SETU Code Lab
Research Document

Diarmuid O'Neill

South East Technological University

1/12/2025

Table of Contents

Abstract

SETU Code Lab is a gamified, web-based study tool for computing students and lecturers at SETU (South East Technological University). Its main goal is to motivate students to study through gamification and convenience, while saving lecturers time when conducting lab work by automating the grading process. This report outlines the research conducted prior to development and aims to address the most critical questions and decisions relating to the project.

This report explores potential technologies for developing the front end, back end, database, and in-built code editor. It also examines code sandboxing—what it is, why it is needed, and how it could be implemented. Furthermore, it discusses how automated testing can be implemented, the available options for gamification and their benefits, hosting, and how similar platforms operate. The resources used for gathering information include official documentation and websites, related blog posts, and several reputable books, journals, and conference proceedings (see bibliography).

Following the research, it was decided that React, TypeScript, SCSS, and Vite will be used to implement the front end of the system. Node.js and Express will be used to implement the back end, and PostgreSQL has been chosen as the database. CodeMirror 6 will serve as the in-built code editor, and Docker will be used to execute submitted code safely in isolation. DigitalOcean's Droplet VPS (Virtual Private Server) hosting service will be used to host the system. Lecturers will define skeleton functions along with their parameters and return types. Test cases will consist of test inputs and expected outputs, allowing for automated testing. The system will also employ leaderboards, points, a rating system, and daily login streaks to keep students motivated to study. These features have been shown to have a positive impact on computer science education overall.

Introduction

SETU Code Lab is a gamified in-browser study tool for students studying computing related courses at SETU and their lecturers. Its purpose is to help students study common coding concepts and help Lecturers conduct and grade student lab work. At its core, it is a repository of coding problems. Students can choose a problem from this repository and solve it in an in-browser code editor. Lecturers can create new problems and test cases and add them to the repository. They can also create class groups, add students to these class groups and assign problems to them so they can easily view all their student's results.

Problem Statement

Today, it has become harder than ever for students to focus and take the time to understand the complex and abstract concepts required to succeed in computing related courses. In university students and lecturers often wrestle with several different platforms and extensions just to conduct coding assignments or study for upcoming coding exams. This wastes student and lecturer time and sometimes causes students to lose focus and spend less time studying.

 First an Integrated Development Environment (IDE) is needed, sometimes the necessary extensions are needed to use a specific language, and then a Learning Management System (LMS) such as Microsoft Teams, Blackboard, Google Classroom or Moodle is needed for giving students assignments, receiving submissions and allowing lecturers to manually grade student work.

Value Proposition

SETU Code Lab aims to motivate students to study through short, exam-focused coding problems and the use of some gamification elements. It also aims to eliminate some of the friction associated with conducting and grading lab work for lecturers, by allowing them to assign coding problems to groups of students and automatically receive their results.

Research

This document aims to answer some key design questions such as what front-end, back-end and database technologies are most suitable for this project. It will also look at how the system will handle student code submissions safely, how student submissions will be tested and graded, what gamification elements can be used to enhance the user experience and what similar platforms exist already.

Front-End

The front-end of SETU Code Lab needs to be clean, responsive and user-friendly so that users can focus on solving coding problems rather than learning how to use the platform. The chosen technology stack needs to support the latest two versions of Firefox, Google Chrome, and Microsoft Edge and remain readable and usable on the top five most popular desktop resolutions in Europe (1920x1080, 1536x864, 1366x768, 1280x720, 2560x1440) (Statcounter, 2025). Below are some of the technologies considered.

React

React is an open-source JavaScript library developed by Meta (formerly Facebook) and first released in 2013. It is used to build user interfaces and works by allowing developers to create reusable user interface (UI) components. These components can then be put together to construct a full UI and can render differently depending on props or state (GeeksforGeeks, 2025).

 React also uses a virtual DOM (Document Object Model) to optimize rendering and improve performance by only updating the parts of the UI that have changed and not the whole page. It is also fully compatible with TypeScript. The below code example demonstrates a simple counter component using React's `useState` hook. When `count` is updated, React automatically re-renders the component to reflect the changed state in the UI (Meta Platforms, Inc., 2025).

```
function MyButton() {
  const [count, setCount] = useState(0);

  function handleClick() {
    setCount(count + 1);
  }

  return (
    <button onClick={handleClick}>
      Clicked {count} times
    </button>
  );
}
```

A key aspect of React is one-way data binding. This means that data can only flow from parent components to child components using props. In this example the parent component passes the `greet` variable as a prop to the child component.

```
function Parent() {
    const greet = "Hello, React!";
    return <Child message={greet} />;
}

function Child({ message }) {
    return <p>{message}</p>;
}
```

This makes React code easier to debug, manage, predict, reuse and scale. There are also additional state management libraries available if needed such as Redux which allows global state management or Vuex which allows two-way binding (GeeksforGeeks, 2025).
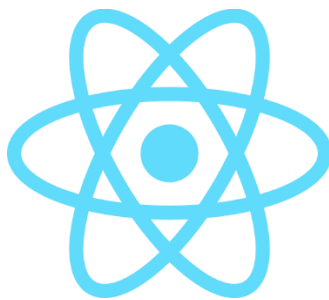

Fig.1. React Logo **Source:** (Wikimedia, 2025)

Angular

Angular is full-featured, TypeScript based framework developed by Google and released in 2016, with a more opinionated architecture suitable for larger-scale, enterprise-level applications. It uses a component-based architecture like React and includes two-way-binding. This means that a change made to the UI can automatically update the component's data model and vice versa. Two-way binding requires careful synchronization of data and can sometimes make code harder to maintain, debug and reuse.

Angular also offers useful back-end development features such as routing, HTTP services, form handling and dependency injection. This reduces the need for external libraries.

The below code example shows how an Angular component is defined. Angular components require a component decorator denoted by the `@Component` tag (Google, 2025).

```
// user-profile.ts
@Component({
  selector: 'user-profile',
  templateUrl: 'user-profile.html',
  styleUrl: 'user-profile.css',
})
export class UserProfile {
  // Component behavior is defined in here
}
```

 (Google, 2025).

Fig. 2. Angular Logo **Source:** (Google, 2025)

Vue

Vue is another front-end framework developed by Evan You and first released in 2014. Vue was considered as a potential front-end technology for this project due to its simplicity, component-based architecture, declarative templates, high performance and reactive two-way binding. This allows automatic synchronization of the UI and the underlying data (GeeksforGeeks, 2025).

The following code example illustrates how Vue binds the `count` variable to the button and updates the UI automatically. It also shows how Vue extends HTML syntax can declaratively render based on JavaScript/TypeScript state.

```
import { createApp, ref } from 'vue'

createApp({
  setup() {
    return {
      count: ref(0)
    }
  }
}).mount('#app')

template
<div id="app">
  <button @click="count++">
    Count is: {{ count }}
  </button>
</div>
```

(Vue.js, 2025)

Fig. 3. Vue Logo **Source:** (Wikimedia, 2025)

Django Template Language

Django is a full-stack web framework written in Python. This means it can be used for the front end and back end of the web application. It uses Django Template Language (DTL) to allow developers to generate HTML pages dynamically on the server and render them for users. This also means that both the front end and back end could be written using the same framework (Django Software Foundation, 2025).

Django Template Language promotes the separation of logic and presentation by using template tags, filters and variables. When a template is rendered, Django evaluates it into HTML and the logic determines what is displayed to the user. The following code example demonstrates inheritance from the base layout `base_generic.html`, references variables like `section.title`, applies a filter `|upper` to a variable and iterates through `story_list` using the `{%for%}` tag (Django Software Foundation, 2025).

```
{% extends "base_generic.html" %}
{% block title %}{{ section.title }}{% endblock %}
{% block content %}
<h1>{{ section.title }}</h1>
{% for story in story_list %}
<h2>
  <a href="{{ story.get_absolute_url }}">
    {{ story.headline|upper }}
  </a>
</h2>
<p>{{ story.tease|truncatewords:"100" }}</p>
{% endfor %}
{% endblock %}
```

(Django Software Foundation, 2025).

Blazor

Blazor is a front-end web framework developed by Microsoft that uses HTML, CSS and C#. Using Blazor would allow a full .NET based technology stack. It supports a component-based architecture like other frameworks which promotes reusability. Blazor also allows developers to call into existing JavaScript libraries and APIs from C# and can run either on the server (Blazor Server) or directly in the browser via WebAssembly (Blazor WebAssembly) (Microsoft, 2024).

Blazor uses plain HTML tags for UI composition. The following example demonstrates a simple counter app which updates dynamically as a button is pressed.

```
<PageTitle>Counter</PageTitle>
<h1>Counter</h1>
<p role="status">Current count: @currentCount</p>
<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>
@code {
    private int currentCount = 0;
    private void IncrementCount()
    {
        currentCount++;
    }
}
```

(Microsoft, 2024)

Blade

Laravel is a full-stack web framework based on PHP. Laravel uses the Blade templating engine which generates HTML pages dynamically and supports reusable components and template inheritance. This is like Django Template Language and provides a separation of logic and presentation. Blade also supports template inheritance and reusable components (Laravel, 2025).

Blade views can be returned from routes using the global `view()` helper shown in the below code example.

```
Route::get('/', function () {

    return view('welcome', ['name' => 'Samantha']);

});
```

The value of `$name` is passed from the route to the view and displayed within the Blade template as shown below.

```
Hello, {{ $name }}.
```

Blade templates can also extend base layouts and define sections as shown in the following example.

```
<!-- resources/views/child.blade.php -->

@extends('layouts.app')


@section('title', 'Page Title')

@section('sidebar')

    @@parent

    <p>This is appended to the master sidebar.</p>

@endsection

@section('content')

    <p>This is my body content.</p>

@endsection
```

(Laravel, 2025)

11

Vite

Vite is a modern front-end build tool and development server known for its fast speed, simplicity and support of modern browsers. It offers Hot Module Replacement (HMR) which updates the application instantly in the browser without needing a full page reload. This can be used with React, Vue, Angular and vanilla JavaScript/TypeScript and is very useful for rapid development.

The following example shows the minimal commands needed to install and set up a local development server using Vite.

```
#Creates Vite project
npm create vite@latest
cd my-project

#Install dependencies
npm install

#Run the local server
npm run dev
```

Once the server is running, Vite will run the application locally. Any changes to the source code will automatically update in the browser via HMR (Vite, 2024).

