1{
    font-size: 1.5em;
  }
```

How it looks:

FruitsOnTheWeb

Choose Your Difficulty Level

Easy

Medium

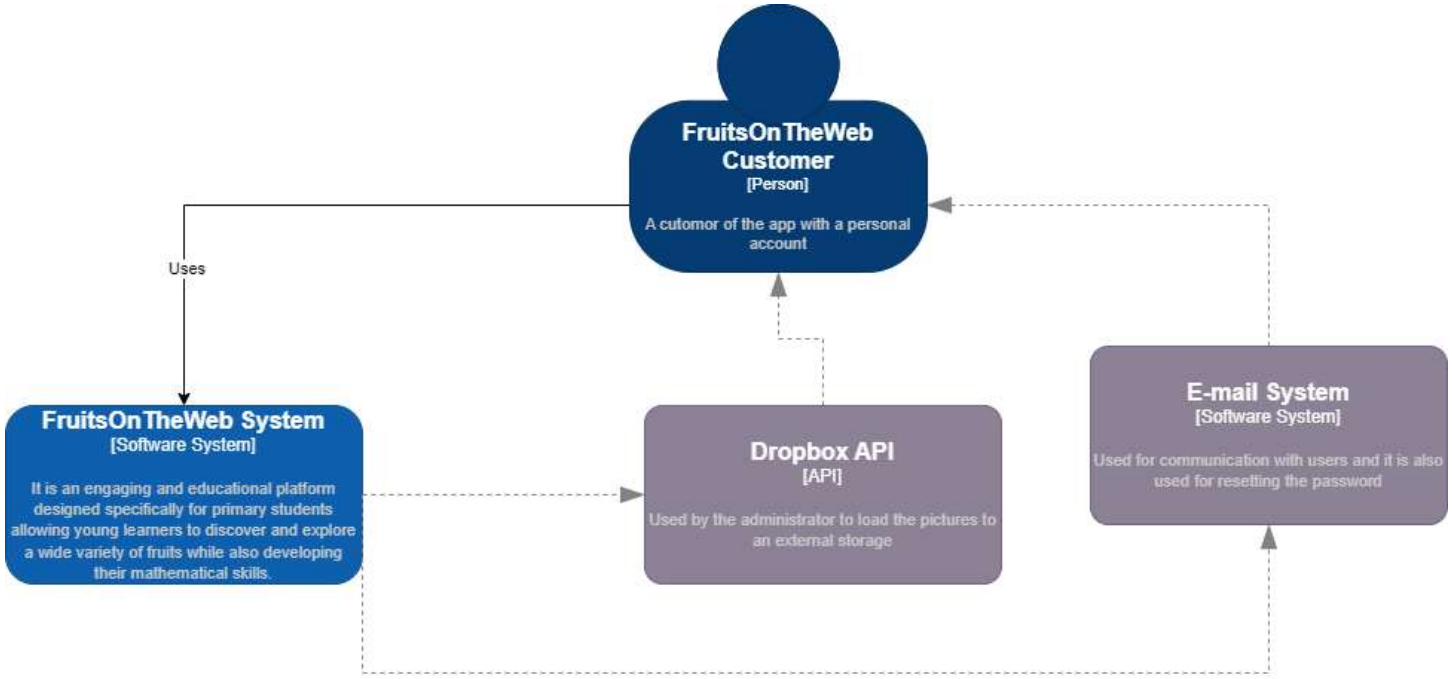
5.2 User Experience

- Colors:
  - light green theme: was chosen for the project based on its association with freshness, health, and nature, which aligns well with the theme of fruits and vegetables. Additionally, the color green is known to have a calming effect on the mind, which can be beneficial for a learning environment. The shade of light green was specifically selected to create a pleasing and harmonious visual experience for the intended audience of primary school students.
  - light blue for sign up: sky blue is a versatile color that can work well with a variety of other colors, making it easy to create a visually appealing and engaging design for the web application.

