Certﬁcation of Project completion

# Rudimentary Checkbook Balancing Application.

April 2023

**BY**

# Table of content

[Acknowledgement](#acknowledge) 2

[E-Project Analysis](#analysis) 3

[User guide / Developer guide](#guide) 4

[Source code](#source_code) 7

[Checklist of validation](#checklist) 14 [Submission checklist](#submission) 15



# Acknowledgement

We would like to acknowledge all those who have given moral support and helped us make the project a success.

We would also like to express our gratitude to all the staff members of our department for not only providing us with the opportunity to work on this project and increase our experience, but also for their support and encouragement throughout the process.

We also express our sincere gratitude to our project guide, for his guidance and support for the completion of this project.

And lastly, thanks to all our colleagues for their valuable suggestions and supportive feedback.

# E-Project Analysis

The development this program is a relatively straightforward project that aims to create a user-friendly program that accurately displays information on the trains, their schedule, time of departure and arrival, destination and other relevant information all for a train api. It can also calculate the train’s estimated time of arrival. The program displays the list of trains available and their assigned train number and also displays the individual information and schedules for each train based on their train number.

The success of the project depended on proper planning, development, and testing to ensure that the program meets the user's needs and expectations. Our team carefully analyzed the requirements, designed the program, and developed it using the appropriate programming language, tools, and techniques. Unit tests and manual testing helped identify and address issues related to invalid inputs and incorrect balance calculation

Moreover, we ensured that the program is user-friendly, easy to use and understand, with clear and concise documentation that guides users through the program's features and functionalities.

Overall, the project offers an opportunity to develop a useful and practical program that simplifies the task of gaining up to date railway information.

# User / Developer guide

**User Guide:**

* The program displays a button for viewing the list of available trains and input bar to collect data (train number) that allows the user to view individual railway information.
* The user is given the chance to choose any of the given options.
* On clicking the ‘View Train List’ Button the user is shown a table with a list of all available trains plus their train number and train id.
* The user can use the train number to access the information of each individual train by inputting it into the input bar back at the home page and clicking the ‘View individual Train Info’ Button.
* This shows a table containing The trains train id, train name, train number, it’s station of departure, station of arrival, time of departure, time of arrival and duration of travel.
* It also shows the estimated time of arrival based on the distance to be covered and the trains average speed.

**Developer Guide:**

Document Title: Train API Web Interface

Purpose:

This HTML document, in conjunction with its associated PHP files, forms a wed interface for accessing the Train API. The interface facilitates users in retrieving comprehensive railway information, including train details and schedules.

Document Structure:

index.html: -

Purpose: Serves as the frontend interface for accessing the train API, facilitating users in retrieving up to date railway information.

Contents:

* HTML Declaration to specify the document type and version of HTML being used.
* Root element which defines the root element of the HTML document, indicating the language as English.
* Head Section which contains the metadata and external resources related to the HTML document.
* Body Section which contains the main content of the HTML document.

trainDetails.php: -

Purpose: Display a list of available trains retrieved from the Train API.

Contents:

* HTML structure defining the document layout, including header and body sections.
* PHP code for fetching data from the train API and displaying train information in a tabular format.
* Integration with external CSS file for styling and layout adjustments.

Styling:

* Utilizes CSS properties defined in “styke.css” to enhance visual appeal and alignment.

trainSchedule.php: -

Purpose: Provides detailed schedules and information for a specific train based on user input.

Contents:

* HTML structure defining the document layout, including header and body sections.
* PHP code for retrieving train details and schedules information from the train API based on user input (train number).
* Calculations of expected time of arrival based on Distance, Speed and departure and arrival times.
* Display of train Schedule and calculated time of arrival in a tabular format.
* Integration with external CSS file (‘style.css) for styling and layout adjustments.

style.css: -

Purpose: Provides styling rules and layout adjustments for HTML elements across all pages.

Content:

* CSS rules defining styles for header, navigation items, buttons, and other elements.
* Background image configuration for the header section to enhance visual appeal.
* Styling of navigation items, including hover effects for improved user interaction.
* Button styling for consistent appearance and user experience.

# Source Code

Index.html:-

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

        <link rel="stylesheet" href="./style.css">

    </head>

    <body>

        <!-- Application header -->

        <div class="header">

            <h1>TRAIN API</h1>

            <h3>A RESTful API to get up to date

                railway information.

            </h3>

            <div style="text-align: left; padding: 40px;">

                <div>

                    <!-- Form to access trainDetails.php -->

                    <form method="get"action ="trainDetails.php">

                        Train List: <button type="submit">View Train List</button>

                    </form>

                </div>

                <div>

                    <!-- Form to access trainSchedule.php -->

                    <form method="get"action ="trainSchedule.php">

                        Enter Train Number : <input type="text" name="t\_no">

                        <button type="submit">Track Train</button>

                    </form>

                </div>

            </div>

        </div>

    </body>

</html>

trainDetails.php:-

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Train API Web Interface</title>

        <link rel="stylesheet" href="./style.css">

    </head>

    <body style="padding: 5%; background-image: url(./images/train2.jpeg);">

        <!-- Application header with inline styling -->

        <div class="header" style="padding: 0;height: 40vh; background: none;">

            <h1>TRAIN API</h1>

            <h3>

                A RESTful API to get up to date

                railway information.