Fig. 5. Vite Logo **Source:** (WikiMedia, 2025)

TypeScript

TypeScript is a syntactic superset of JavaScript developed by Microsoft and first released in 2012. It extends JavaScript by adding static typing, which allows compile-time type checking. This means TypeScript will report any mismatched type errors before running the code whereas JavaScript will not. This is very helpful for debugging and helps improve the quality and performance of the code (Microsoft, 2025).

The code examples below show typing and the creation of objects in TypeScript. This is the key difference between TypeScript and JavaScript. In the first example TypeScript will infer the types of String and Int for name and id.

```
const user = {
  name: "Hayes",
  id: 0,
};

interface User {
  name: string;
  id: number;
}
```


Fig. 6. TypeScript Logo **Source:** (Microsoft, 2025)

SCSS

SCSS is a stylesheet language that is compiled into CSS. SCSS syntax is fully compatible with CSS and includes more advanced features such as variables, nested rules, mixins, and built-in modules. SCSS variables differ from CSS variables in that CSS variables have different values for different elements, whereas SCSS variables remain the same across multiple elements. This is very useful for reducing code repetition. Nested rules further reduce code repetition by letting an inner rule inherit selectors from an outer rule. The following code example demonstrates this.

```scss
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }

  li { display: inline-block; }

  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

Mixins are reusable blocks of CSS that can be defined and used throughout the stylesheet mixed in with other styles. The following example shows how mixins are defined using the `@mixin` tag and how they are used using the `@include` tag.

```scss
@mixin reset-list {
  margin: 0;
  padding: 0;
  list-style: none;
}

@mixin horizontal-list {
  @include reset-list;

  li {
    display: inline-block;
    margin: {
      left: -2px;
      right: 2em;
    }
  }
}

nav ul {
  @include horizontal-list;
}
```

The Built-in modules feature provides functions that are useful for manipulating numbers, strings, colours and more, making it easier to build dynamic stylesheets (SASS Team, 2025).

Tailwind CSS

Tailwind CSS uses utility classes for rapid development. Tailwind utility classes allow developers to combine many single purpose utility classes directly in the markup i.e. not in a dedicated CSS file. This allows changes to be made faster as the developer does not have to consider what to name their classes and how their changes will affect other pages. Tailwind is also efficient as it purges any unused CSS, reducing the final bundle size (Tailwind Labs, 2025).
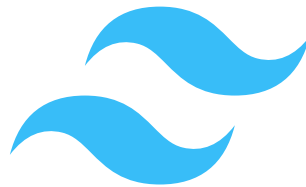

Fig. 7. Tailwind CSS Logo **Source:** (Tailwind Labs, 2025)

One issue with Tailwind CSS is that its utility first approach makes readability and maintainability more difficult as all styles are applied in an inline fashion. This bloats the codebase and becomes a headache for the developer (see code example below). Another issue with Tailwind is that it's utility first design and lack of dedicated stylesheets is unfamiliar to new developers.

```
<div class="mx-auto flex max-w-sm items-center gap-x-4 rounded-xl bg-white
p-6 shadow-lg outline outline-black/5 dark:bg-slate-800 dark:shadow-none
dark:-outline-offset-1 dark:outline-white/10">
  <img class="size-12 shrink-0" src="/img/logo.svg" alt="ChitChat Logo" />
<div>
    <div class="text-xl font-medium text-black dark:text white">ChitChat
</div>
<p class="text-gray-500 dark:text-gray-400">You have a new message!</p>
</div>
</div>
```

(Tailwind Labs, 2025)

Selected Front-End Technologies

The chosen technologies for the front-end are React, Vite, TypeScript, and SCSS.

React was chosen as the front-end framework for SETU Code Lab due to its unidirectional data flow which simplifies debugging and state management. The MobX state management library allows for the implementation of two-way binding if needed (MobX, 2025). React also allows for increased interactivity as opposed to Django template language, Blazor and Blade. With React, the user interface can update in real time without needing a full page reload. It is also more flexible and integrates more naturally with the chosen technology stack, particularly TypeScript which provides strong static typing and full compatibility with JSX the syntax extension used by React (Microsoft, 2025).

Vite has been chosen as a build tool for use with React to speed up development and for its native support of the latest versions of modern browsers. SCSS has been chosen ahead of Tailwind CSS because of its better readability, maintainability, additional features and familiar design methodology.

Back-End

The back end of SETU Code Lab is responsible for all server-side operations and data processing. It will implement CRUD (Create, Retrieve, Update, Delete) functionality for lots of different types of data such as, code problems, test cases, profiles, submissions and metrics. It will also handle user authentication allowing role-based access (e.g. Student and Lecturer) and integrate with the chosen code sandboxing technology to ensure user-submitted code is executed safely.

Node.js

Node.js is an open-source JavaScript runtime environment. It allows developers to run JavaScript outside of the web browser. It is asynchronous and event-driven meaning it can handle multiple tasks at once without blocking others, making it very efficient. It is built on Google's V8 JavaScript engine, which compiles JavaScript into machine code, resulting in a high level of performance. It's large ecosystem of packages available through NPM (Node Package Manager) also make it a popular choice for backend development (OpenJS Foundation, n.d.).

The below code example shows how to set up a web server that responds with 'Hello, World!' using Node.js.

```
const http = require('node:http');
const hostname = '127.0.0.1';
const port = 3000;
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello, World!\n');
});
server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Choosing Node.js as part of the backend technology stack allows both the front-end and back-end of SETU Code Lab to be based on TypeScript simplifying integration.

Fig. 8. Node.js Logo **Source:** (OpenJS Foundation, 2025)

Express

Express is the most popular web framework for Node.js. It is unopinionated, lightweight and fast. It provides features such as routing, HTTP helpers, powerful middleware functions and middleware error handling. It is also used for building RESTful APIs and simplifies database interactions (W3Schools, n.d.).

The below code example shows how to set up a web server that responds with 'Hello World!' using Express. This also demonstrates routing and the use of a HTTP helper in Express.

```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => {
  res.send('Hello World!')
})

app.listen(port, () => {
  console.log(`Example app listening on port ${port}`)
})
```

(OpenJS Foundation, 2025)

Express supports built-in, third party and custom middleware which can be used for authentication, logging and more. This modular middleware design helps separate concerns, making code more maintainable and reusable. Because of its low overhead and simple API, Express is widely used for building RESTful APIs and microservices (GeeksforGeeks, 2025).

Fig. 9. Express Logo **Source:** (OpenJS Foundation, 2025)

Django

As mentioned in the front-end technologies section, Django is a full-stack web framework written in Python. Django offers many built-in features for back-end development like authentication and authorization tools, Object Relational Mapping (ORM) and an admin interface (GeeksforGeeks, 2025).

The Django admin interface seen below allows developers to add, edit and delete records, manage user groups and their permissions and can be extended if needed to address project requirements (GeeksforGeeks, 2025). Its middleware system provides a structured way to process requests and responses. This enables developers to handle security, sessions, and logging in a modular and reusable way (Django Software Foundation, 2025).
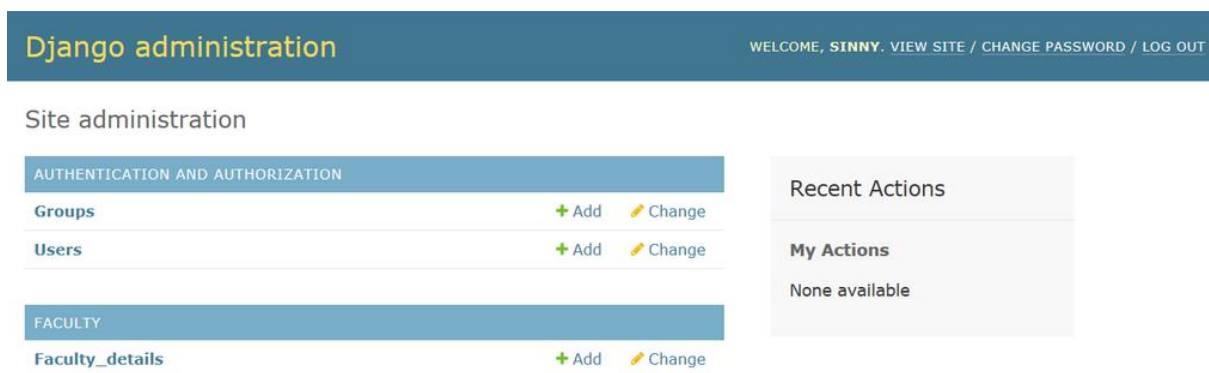
Fig. 10. Django admin interface **Source:** (GeeksforGeeks, 2025)

Fig. 11. Django Logo **Source:** (Django Software Foundation, 2025)

Flask

Flask is a lightweight Python micro web framework. This means unlike Django it is more unopinionated and flexible about how features are implemented. However, it does include features such as URL routing and the Jinja2 template engine which make routing and developing a front end easier. Features such as authentication and authorization are not included by default and require extensions (Flask-Login and SQLAlchemy) (GeeksforGeeks, 2025) (Pallets Projects, 2025).

The following Python code example shows how to set up a minimal web server and a basic route in Flask.

```
from flask import Flask

app = Flask(__name__)


@app.route('/')
def hello():
    return 'Hello, World!'
```

(Walia, 2025)

Fig. 12. Flask Logo **Source:** (Pallets Projects, 2025)

FastAPI

FastAPI is a fast web framework, used for building APIs with Python. As its name suggests, FastAPI has very high performance claiming to be on par with Node.js. It is fully compatible with the Pydantic library allowing custom data types and validation. It also supports automatic API documentation generation using either Swagger UI or ReDoc (Ramírez, n.d.).

The following code example shows how to set up a web server using Python and FastAPI.

```
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
def read_root():
    return {"Hello": "World"}
```

(GeeksforGeeks, 2025)

Fig. 13. FastAPI Logo **Source:** (Ramírez, n.d.)

Laravel

Laravel is a full-stack web framework based on PHP. It is highly opinionated and features built-in authentication and authorization tools, URL routing and generation, an ORM making it easy to interact with the database and uses the Blade templating engine to dynamically generate HTML pages. Laravel also includes a Command Line Interface (CLI) called Artisan. This gives developers useful commands that they can use to automate common tasks such as generating boilerplate code, running tests, and performing database migrations (Laravel, 2025).

The following example defines a simple route in Laravel:

```
use Illuminate\Support\Facades\Route;


Route::get('/greeting', function () {
    return 'Hello World';
});
```

(Laravel, 2025)

The below example shows how authentication middleware can be added to a Laravel route.

```
Route::get('/profile', ProfileController::class)
    ->middleware('auth');
```

(Laravel, 2025)

Fig. 14. Laravel Logo **Source:** (Ramírez, n.d.)