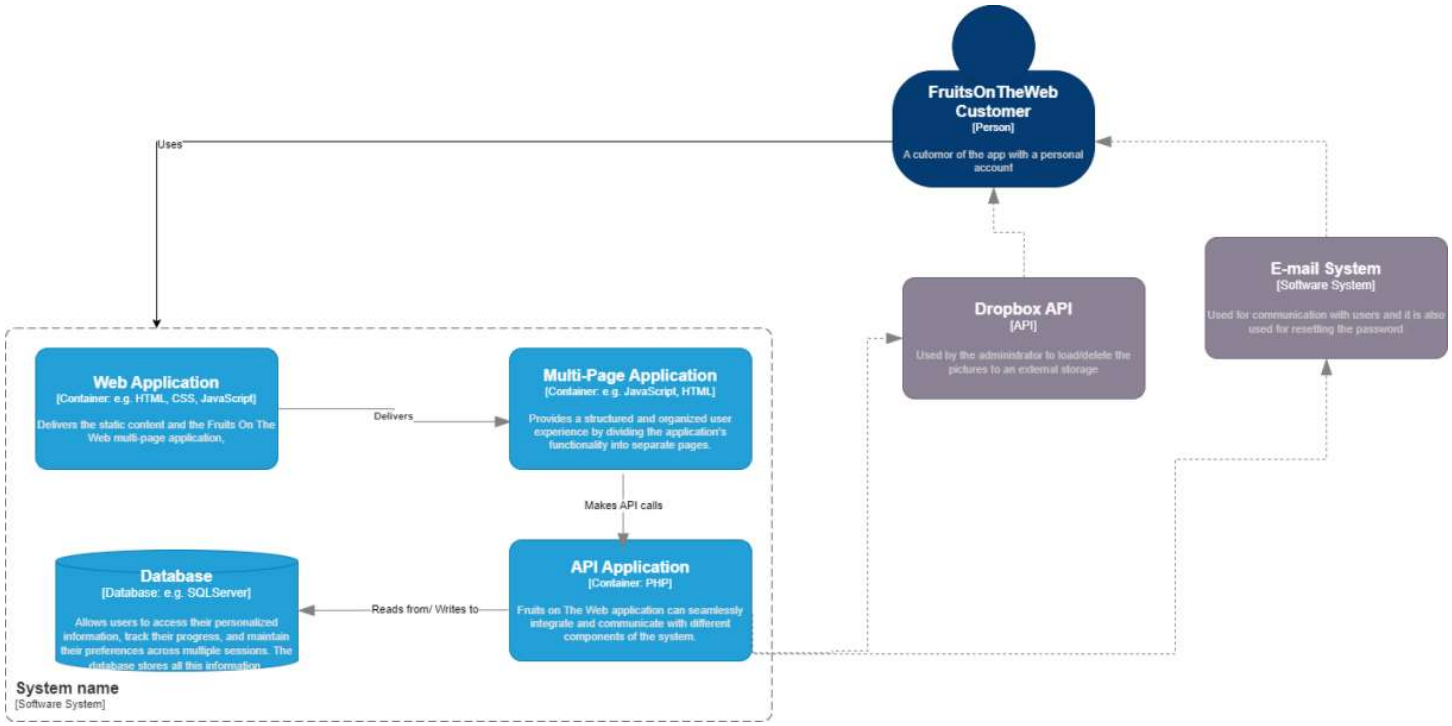
6. Backend Description

6.1 C4 diagrams

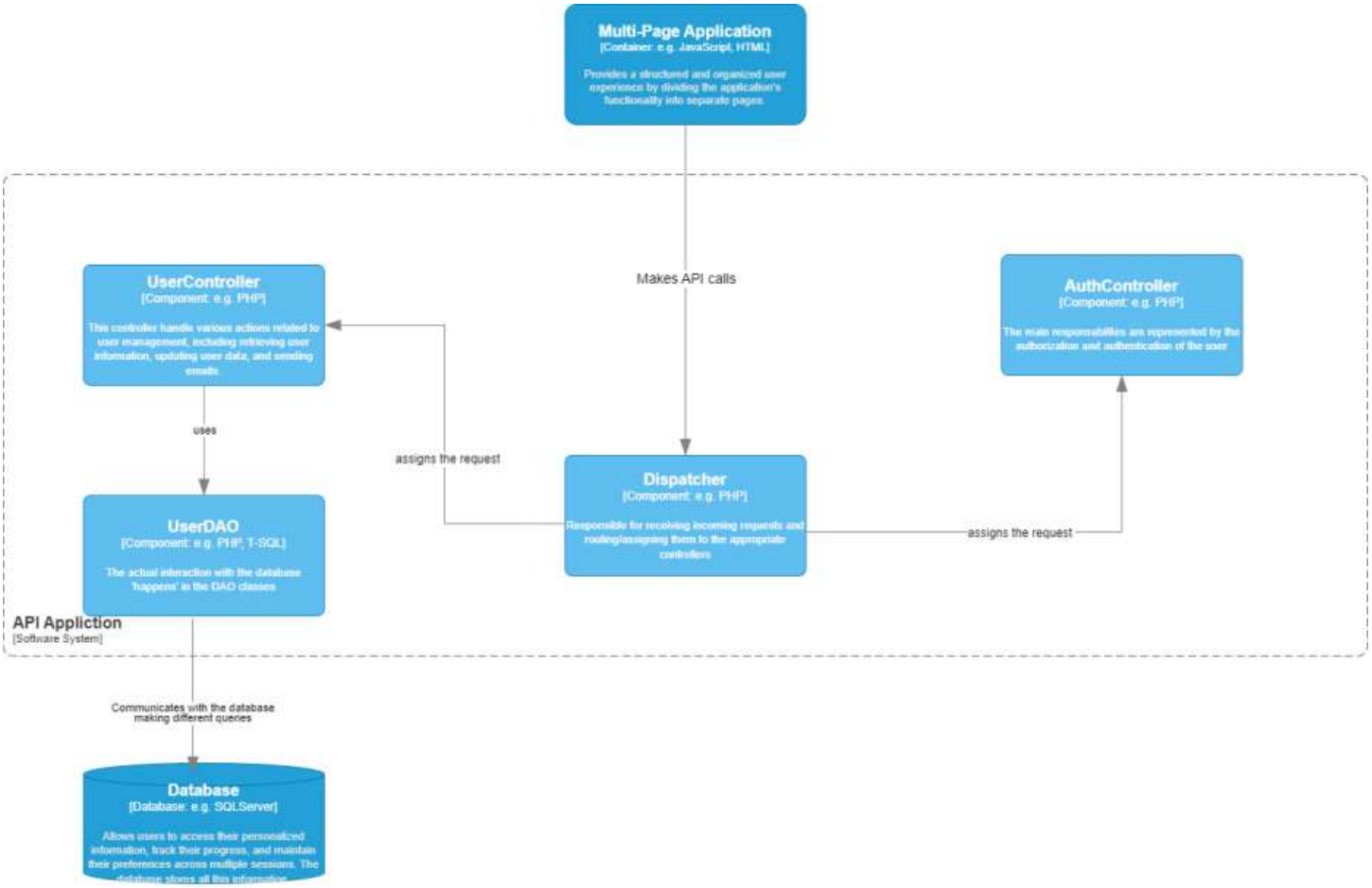
Level 1



Level 2



Level 3



6.2 Introduction

The backend of our web project serves as the server-side infrastructure that handles data processing, logic implementation, and integration with external systems.

It is developed using PHP and MS SQL(from Microsoft) database for storage. The database is hosted on Microsoft Azure.

6.3 Architecture

The backend uses some of the concepts of **MVC**(Model-View-Controller) architecture. In this sense, the backend is divided into three main components: **Models**, **Controllers**, and **DAOs**(classes for database operations). This allows for easier maintenance and testing of the code, as well as better scalability and reusability of the code.

Our backend follows the REST architectural style, which is a widely adopted design pattern for building web services. REST allows for a scalable and stateless communication approach between clients and servers, promoting loose coupling and interoperability. Using REST, our backend exposes a set of RESTful APIs that adhere to the principles of resource-oriented design. These APIs enable clients to interact with the backend by performing standard CRUD (Create, Read, Update, Delete) operations on

resources. Each resource in our system is uniquely identified by a URL (Uniform Resource Locator), and clients can utilize different HTTP methods, such as GET, POST, PUT, and DELETE, to perform operations on these resources. The backend returns responses in a standardized format, JSON (JavaScript Object Notation), which allows for easy consumption by client applications. By adopting **REST**, our backend ensures a uniform and predictable interface for clients, facilitating easy integration with various front-end applications and external systems. It also enables **scalability** and **extensibility**, allowing us to add new resources and functionalities to the backend API without impacting existing clients.