            </h3>

        </div>

            <?php

                // Create a variable containing the url to the Train API.

                $url\_trainlist = "https://api.railwayapi.site/api/v1/trains?q=126";

                // Assign $data the function used to collect data from the url

                $data = file\_get\_contents($url\_trainlist);

                $json = json\_decode($data,true);

                // Create a table display information gotten from the url

                $i =0;

                echo "<table border=1><tr><td>S.No.</td><td>Train id</td><td>Train Name</td><td>Train Number</td></tr>";

                while(isset($json['data'][$i]['id'])){

                    $num = $i + 1;

                    echo "<tr><td>".$num."</td>";

                    echo "<td>". $json['data'][$i]['id'] ."</td>";

                    echo "<td>". $json['data'][$i]['trainName'] ."</td>";

                    echo "<td>". $json['data'][$i]['trainNumber'] ."</td></tr>";

                    $i++;

                }

                echo "</table>";

            ?>

    </body>

</html>

TrainSchedule.php:-

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title>Document</title>

        <link rel="stylesheet" href="./style.css">

    </head>

    <body style="padding: 5%; background-image: url(./images/train2.jpeg);">

        <!-- Application header with inline styling -->

        <div class="header" style="padding: 0;height: 40vh; background: none;">

            <h1>TRAIN API</h1>

            <h3>A RESTful API to get up to date

                railway information.

            </h3>

        </div>

        <?php

            $t\_no = $\_GET['t\_no']; // variable to get train number from index.html.

            // Create a variable containing the url to the Train API.

            $url\_traindetails = "https://api.railwayapi.site/api/v1/trains/".$t\_no."? fullSchedule=false";

            // Assign $data the function used to collect data from the url

            $data = file\_get\_contents($url\_traindetails);

            $json = json\_decode($data,true);

            // Create a table display information gotten from the url

            $i =0;

            echo "<table border=1><tr><td>S.No.</td><td>Train id</td><td>Train Name</td><td>Train Number</td>";

            echo "<td>Station of Departure</td><td>Station of Arrival</td><td>Time of Departure</td>";

            echo "<td>Time of Arrival</td><td>Duration of Travel</td></tr>";

            while(isset($json['data'][$i]['id'])){

                $num = $i + 1;

                echo "<tr><td>".$num."</td>";

                echo "<td>". $json['data'][$i]['id'] ."</td>";

                echo "<td>". $json['data'][$i]['trainName'] ."</td>";

                echo "<td>". $json['data'][$i]['trainNumber'] ."</td>";

                echo "<td>". $json['data'][$i]['stationFrom']['stationName'] ."</td>";

                echo "<td>". $json['data'][$i]['stationTo']['stationName'] ."</td>";

                echo "<td>". $json['data'][$i]['departureTime'] ."</td>";

                echo "<td>". $json['data'][$i]['arrivalTime'] ."</td>";

                echo "<td>". $json['data'][$i]['duration'] ."</td></tr>";

                $i++;

            }

            echo "</table>";

            // Calculate estimated time of arrival

            $distance = $json['data'][0]['distance'];

            $speed = $json['data'][0]['avgSpeed'];

            $expectedTimeOfArrival = $distance/$speed;

            $hours = floor($expectedTimeOfArrival);

            $minutes = ( $expectedTimeOfArrival - $hours) \* 60;

            echo "Estimated time of arrival is ". $hours . " hours and " . round($minutes) . " minutes from Time of Depature.";

        ?>

    </body>

</html>

style.css:-

/\* Gneral styling \*/

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

/\* Style for application header and background display \*/

.header{

    background-image: url(./images/train2.jpeg);

    text-align: center;

    padding: 10%;

    font-size: 200%;

    background-size: cover;

    background-repeat: no-repeat;

    height: 100vh;

}

/\* Button Style \*/

button{

    height: 60px;

    width: 100px;

    background-color: orangered;

    border: none;

    border-radius: 5%;

    margin: 10px;

}

**Checklist Of Validation**

|  |  |
| --- | --- |
| **Option** | **Validated** |
| Have all the modules been properly integrated and are completely functional? | Yes |
| Is exception handling mechanism implemented in all the modules? | Yes |
| Are all blocks of code, meaningfully commented? | Yes |
| Are all methods and classes properly named? | Yes |
| Are all the program codes working properly? | Yes |



# Submission Checklist

|  |  |  |
| --- | --- | --- |
| **Particulars** | **Answer** | **Comments** |
| Has all modules been tested thoroughly? | Yes | The entire program has been overtly tested. |
| Does the program run adequately on the given software & hardware requirement? | Yes | The program was tested under the given requirement. |
| Does the Documentation give comprehensive details about the project? | Yes | Time was taken to make sure the Documentation accurately represented the project. |