Selected Back-End Technologies

Node.js and Express have been chosen as the back-end technologies for SETU Code Lab. This is due to their seamless integration with React and TypeScript, high performance, large number of useful libraries available through NPM, and their asynchronous, event driven architecture. This should allow for rapid, high-quality development and give the developer fine-tuned control over implementation.

Database

Types of Databases

A database is where data is stored, organized and managed for an application. **Relational (SQL)** databases hold structured data, and **Non-Relational (NoSQL)** databases can hold semi-structured and unstructured data. For example, a relational database will store data in tables containing specific rows and columns whereas a non-relational database stores data differently depending on what it is. Relational databases naturally scale vertically (e.g. one database scales up) and non-relational databases typically naturally scale horizontally (e.g. the number of databases increases) (MongoDB, n.d.).

Structured Query Language (SQL) is a high-level programming language for manipulating data in relational databases. SQL statements can be written to create, retrieve, update and delete specific data in a database. For example, the following SQL statement:

`INSERT INTO Mattress_table (brand_name, cost)` VALUES('A','499');

inserts the value 'A' into the `brand_name` column and '499' into the `cost` column in the `Mattress_table` table (see example table below) (Amazon Web Services, 2025).

Matteress_table		
id	**brand_name**	**cost**
1	A	499

In non-relational databases, data is usually stored in JSON documents, a group of documents together is called a collection. Data is stored in these JSON documents in key-value pairs and is accessed using these keys. An example of a MongoDB insert statement might look something like this:

```
db.transport.insert({"Brand":"Hyundai"},
{"Max_speed":100},{"Color":"blue"})
```

This inserts the data into the transport collection (see example JSON below) (GeeksforGeeks, 2022).

```
{

  "Brand": "Hyundai",

  "Max_speed": 100,

  "Color": "blue"

}
```

Choosing the right database for SETU Code Lab depends on the kind of data that needs to be stored. The system will likely store mainly structured data (users, problems, results, etc.) with the possibility of some semi-structured data likely in JSON format (submission logs, test cases).

PostgreSQL

PostgreSQL is a relational (SQL) database management system known for its reliability, performance, robust features and extensibility. With nearly forty years of active development PostgreSQL offers advanced features such as full-text search, JSON and JSONB support and custom data types (PostgreSQL, 2025).

Fig. 15. PostgreSQL Logo **Source:** (PostgreSQL, 2025)

Instead of searching for exact strings, full-text search can find results that semantically match. For example, if a student entered a search containing the word "loop", PostgreSQL can return any problems whose titles or descriptions contain the word "loop" or similar words such as "loops" or "looping".

Support for JSON and JSONB is also useful as the system may need to store some semi-structured data such as problem test cases or submission logs. Support for custom data types is also needed for SETU Code Lab as there will be many custom data types such as difficulty level (e.g. easy, medium, hard) (PostgreSQL, 2025).

PostgreSQL is also developed using test-driven development. This makes it highly stable and ensures previous bugs do not reappear in future releases (Ravoof, 2024).

MariaDB

MariaDB is an open-source, highly scalable relational (SQL) database management system developed by the original creators of MySQL. Like PostgreSQL it is fully ACID (Atomicity, Consistency, Isolation and Durability) compliant and highly compatible with MySQL.

MariaDB supports dynamic columns, a feature allowing developers to store multiple columns in a single row in a database. This is useful for storing data with many different attributes however it does not support fully custom data types like PostgreSQL (MariaDB Foundation, n.d.).

MariaDB uses thread pooling, this reduces thread cycle overhead leading to faster database queries. It also supports parallel query execution which allows multiple queries to run simultaneously without performance degradation. This further improves task execution time (Ravoof, 2024).

Fig. 16. MariaDB Logo **Source:** (MariaDB Foundation, n.d.)

Supabase

Supabase is an open-source Back End as a Service (BaaS) platform founded in 2020. It provides many features for server-side development such as authentication, auto-generated APIs, edge functions and a full PostgreSQL database (Ayezabu, 2022).

An advantage of a BaaS technology is the simplification and speeding up of back-end development and database set up. However, some applications require custom functionality that is not provided by these services (Ayezabu, 2022). In this case Supabase does not provide functionality for code sandboxing or the ability to spin up docker containers making it unsuitable for SETU Code Lab.

Fig. 17. Supabase Logo **Source:** (Supabase Inc., 2025)

MongoDB

MongoDB is a non-relational (NoSQL) database management system. It allows developers to store non-relational data such as JSON-like documents which can better reflect how the stored objects are used in code. This could prove useful in case of changing requirements. Other features listed on the MongoDB website include consistency with ACID transactions, built-in querying capabilities and serverless horizontal scaling which may assist future scaling of the application (MongoDB, n.d.). MongoDB is not suitable for SETU Code Lab as it does not support relational data.

Fig. 18. MongoDB Logo **Source:** (MongoDB, n.d.)

Firebase

Firebase was founded in 2011 and acquired by google in 2014. It is a Back End as a Service (BaaS) platform which features services like hosting, authentication, cloud functions, cloud messaging, A/B testing and two types of non-relational (NoSQL) databases.
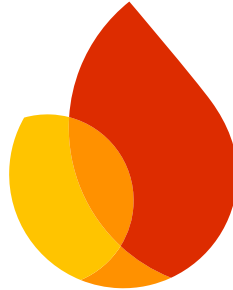
Fig. 19. FireBase Logo **Source:** (Google LLC., n.d.)

The first database is called the Firebase Realtime Database. It is a cloud-hosted database that stores JSON data and can synchronize for all users in real time. The other database service available with Firebase is the Cloud Firestore. This database is like the real time database; however, it is a document-oriented NoSQL database that stores data in collections of documents (Ayezabu, 2022). Firebase is not suitable for SETU Code Lab as it does not allow direct Docker runtime management, making code sandboxing more difficult and does not integrate well with the rest of the technology stack.

Selected Database

PostgreSQL has been selected as the database to support SETU Code Lab. This is because it can store relational data unlike MongoDB. Although MariaDB is faster for smaller relational database applications, PostgreSQL excels for larger databases and supports custom data types and semi structured data in JSON and JSONB format. PostgreSQL has also been chosen for its seamless integration with the rest of the technology stack and familiarity to the developer. Supabase and Firebase are not suitable as they do not natively support the creation of Docker containers and do not integrate well with the rest of the chosen technology stack.

Code Editors

SETU Code Lab aims to have a built-in code editor in the browser in which the user can input their solution code. This editor needs to be fast, reliable and user friendly. Several libraries have been identified for this task such as Monaco, CodeMirror and Ace. These are all available through npm and compatible with the rest of the chosen technology stack.

Monaco

Monaco is the code editor used by Visual Studio Code and thus has the same look and feel. It is written in TypeScript, and the latest version is version 0.54.0. It has many of the same features as Visual Studio Code such as IntelliSense meaning it has a rather large bundle size. IntelliSense is Visual Studio Codes signature code completion, content assist, and code hinting tool. This functionality can be enabled or disabled as needed and supports almost any programming language (Microsoft, n.d.).
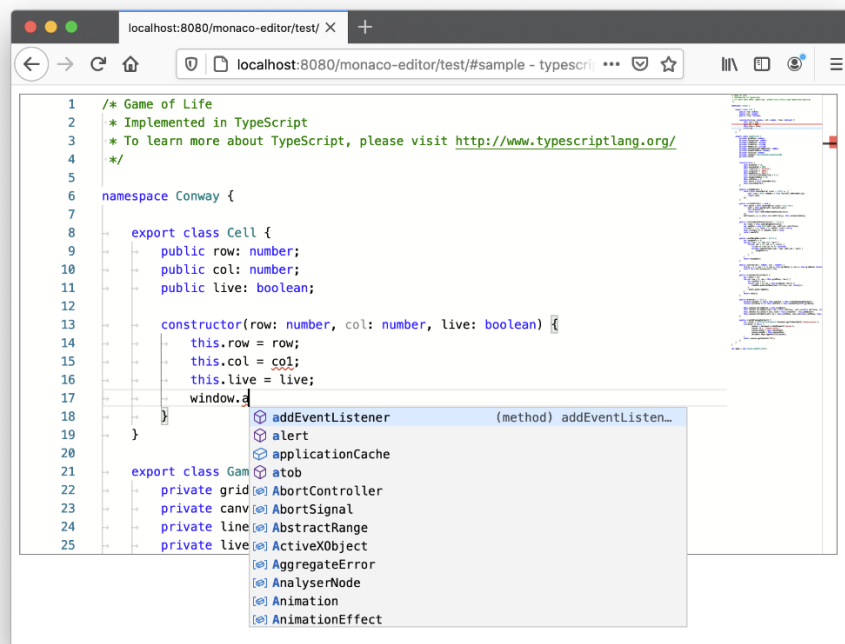
Fig. 20. Monaco editor example **Source:** (Microsoft, n.d.)