6.4 Technologies used

Languages: PHP, JavaScript, T-SQL, HTML, CSS

Server: Apache from XAMPP

Email Service: swiftmailer(downloadable with composer)

Image Storage: Dropbox

6.5 Endpoints

Public APIs used

Dropbox APIs

In order to access the content of a Dropbox we permanently need to have an access token. This token is generated by the Dropbox API and it is unique for each user. This token is automatically refreshed by calling an API provided by Dropbox.

- **https://api.dropboxapi.com/2/check/user**

  This API is used to check if the current access token is still valid or not. If it is not valid, the token is refreshed.

- **https://api.dropbox.com/oauth2/token**

  This API is used to generate a new access token. We need to provide the client id, the client secret and a refresh token(this token will never expire) in order to generate the access token.

- **https://content.dropboxapi.com/2/files/upload**

  This API is used to upload a file to a Dropbox. We need to provide the access token, file pointer and size of the file. Also in the header we need to specify the path where the file will be uploaded. It returns the file path in the Dropbox.

- **https://api.dropboxapi.com/2/sharing/create_shared_link_with_settings**

  This API is used to create a shared link for a file(used to display images in the html pages). We need to provide the access token and the file path in the Dropbox. It returns the shared link, which will be inserted in the database.

- **https://api.dropboxapi.com/2/files/delete_v2**

  This API is used to delete a file from a Dropbox. We need to provide the access token and the file path in the Dropbox.

6.6 Data Models and Databases

6.6.1 Models

User class:

Fields:

- id
- firstName
- lastName
- username
- email
- password -> hashed int the database
- gender
- points
- ranking
- created_at
- reset_code -> used for reseting the password
- verified -> used to verify if the email is valid

Equation class:

Fields:

- id
- equation_text

Picture class:

Fields:

- id
- text
- pathInDropbox -> this is the file path in the dropbox
- downloadLink -> this is the generated link, later used in html pages

Question class:

Fields:

- id
- question_text
- difficulty

- points
- id_picture -> the id of the picture associated with this question(can be null)
- id_equation -> the id of the equation associated with a math related question(can be null, but not when id_picture is null)

Answer class:

Fields:

- id
- answer_text
- is_correct
- question_id

6.6.2 Database

Database class:

This class is used to get a connection to the database. It is a singleton class, so we can have only one instance of it. It has a field "connection" which is the connection to the database, that will be returned when "getConnection" will be called.

6.6.3 DAOs

UserDAO, QuestionDAO, PictureDAO, EquationDAO, AnswerDAO

All these classes are used to access the database. They have methods for inserting, updating, deleting and selecting from the database.

SECURITY: SQL injection is prevented by using prepared statements and binding parameters to their expected types.

6.6.4 Controllers

UserController class

This controller class is used to handle the requests with the "/users" path.

The methods are:

- "get": all users, with a specific id, username or email
- "post": send email from contact page
- "put": modify the user info, password, add points
- "delete": delete an user -> only the admin

AuthController class

This controller class is used to handle the requests with the "/auth" path. This controller is important for the authentication and authorization of the users, and also overall security.

The methods are:

- "post": login, register, reset password, enter-code, change password, send email, verify email

QuestionController class

This controller class is used to handle the requests with the "/questions" path.

The methods are:

- "get": all questions, with a specific id, a full quizz
- "post": add a question -> only the admin
- "put": modify a question -> only the admin
- "delete": delete a question -> only the admin

PictureController class

This controller class is used to handle the requests with the "/pictures" path.