CodeMirror

CodeMirror is another feature rich code editor though it has a much smaller bundle size compared to Monaco (153kb vs 21.3kb) (NPM, 2025) (NPM, 2025). It is written in JavaScript, and the latest version is CodeMirror 6. Some of the potentially useful features include syntax highlighting which colours code to reflect its syntactic structure, language specific autocompletion hints, accessibility support for screen readers and keyboard only users and undo and redo functionality. CodeMirror also remains responsive even for very large documents (Haverbeke, n.d.).
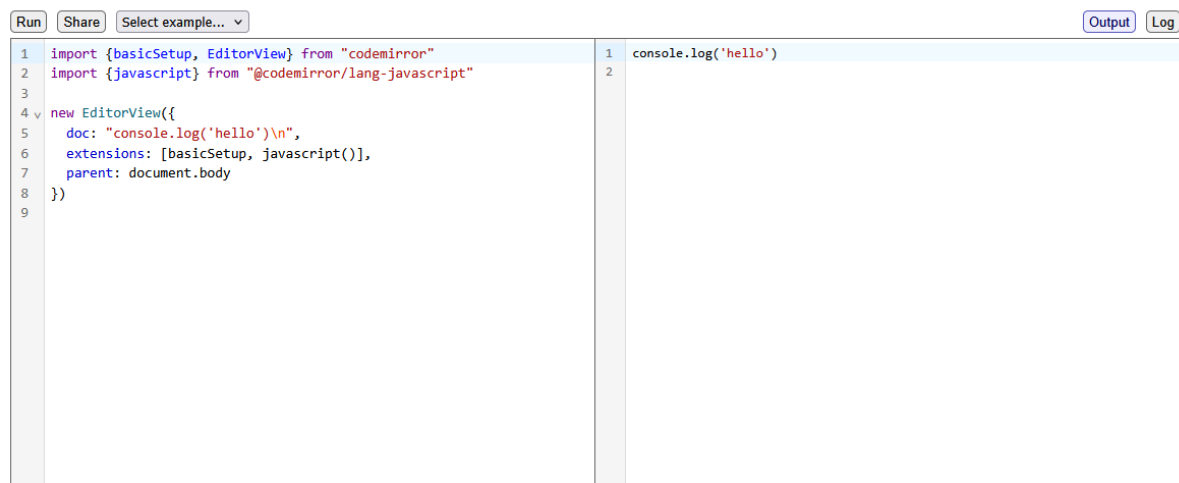
Fig. 21. CodeMirror 6 editor example **Source:** (CodeMirror, 2025)

Ace

The Ace code editor is another lightweight, embeddable code editor like CodeMirror. It has many of the same features such as syntax highlighting, code folding, themes and customizable key bindings. It also includes basic autocompletion and search and replace functionality (Ace, 2024). While Ace provides good language support and performance, its architecture is older and less modular compared to CodeMirror or Monaco. This means it can be more difficult to integrate with React (Masad, 2025).
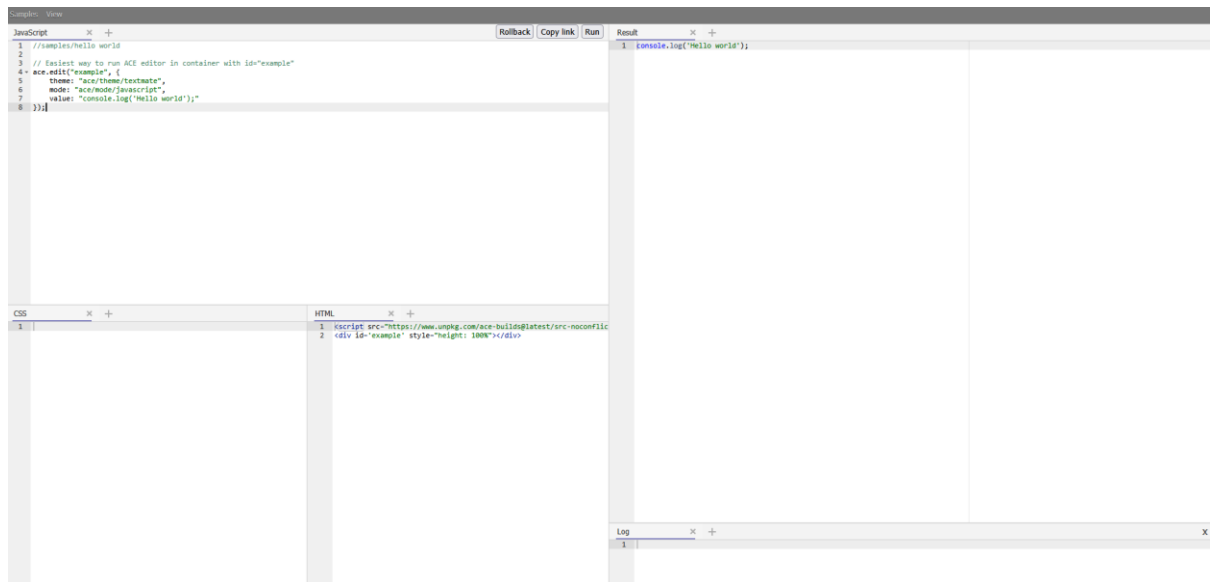
Fig. 22. Ace code editor example **Source:** (Ajax.org, 2025)

Selected Code Editor

CodeMirror 6 has been selected as the code editor for SETU Code Lab as it strikes the right balance between performance, due to its small bundle size, functionality, which is extensible if needed and ease of integration with front-end frameworks such as React.

Code Sandboxing

When a user of SETU Code Lab runs or submits a solution to a code problem, it must be run safely in a containerized environment. This is to prevent any potentially malicious code from causing harm to the system. Running code inside a docker container for instance allows developers to: limit the amount of system resources available, preventing any Denial of Service (DOS) type attacks and disable outbound networking, preventing any network abuse or exfiltration of data. Another benefit of running submitted code in a containerized environment is that every student's submission will be run under the exact same conditions ensuring fairness and reliability.

Judge0 API

Judge0 is an open source, online code execution system. It provides detailed API documentation, a highly scalable architecture, support for over sixty programming languages, detailed execution results and HTTP callbacks (Došilović, 2024). The service is free under the GPL-3.0 licence for self-hosted applications, but various paid plans are available for other hosting options (Došilović, 2025).

Judge0 works by taking in a language id (this specifies which programming language is being inputted) and the source code you wish to run via its API. This code is then run in a sandboxed environment with limited resources, and the results are returned. There is also support for multi-file programs (Došilović, 2024).

Docker

Docker is a platform for developing and running applications inside isolated environments. These isolated environments are called containers. Developers can give containers a limited amount of system resources and install the needed dependencies to allow an application, or in the case of SETU Code Lab, allow users' code submissions to run safely. This ensures that every code submission runs under the exact same conditions, restricts network access and prevents any potential security issues (Docker, inc., 2024).

Docker Engine is the technology that would be used by SETU Code Lab to create containers. When a user makes a submission, the backend communicates with the Docker Engine API to create a new container. This container is based on a predefined image that includes the required dependencies to run the submitted code. An image is like a blueprint which tells Docker how to set up the container and what needs to be installed inside of it (GeeksforGeeks, 2025). For example, the `eclipse-temurin` docker image provides a Java Development Kit (JDK) which allows Java code to execute inside a container (Eclipse Foundation, 2025).

Selected Code Sandboxing Technology

Docker has been chosen as the code sandboxing technology for SETU Code Lab. While the Judge0 API offers a quick and easy solution for code sandboxing, it lacks the precise control over resource limits that Docker provides.

 A Judge0 implementation lacks the technical depth required for a final year project and supports a limited amount of programming languages (60+), whereas a custom Docker solution could theoretically support any language and its libraries. Finally, a fully Docker based solution will allow the developer to decide exactly how submissions are run and what format the results are returned in. This can help optimize performance, maintenance and promote rapid development.

Hosting

For a website to be accessible on the internet its data needs to be stored on a server. When someone attempts to access this website, their browser contacts the server. The server responds by sending back the needed files to display the web page. It is possible to buy and set up a server however this requires considerable technical knowledge and maintenance (configuration, installing security patches, software updates etc.). It is a common practice to rent server space from third party companies. This approach is cheaper and easier to set up and minimizes the risk of downtime due to power outages or software errors (Coopersmith, 2025).

Types of Hosting

Shared Hosting is when multiple websites share a single server. Set up and security are handled for the whole server by the hosting provider. This method is well suited for small websites and is generally more affordable. This type of hosting is not suitable for SETU Code Lab as most shared hosting providers do not allow the installation of Docker. This is because it poses security risks to all users hosted on the shared server (Ecenica, 2025) .

VPS (Virtual Private Server) Hosting is where a private virtual server environment is provided inside of a shared server. This method provides more control over server set up and mitigates the risk of other sites using too much of the shared resources. VPS hosting may be a suitable option for SETU Code Lab as it is possible to install and spin up Docker containers inside a VPS. This is because unlike shared servers, a VPS gives developers root access (Machado, 2025).

With **Dedicated Hosting**, a platform has its own physical server. This provides the highest level of control over configuration, but it can be very expensive and often requires ongoing support and maintenance. This is typically more suitable for enterprise and is not feasible for SETU Code Lab (Coopersmith, 2025).

Cloud Hosting stores website data on a distributed network of servers. This provides high uptime in cases where one server may go down or there is very high traffic. Cloud hosting is often more flexible and scalable than VPS hosting but does require more set up, configuration and learning compared to VPS hosting. Cloud hosting often offers a pay-as-you-go payment model which can reduce costs if the website does not need to be online all the time but this can sometimes be complex and hard to estimate (Amazon Web Services, 2025).

DigitalOcean (VPS Solution)

A Digital Ocean Droplet is a VPS implemented through a Linux based virtual machine that is run on top of virtualized hardware. Digital Ocean is known for providing a developer friendly platform, easier set up process and provides extensive documentation and tutorials for set up. Digital Ocean also supports vertical and horizontal scaling using a load balancer and has an affordable and transparent pricing model. Droplets start from $4 per month and payments can be made by the hour or monthly (Thompson, 2025).

AWS EC2 (Cloud Solution)

One of the most popular cloud hosting platforms in the world is Amazon Web Services' (AWS) Amazon Elastic Compute Cloud (EC2). EC2 provides highly scalable, on-demand computing capacity and allows total control over security, networking and storage. EC2 instances consist of a virtual server in the cloud and are highly customizable (Amazon Web Services, 2025). They support variable custom storage sizes and many different programming languages and operating systems. Some drawbacks of AWS EC2 which may make it unsuitable for SETU Code Lab include hidden costs if not properly managed and a notably difficult set up process for new developers (Bali, 2023).

Chosen Hosting Platform

The DigitalOcean Droplet service has been selected as the hosting solution for SETU Code Lab. This is due to the following reasons:

- A simpler and more developer friendly interface and set-up process. Figure 23 shows the complex interface used for managing AWS EC2 instances. This can be difficult to work with and confusing to learn for new developers and could take up considerable project time. Compare this to the much simpler interface in figure 24, which shows the streamlined interface used for creating DigitalOcean Droplets (Bali, 2023).

- A more affordable and transparent pricing model. Droplets start from $4 per month whereas AWS EC2 costs may vary by usage. In some cases, Digital Ocean Droplets can save from 34% to 53% of hosting costs when compared to the same application hosted using AWS EC2 (DigitalOcean, 2018).

- Droplets provide enough resources and scalability for SETU Code Lab while remaining simple and easy to use. AWS EC2 offers many additional services (over 200) and features suitable for enterprise applications that are not needed for SETU Code Lab (Adegbuyi, 2025).
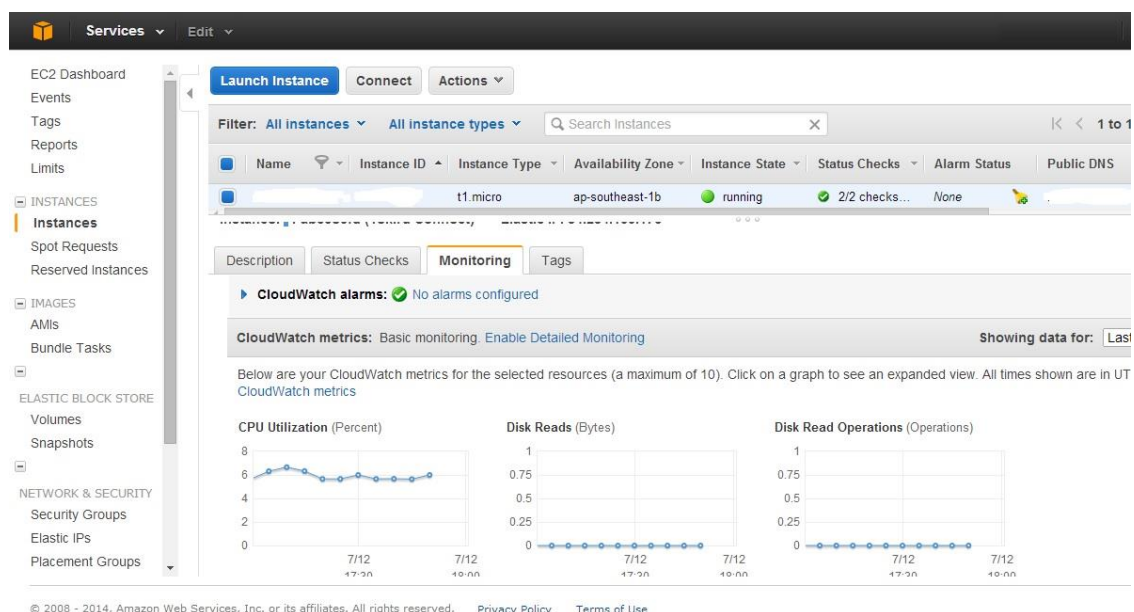
Fig. 23. AWS EC2 Dashboard **Source:** (Bali, 2023)

Fig. 24. DigitalOcean Droplet creation interface **Source:** (Bali, 2023)

Testing

When a student submits a solution to a particular code problem, the system must have some way of determining if the code is correct or not. This is where automated code testing comes in. Testing is the process of ensuring software works as intended by observing its execution and comparing actual results with expected results. Unit or component testing focuses on testing the smallest testable parts of code and can easily be automated (Dorothy, et al., 2008).

Test Cases

A test case contains a test input value and an expected output value (Dorothy, et al., 2008). A similar platform called HackerRank uses two types of test cases for testing submissions. Sample Test Cases are visible to the user and helps them to understand the problem, and Hidden Test Cases are not visible to the user and are there to prevent users from hard coding expected outputs. Hidden Test Cases encourage users to write more robust code and generalized solutions. HackerRank recommends 2-3 sample test cases to help users grasp the problem and 8-15 test cases in total to ensure a full evaluation (HackerRank, 2025).

When creating a problem, HackerRank also allows users to define skeleton functions with specific return types and parameters. This way, test inputs and test outputs will match up exactly with the parameters and return types defined in the empty function (HackerRank, 2025). Another similar platform called LeetCode also does this.
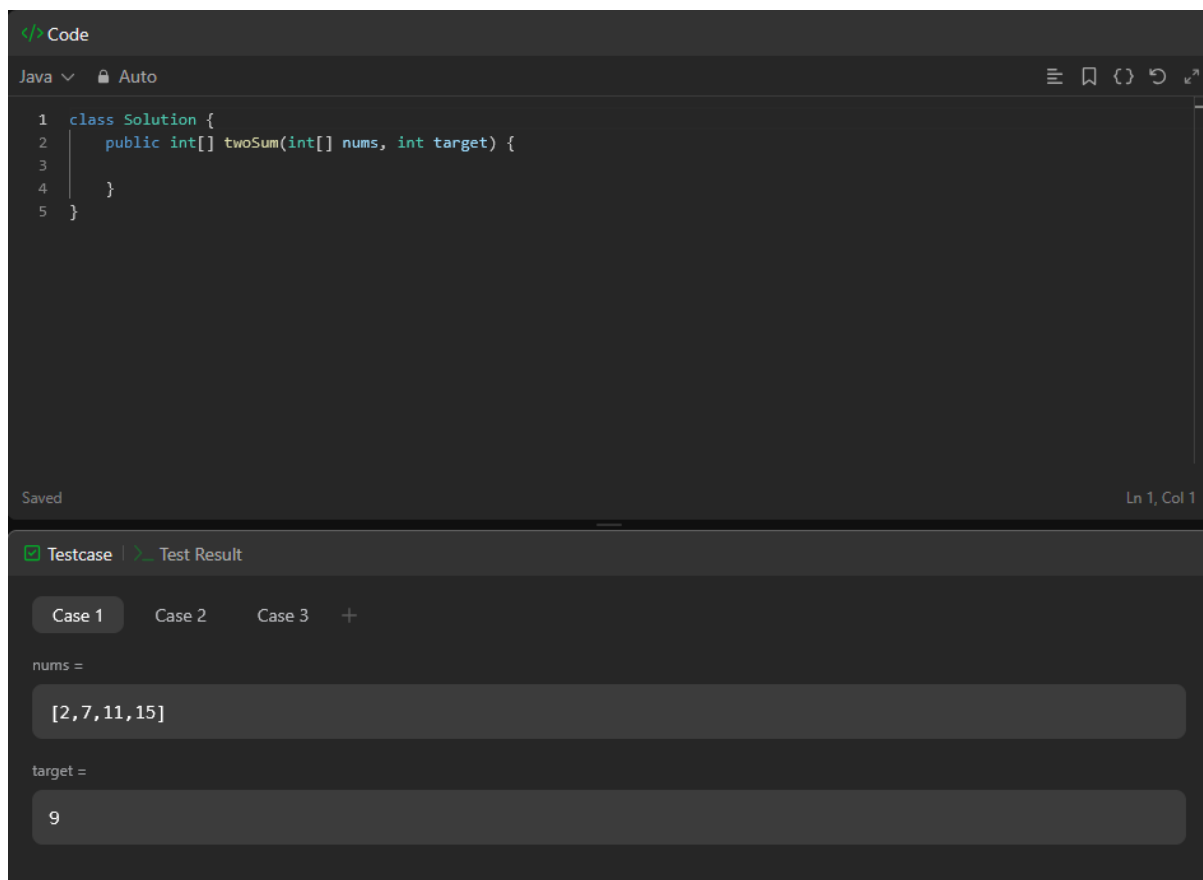

Fig. 25. Skeleton code from LeetCode **Source:** (LeetCode, 2021)

Benefits

This testing approach eliminates human error in grading which improves consistency and provides instant feedback to students improving efficiency. It does require however, that when creating new code problems and test cases, Lecturers input correctly formatted skeleton functions and test cases.

Gamification

Gamification refers to the implementation of "game-like" mechanics such as badges, points, levels and leaderboards into non-game contexts to increase user motivation and engagement (Sebastian, et al., 2011). Programming is skill that is developed through deliberate and continuous practice however, this can be difficult when students are faced with abstract or difficult concepts. Gamification can help with this by rewarding students for pushing through difficult learning curves and practicing regularly.

According to Marissa Venter in "Gamification in STEM Programming Courses: State of the Art," integrating gamification features into programming contexts can lead to positive gains in engagement, motivation and completion rate, though thoughtful design is a must. From her research, the most popular gamification elements found in computer science education are leaderboards, badges, points, levels, progress bars and avatars. She also investigates the effectiveness of these elements finding a strong positive impact overall (see below). 'N' in the table below refers to the number of studies that show the associated outcomes (Marissa, 2020).

Response Variable	Impact	N
Student engagement	Positive	6
Student Programming Knowledge	Positive	5
Student Programming Knowledge	No Impact	4
Student Motivation	Positive	3
Student Attitude	Positive	2
Student Perception About Gamification	Positive	1
Student Programming Knowledge	Negative	1
Student engagement	No Impact	1

Fig. 26. Impact of gamification in computer science education **Source:** (Marissa, 2020)

Similar Platforms

LeetCode

There are many similar platforms to SETU Code lab. One such platform is called LeetCode. This platform is designed to help software developers improve their programming skills and prepare for technical job interviews (Lupa Editorial Team, 2025). Figure 27 below shows what a sample problem and coding interface looks like. There is a problem description on the left-hand side of the screen, an in-built code editor in the top right and some sample test cases in the bottom right.
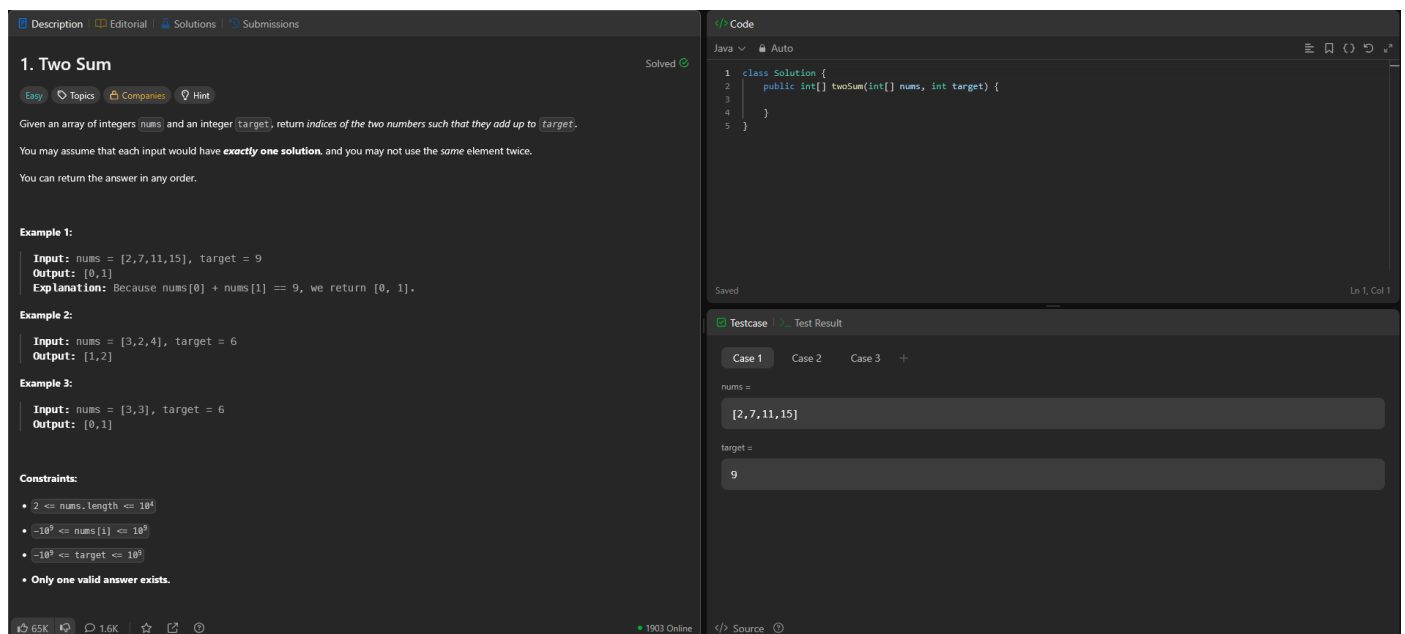