The methods are:

- "get": all pictures, with a specific id

- "post": add a picture -> only the admin
- "put": modify a picture -> only the admin
- "delete": delete a picture -> only the admin

EquationController class

This controller class is used to handle the requests with the "/equations" path.

The methods are:

- "get": all equations, with a specific id
- "post": add an equation -> only the admin
- "put": modify an equation -> only the admin
- "delete": delete an equation -> only the admin

AnswerController class

This controller class is used to handle the requests with the "/answers" path.

The methods are:

- "get": all answers, with a specific id, all answers for a question id
- "post": add an answer -> only the admin
- "put": modify an answer -> only the admin
- "delete": delete an answer -> only the admin

6.6.5 Dispatcher

This class handles incoming HTTP requests and routes them to the appropriate controller or handler based on the requested URL or route configuration. It maps URLs to corresponding actions or functions that handle the requested operation.

6.6.6 RSS

This class updates the rss feed when user's ranking is updated(increase points, delete/insert user)

6.7 Authentication and Authorization

The authentication is done using JWT. The user sends the email and password to the server. The server checks if the email and password are correct. If they are, the server sends back a JWT token. The user will use this token to access the protected routes. The token is stored in the local storage. The token is valid for 24 hour. After 24 hour, the user will have to login again.

The admin is the only one who can add, modify or delete question, pictures, equations, answers and users.

The other users can play a game, see the ranking board, see their profile, modify their profile, see the contact page.

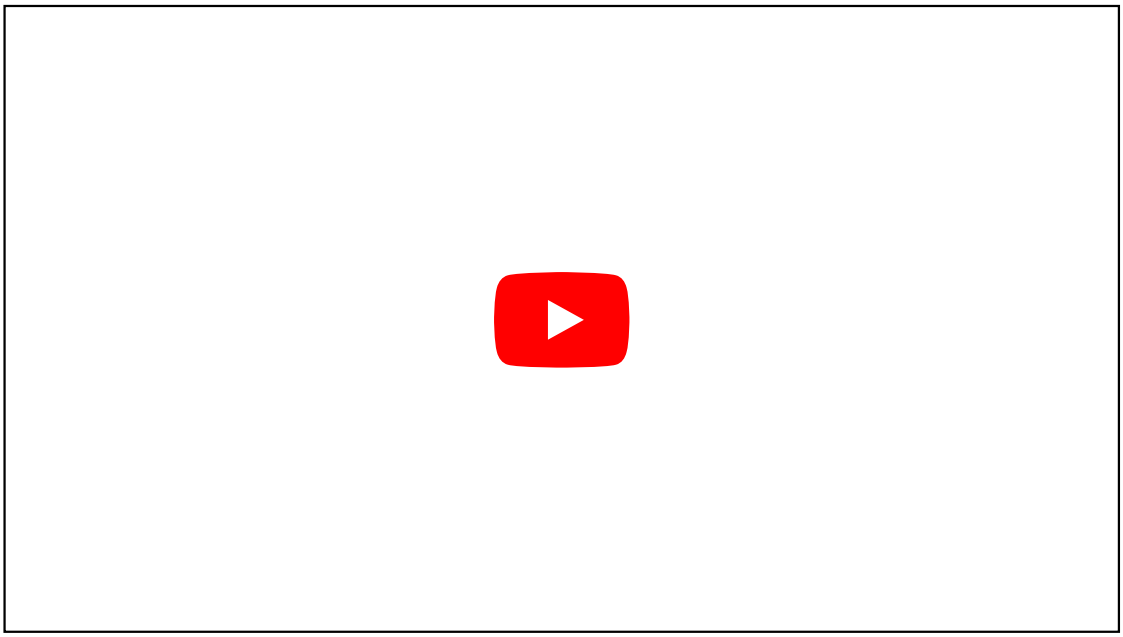
Validating the email on register

When the user registers, he will receive an email with a link to verify his email. Accessing the link will verify the email.

Reset password

If the user forgets his password, he can reset it. He will receive an email with a reset code. He will have to enter the code and then the new password.

7. Demo



8. References

Images:

- Icons for play page, question page, etc
- Images of fruits and vegetables, alpacas

Materials:

- TW course
- TW laboratory