Fig. 27. Sample problem on Leetcode **Source: (**LeetCode, 2025)

Problems on LeetCode are organized into different difficulty levels (Easy, Med., Hard) and categories such as array problems, hash function problems, dynamic programming, sorting, matrices and many more (see Fig. 28.).
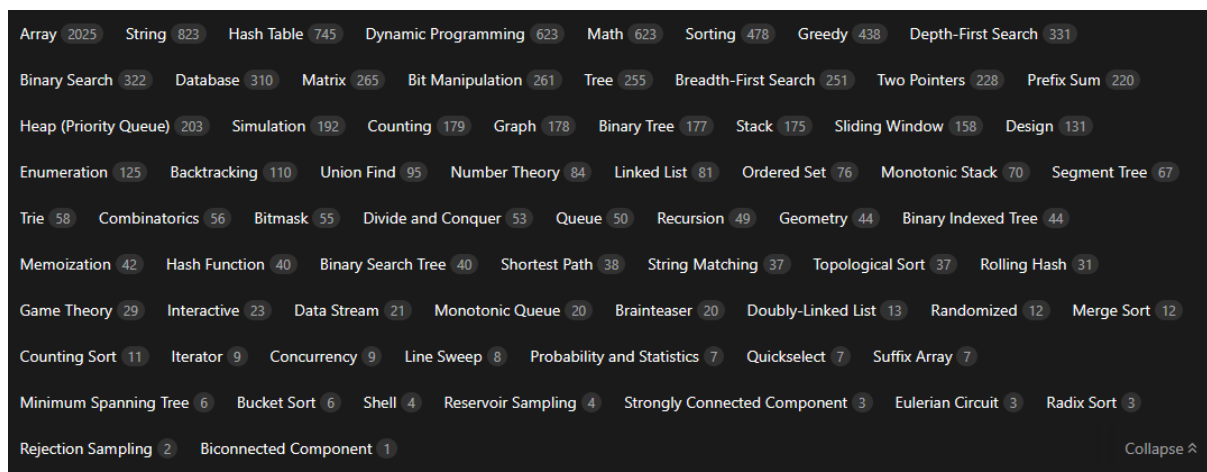

Fig. 28. Sample problem categories on LeetCode **Source:** (LeetCode, 2025)

LeetCode also supports nineteen different programming languages (see Fig. 29.) and utilizes many gamification mechanics to keep users engaged. Some of these gamification elements include points and badges earned from completing specific tasks and log-in streaks which keep track of how many days the user has logged in to LeetCode consecutively. LeetCode also holds contests and challenges, some with rewards and displays leaderboards so users can see how they rank among their peers (Lupa Editorial Team, 2025).
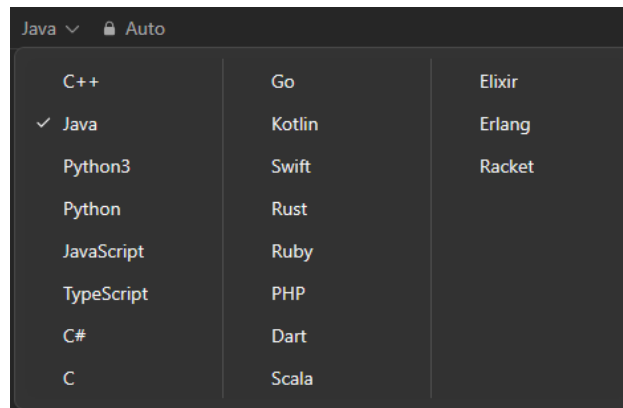
Fig. 29. Supported languages in LeetCode **Source:** (LeetCode, 2025)

HackerRank

Another similar platform to LeetCode is called HackerRank. HackerRank also focuses on allowing developers to practice coding questions for technical interviews, enter hackathons and practice take-home assignments. It even provides a paid services to allow companies to conduct live coding interviews through HackerRank and create custom problems for their internal development teams. Figure 30 below shows a sample coding problem in HackerRank. This interface is simpler than LeetCode and features a coding problem description on the left which includes sample inputs and outputs and a coding interface on the right.
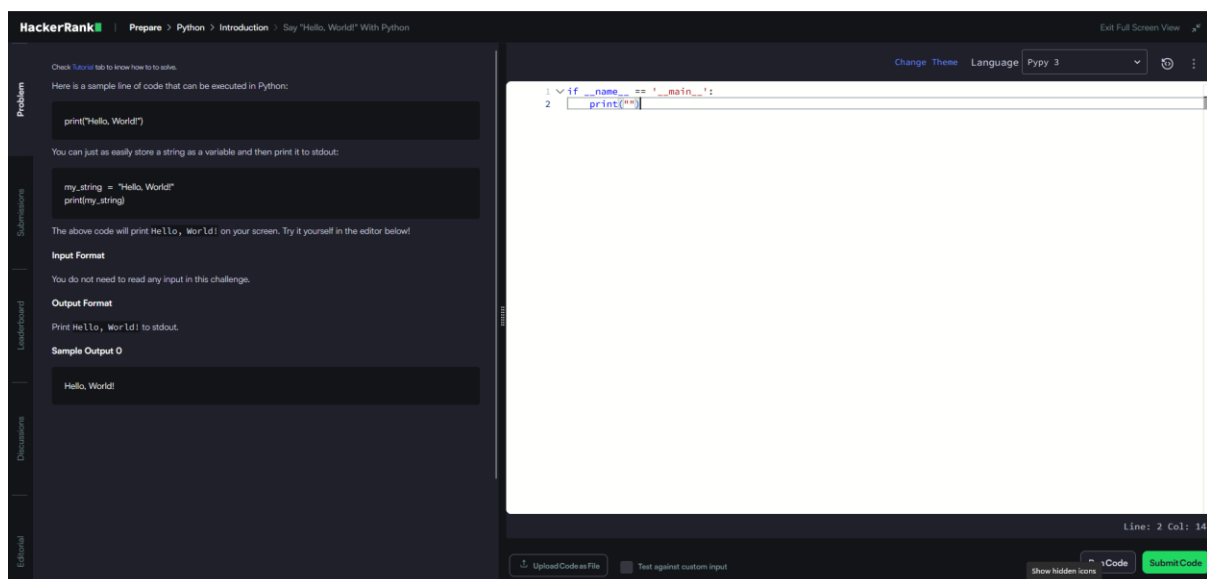
Fig. 30. Sample problem on HackerRank **Source:** (HackerRank, 2025)

Some of the gamification features present in HackerRank include points, badges, leaderboards (see Fig. 31.) and a ranked system (HackerRank, 2025).
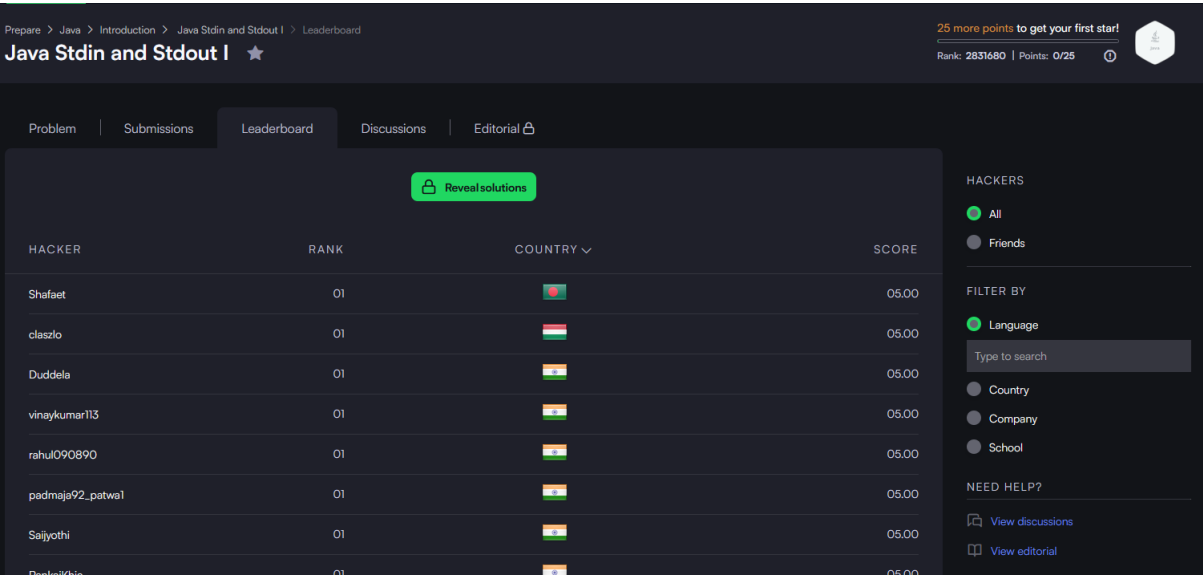
Fig. 31. Leaderboard for a particular problem in HackerRank **Source:** (HackerRank, 2025)

CodeWars

CodeWars is another similar platform to HackerRank and LeetCode. CodeWars calls its coding problems "Kata" and supports 58 different programming languages. Figure 32 below shows what a sample kata looks like on CodeWars. Similarly to LeetCode there is a description of the code problem on the left, a coding interface on the top right and sample test cases on the bottom right.
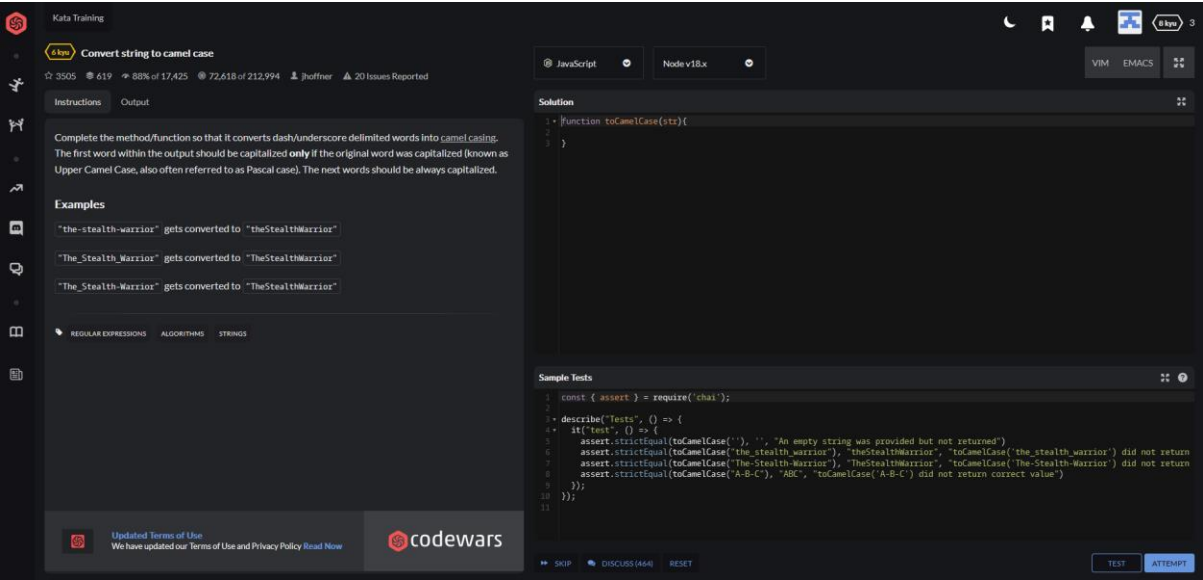
Fig. 32. Sample Kata on CodeWars **Source:** (CodeWars, 2025)

When user's complete kata, complete achievements or rank up they can earn "Honor", this is like earning experience points in a video game. CodeWars also has leaderboards ranking users by amount of honor. Users also gain more privileges on the CodeWars platform as their honor grows, for example users gain the ability to create kata upon reaching 300 honor (CodeWars, 2025).

| Required Honor Points		
Privilege	**Required Honor**	**Description**
Vote Kata	25	Vote on how satisfied you were with a kata
Mark comment as spoiler	50	Mark another's comment as a spoiler
Estimate ranking of own beta kata	75	Estimate on what rank you think your beta kata should be
Assess Rank	100	Vote on what rank you think a beta kata should be
Create Kata	300	Contribute your own kata to the community
Unmark comment as spoiler	500	Unmark another's comment as being a spoiler
Vote 2x power	1,000	Your vote counts 2x towards getting a kata out of beta
Vote 3x power	2,000	Your vote counts 3x towards getting a kata out of beta
Vote 4x power	3,000	Your vote counts 4x towards getting a kata out of beta
Coauthor Kata † / Approve Translation	4,000	Ability to edit other author's kata and approve translations
Resolve comment	5,000	Ability to resolve other's comments
Approve Kata	6,000	Ability to approve a beta kata and assign its rank
Edit locked test cases	10,000	Ability to edit test cases when they're locked

Fig. 33. CodeWars honor system **Source:** (CodeWars, 2025)

CodeWars also features a comprehensive ranked system inspired by Japanese martial arts. Ranks go from 8 Kyu to 1 Kyu and then from 1 Dan to 8 Dan, in increasing proficiency/difficulty. This rating system is used to indicate the proficiency of users, and the difficulty of Kata. The users rating is increased by completing Kata (CodeWars, 2025).
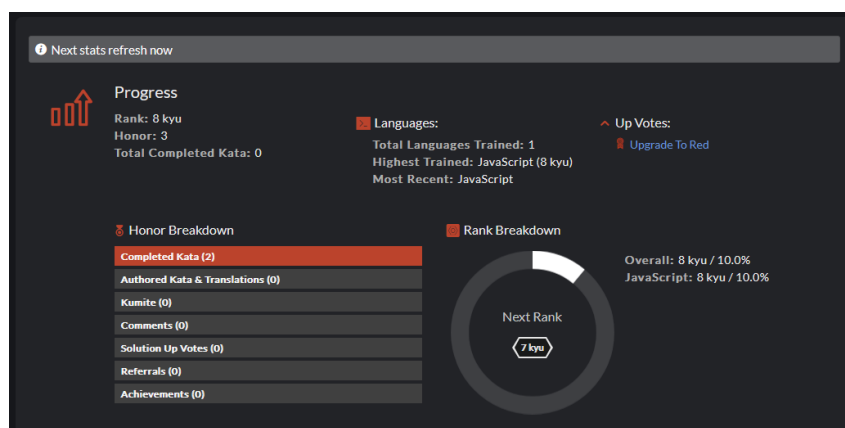

Fig. 34. Example progress breakdown on a profile on CodeWars **Source** (CodeWars, 2025)

Conclusion

In summary React, TypeScript, SCSS, and Vite have been chosen as the front-end technologies for SETU Code Lab due to their fast performance, seamless integration with one another and their strong potential for building highly interactive and responsive user interfaces. TypeScript and React have also become somewhat of an industry standard recently and there is lots of useful documentation and tutorials available for developers.

Node.js and Express have been chosen for the back end due to their seamless integration with the React and TypeScript based front-end, their high performance and flexibility. Express will help simplify tedious and repetitive back-end development tasks and database connections while also giving the developer flexibility over how features are implemented.

PostgreSQL has been chosen as the database to support SETU Code Lab because it is a relational (SQL) database with support for some semi structured data in JSON and JSONB format if needed. It also integrates easily with Node.js.

CodeMirror 6 has been chosen as the in-browser code editor for SETU Code Lab due to its small bundle size, rich functionality and easy extensibility.

The Docker Engine will be used to implement code sandboxing. This allows the system to create Docker containers with no network access and, strict resource limits. The necessary dependencies are installed into the Docker container (to run Java code for example, a JDK. is installed), the code is executed safely inside the containerized environment, the results are sent back to the system back-end, and the Docker container is destroyed to prevent a memory leak. Code sandboxing is necessary to prevent potentially malicious code from causing harm to the system or exfiltrating data.

To automate the testing and grading process SETU Code Lab will need to use test cases alongside skeleton functions to ensure the input and output types defined in the test cases match up the input and output types of student submissions. There will also need to be two types of test cases; sample test cases that the user can see to help them understand the problem; and hidden test cases that the user cannot see to prevent hard coding of solutions.

The platform will also employ some gamification mechanics to keep students motivated and engaged with the platform. SETU Code Lab will use leaderboards, points, a rating system (like CodeWars) and daily log in streaks.

Bibliography

Ace, 2024. *Ace - The High Perfromance Code Editor for the Web*. [Online]
Available at: https://ace.c9.io/
[Accessed 13 October 2025].

Adegbuyi, F., 2025. *AWS vs DigitalOcean: Which cloud platform is the best fit for you? | DigitalOcean*. [Online]
Available at: https://www.digitalocean.com/blog/aws-vs-digitalocean-cloud-platform
[Accessed 8 November 2025].

Ajax.org, 2025. *Ace Playground*. [Online]
Available at: https://mkslanc.github.io/ace-playground/#hello-world
[Accessed 18 October 2025].

Amazon Web Services, 2025. *What is Amazon EC2? - Amazon Elastic Compute Cloud*. [Online]
Available at: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html
[Accessed 8 November 2025].

Amazon Web Services, 2025. *What is Cloud Hosting? - Cloud Server Hosting Explained - AWS*. [Online]
Available at: https://aws.amazon.com/what-is/cloud-hosting/
[Accessed 8 November 2025].

Amazon Web Services, 2025. *What is SQL? - Structured Query Language (SQL) Explained - AWS*. [Online]
Available at: https://aws.amazon.com/what-is/sql/
[Accessed 8 November 2025].

Ayezabu, A., 2022. *Supabase vs Firebase: Evaluation of Performance and Development of Progressive Web Apps*. 1st ed. Helsinki: Metropolia University of Applied Sciences.

Bali, K., 2023. *DigitalOcean vs AWS EC2: 8 Factors to Decide Who is the Winner?*. [Online]
Available at: https://serverguy.com/digitalocean-vs-aws-ec2/
[Accessed 8 November 2025].

CodeMirror, 2025. *Try CodeMirror*. [Online]
Available at: https://codemirror.net/try/
[Accessed 18 October 2025].

CodeWars, 2025. *Privileges | The CodeWars Docs*. [Online]
Available at: https://docs.codewars.com/gamification/privileges/
[Accessed 22 October 2025].

CodeWars, 2025. *Ranks | The CodeWars Docs*. [Online]
Available at: https://docs.codewars.com/gamification/ranks/
[Accessed 22 October 2025].

Coopersmith, A., 2025. *What is Web Hosting: Learn How a Website Gets Online. WordPress.com*. [Online]
Available at: https://wordpress.com/blog/2025/04/28/what-is-web-hosting/
[Accessed 7 November 2025].

DigitalOcean, 2018. *Lowering AWS Costs with DigitalOcean: White Paper,* New York: DigitalOcean.

Django Software Foundation, 2025. *Middleare | Django Documentation | Django*. [Online]
Available at: https://docs.djangoproject.com/en/3.0/topics/http/middleware/
[Accessed 15 November 2025].

Django Software Foundation, 2025. *Templates | Django Docs*. [Online]
Available at: https://docs.djangoproject.com/en/5.2/topics/templates/
[Accessed 15 October 2025].

Django Software Foundation, 2025. *The Django template language | Django documentation | Django*. [Online]
Available at: https://docs.djangoproject.com/en/5.2/ref/templates/language/
[Accessed 10 November 2025].

Docker, inc., 2024. *What is Docker? | Docker Docs*. [Online]
Available at: https://docs.docker.com/get-started/docker-overview/
[Accessed 17 October 2025].

Dorothy, G., Erik, v. V., Isabel, E. & Rex, B., 2008. *Foundations of Software Testing*. 1st ed. London: Cengage Learning EMEA.

Došilović, H. Z., 2024. *Judge0 CE - API Docs*. [Online]
Available at: https://ce.judge0.com/
[Accessed 17 October 2025].

Došilović, H. Z., 2025. *Judge0 - Pricing*. [Online]
Available at: https://judge0.com/#pricing
[Accessed 17 October 2025].

Ecenica, 2025. *Can I Install Docker on Ecenica Shared-Hosting? - Ecenica*. [Online]
Available at: https://www.ecenica.com/support/answer/can-i-install-docker-on-ecenica-shared-hosting/
[Accessed 7 November 2025].

Eclipse Foundation, 2025. *Eclipse Temurin Docker Image*. [Online]
Available at: https://hub.docker.com/_/eclipse-temurin
[Accessed 17 October 2025].

GeeksforGeeks, 2022. *Querying in NoSQL - GeeksforGeeks*. [Online]
Available at: https://www.geeksforgeeks.org/dbms/querying-in-nosql/
[Accessed 8 November 2025].

GeeksforGeeks, 2025. *Difference Between React.js and Angular.js*. [Online]
Available at: https://www.geeksforgeeks.org/reactjs/difference-between-react-js-and-angular-js/
[Accessed 8 10 2025].

GeeksforGeeks, 2025. *Django Admin Interface - GeeksforGeeks*. [Online]
Available at: https://www.geeksforgeeks.org/python/python-django-admin-interface/
[Accessed 10 November 2025].

GeeksforGeeks, 2025. *Flask vs Django*. [Online]
Available at: https://www.geeksforgeeks.org/python/differences-between-django-vs-flask/
[Accessed 15 October 2025].

GeeksforGeeks, 2025. *History and Evolution of React*. [Online]
Available at: https://www.geeksforgeeks.org/reactjs/history-and-evolution-of-react/
[Accessed 8 October 2025].

GeeksforGeeks, 2025. *Install Fastapi And Run Your First Fastapi Server On Windows - GeeksforGeeks*. [Online]
Available at: https://www.geeksforgeeks.org/python/install-fastapi-and-run-your-first-fastapi-server-on-windows/
[Accessed 10 November 2025].

GeeksforGeeks, 2025. *Middleware in Express - GeeksforGeeks*. [Online]
Available at: https://www.geeksforgeeks.org/node-js/middleware-in-express-js/
[Accessed 15 November 2025].

GeeksforGeeks, 2025. *ReactJS Unidirectional Data Flow*. [Online]
Available at: https://www.geeksforgeeks.org/reactjs/reactjs-unidirectional-data-flow/
[Accessed 10 November 2025].

GeeksforGeeks, 2025. *Vue.js Tutorial - GeeksforGeeks*. [Online]
Available at: https://www.geeksforgeeks.org/javascript/vue-js/
[Accessed 8 October 2025].

GeeksforGeeks, 2025. *What is Docker Engine?*. [Online]
Available at: https://www.geeksforgeeks.org/devops/what-is-docker-engine/
[Accessed 17 October 2025].

Google, 2025. *What is Angular?*. [Online]
Available at: https://angular.dev/overview
[Accessed 8 10 2025].

HackerRank, 2025. *HackerRank Coding Interviews - Code Live with Candidates*. [Online]
Available at: https://www.hackerrank.com/products/interview
[Accessed 21 October 2025].

HackerRank, 2025. *Test Cases in Coding Questions*. [Online]
Available at: https://support.hackerrank.com/articles/3245197419-test-cases-in-coding-questions
[Accessed 18 October 2025].

HackerRank, 2025. *Test Cases in Coding Questions*. [Online]
Available at: https://support.hackerrank.com/articles/3245197419-test-cases-in-coding-questions
[Accessed 18 October 2025].

Haverbeke, M., n.d. *CodeMirror Documentation*. [Online]
Available at: https://codemirror.net/
[Accessed 13 October 2025].

Laravel, 2025. *Blade Templates - Laravel 12.x - The PHP Framework For Web Artisans*. [Online]
Available at: https://laravel.com/docs/12.x/blade
[Accessed 10 November 2025].

Laravel, 2025. *Blade Templates | Laravel Docs*. [Online]
Available at: https://laravel.com/docs/12.x/blade
[Accessed 15 October 2025].

Laravel, 2025. *Laravel Docs*. [Online]
Available at: https://laravel.com/docs/12.x
[Accessed 15 October 2025].

Laravel, 2025. *Routing - Laravel 12.x - The PHP Framework For Web Artisans*. [Online]
Available at: https://laravel.com/docs/12.x/routing
[Accessed 10 November 2025].

LeetCode, 2021. *Two Sum - LeetCode*. [Online]
Available at: https://leetcode.com/problems/two-sum/description/
[Accessed 18 10 2025].

Lupa Editorial Team, 2025. *What is LeetCode: Beginner's Guide for Coding Interview Preparation*. [Online]
Available at: https://www.lupahire.com/blog/what-is-leetcode
[Accessed 20 October 2025].

Machado, R., 2025. *Setting Up a VPS with Docker: A Step-by-Step Guide - DEV Community*. [Online]
Available at: https://dev.to/ryangst/setting-up-a-vps-with-docker-a-step-by-step-guide-55hb
[Accessed 7 November 2025].

MariaDB Foundation, n.d. *Dynamic Columns | MariaDB Documentation*. [Online]
Available at: https://mariadb.com/docs/server/reference/sql-structure/nosql/dynamic-columns#dynamic-columns-basics
[Accessed 16 October 2025].

Marissa, V., 2020. *Gamification in STEM programming courses: State of the art*. Porto, Portugal, IEEE.

Masad, F., 2025. *Replit - Comparing Code Editors*. [Online]
Available at: https://blog.replit.com/code-editors
[Accessed 13 October 2025].

Meta Platforms, Inc., 2025. *Quick Start - React*. [Online]
Available at: https://react.dev/learn
[Accessed 8 October 2025].

Microsoft, 2024. *Blazor Docs*. [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website
[Accessed 15 October 2025].

Microsoft, 2025. *TypeScript Documentation*. [Online]
Available at: https://www.typescriptlang.org/docs/
[Accessed 9 October 2025].

Microsoft, 2025. *TypeScript Documentation - JSX*. [Online]
Available at: https://www.typescriptlang.org/docs/handbook/jsx.html
[Accessed 8 October 2025].

Microsoft, n.d. *IntelliSense in Visual Studio Code*. [Online]
Available at: https://code.visualstudio.com/docs/editing/intellisense
[Accessed 12 October 2025].

Microsoft, n.d. *microsoft/monaco-editor: A browser based code editor*. [Online]
Available at: https://github.com/microsoft/monaco-editor
[Accessed 18 October 2025].

MobX, 2025. *MobX Documentation*. [Online]
Available at: https://mobx.js.org/README.html
[Accessed 8 October 2025].

MongoDB, n.d. *MongoDB Documentation | Homepage*. [Online]
Available at: https://www.mongodb.com/docs/
[Accessed 15 October 2025].

MongoDB, n.d. *NoSQL VS SQL Databases | MongoDB*. [Online]
Available at: https://www.mongodb.com/resources/basics/databases/nosql-explained/nosql-vs-sql
[Accessed 15 October 2025].

NPM, 2025. *Npm - CodeMirror V6*. [Online]
Available at: https://www.npmjs.com/package/codemirror
[Accessed 13 October 2025].

NPM, 2025. *Npm - Monaco Editor React*. [Online]
Available at: https://www.npmjs.com/package/@monaco-editor/react
[Accessed 13 October 2025].

OpenJS Foundation, 2025. *Express - Node.js Web Application Framework*. [Online]
Available at: https://expressjs.com/
[Accessed 26 October 2025].

OpenJS Foundation, 2025. *Express "Hello World" Example*. [Online]
Available at: https://expressjs.com/en/starter/hello-world.html
[Accessed 15 November 2025].

OpenJS Foundation, n.d. *Node.js - Introduction to Node.js*. [Online]
Available at: https://nodejs.org/en/learn/getting-started/introduction-to-nodejs
[Accessed 11 10 2025].

Pallets Projects, 2025. *Flask Docs*. [Online]
Available at: https://flask.palletsprojects.com/en/stable/design/#the-routing-system
[Accessed 15 October 2025].

PostgreSQL, 2025. *PostgreSQL: Documentation*. [Online]
Available at: https://www.postgresql.org/docs/current/
[Accessed 11 October 2025].

Ramírez, S., n.d. *Features | FastAPI*. [Online]
Available at: https://fastapi.tiangolo.com/features/
[Accessed 15 October 2025].

Ravoof, S., 2024. *MariaDB vs PostgreSQL: 14 Critical Differences.* [Online]
Available at: https://kinsta.com/blog/mariadb-vs-postgresql/#mariadb-vs-postgresql-headtohead-comparison
[Accessed 9 November 2025].

SASS Team, 2025. *SASS Documentation*. [Online]
Available at: https://sass-lang.com/documentation/
[Accessed 9 October 2025].

Sebastian, D., Dan, D., Lennart, N. & Rilla, K., 2011. *From Game Design Elements to Gamefulness: Defining Gamification*. Tampere, Finland, Association for Computing Machinery (ACM).

Statcounter, 2025. *Desktop Screen Resolution Stats Europe*. [Online]
Available at: https://gs.statcounter.com/screen-resolution-stats/desktop/europe
[Accessed 4 October 2025].

Tailwind Labs, 2025. *Getting Started with Tailwind CSS*. [Online]
Available at: https://tailwindcss.com/docs/installation/using-vite
[Accessed 10 October 2025].

Thompson, M., 2025. *What is a Droplet in DigitalOcean? | Disruptiv-e*. [Online]
Available at: https://www.disruptiv-e.com/article/what-is-a-droplet-in-digitalocean
[Accessed 8 November 2025].

Vite, 2024. *Getting Started | Vite*. [Online]
Available at: https://vite.dev/guide/
[Accessed 8 October 2025].

Vue.js, 2025. *Introduction | Vue.js*. [Online]
Available at: https://vuejs.org/guide/introduction.html
[Accessed 10 November 2025].

W3Schools, n.d. *Node.js Express.js*. [Online]
Available at: https://www.w3schools.com/nodejs/nodejs_express.asp
[Accessed 16 October 2025].

Walia, A. S., 2025. *Build a Flask Web App from Scratch | DigitalOcean*. [Online]
Available at: https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3
[Accessed 10 November 2025].

Wikimedia, 2025. *File:React-icon.svg - Wikimedia Commons*. [Online]
Available at: https://commons.wikimedia.org/wiki/File:React-icon.svg
[Accessed 26 October 2025].