

32386120 | Alec Ng
32136111 | Justin Lane
45909116 | Aviral Garg
31580129 | Stephen Hu
90294133 | Yue (Mark) Chen
15991152 | Anh Duc (Andrew) Bui
54952130 | Tushar Kalra

CodePal

Motivation/Opportunity

CodePal will be a web based application that will provide a workspace for developers to quickly compile and execute small pieces of code, and search for online resources without having to manage several traditional heavyweight applications. This application will address the issue of developers needing to launch a full fledged IDE and keep track of several browsers and tabs to complete tasks such as interview prep or small homework assignments which do not require all of that functionality. CodePal differs from any other web application that is currently available as one of its core functionalities is to integrate resource search through web apps such as YouTube and StackOverflow to increase efficiency by eliminating the need to multitask between different browser tabs and windows.

Problem Statement

The problem of	Managing multiple applications whose full functionality is not required to develop and test small pieces of code
affects	Software developers
the impact of which is	Having a disorganized workspace with unnecessary memory/CPU usage
a successful solution would be	Creating a singular space to present and manage all resources, while keeping functionalities to the bare minimum

Product Position Statement

For	Software developers of any skill level
Who	Want to execute and test small pieces of code while having online resources to help them debug and learn
Our System	Is a web based application
That	Provides a singular workspace incorporating a lightweight code editor and UI modules to search and present online resources

Unlike	Other services that do not provide integration for resource searching
Our Product	Requires no installation and exists as a free to use web application

User Demographics

Users of CodePal fall into the following categories: Developers, and Generic End Users.

Software Developers

Developers will be experienced users of various software development styles that can integrate CodePal into their smaller projects. Their objective is to have singular space to present and manage all resources.

Generic End Users

End users will have a wide variety of skillsets ranging from complete beginners to expert users of software development suites. Their primary goal is to have an easy-to-use UI that is a singular space to present and manage all resources.

Process Model

We have chosen the Agile model for our project based on the following reasons:

- CodePal is essentially a customizable slate which includes several independent modules (code editor, collaborative chat, etc.) It makes sense to develop a working system early (which is the responsive blank slate including login mechanism, generic UI design, etc) get feedback and accordingly accommodate suggested changes at a very little cost because of the frequency of the new increments.
- Our core modules are known but the requirements for each are unknown at the moment. For example, for the integrated code editor, it is unsure which languages to provide support for, or integration for language-specific features (i.e. Python and packages). Agile addresses these issues. Creation and accommodation of such requirements can be discussed and features can be added or removed based on end-user feedback. This effectively gives us the required finished system with great flexibility and freedom for developers.

Feature List

- An embedded code editor with support for various programming languages and real-time compilation and execution with the ability to save and share code snippets
- User settings to change the layout/appearance of the CodePal page
- An embedded Youtube player with search function
- A pane which supports searching/viewing StackOverflow posts
- Login to access the application via Facebook
- Ability to live chat with other users
- Real time recommendations of resou
- rces relevant to the user's current activity

Constraints

- CodePal should run in any OS with internet access in the latest version of Chrome
 - Specific target is Chrome Version 53 and beyond
 - Additional modern web browsers such as Firefox, Edge, and Safari should be compatible with our product, but will not be specifically targeted
 - Minimum desktop resolution targeted: 1024 x 768 viewport and beyond
 - There will be no specific UI optimizations made for mobile or tablet interface, as that is not the target usage
- Facebook Login will be used for identity and authentication
 - This service is provided free of charge
- AWS will be used for hosting the backend infrastructure
 - Pricing is determined by usage and tier
 - We can take advantage of the Free AWS Student Pack with \$140 credits, which should be enough to meet our needs
- Dependencies on YouTube/StackExchange API as well as the Repl.It API, which will provide functionality to compile and run code
- Dependencies on Ace code editor and Golden Layout script libraries for styling

Scope and Limitations

- An embedded code editor with support for various programming languages
 - This is the core component of our product that should be integrated
 - The editor will have configurable settings such as syntax highlighting for different languages, multiple themes, as well options such as tab spaces, etc
 - This will be a lightweight version of a traditional IDE that will serve its purpose for small, singular file pieces of code that do not require the full functionality of an IDE
- An embedded YouTube player
 - This is a major selling point of our product, which is to enable a streamlined environment for development and simultaneous video playback
 - This should also not be too difficult to implement, as the YouTube API is well-documented and well-supported
- A pane to display StackOverflow resources
 - This is another major selling point of our product, which is to recommend useful, relevant resources to the user's search query
 - This should also not be too difficult to implement, as the StackExchange API is well-documented and well-supported

The following features will not be implemented due to time constraints:

- Ability to share code snippets
- Real time recommendations of resources relevant to the user's current activity

Assumptions and Dependencies

Upstream Dependencies:

- Embedded Ace code editor: <https://ace.c9.io>
 - The embedded code editor that will be core component of our product
- Repl.it API: <https://repl.it/site/api>
 - Functionality to compile and run code
- YouTube: <https://developers.google.com/youtube/>
 - The embedded video player enabling users to follow along video lessons
- StackOverflow: <https://api.stackexchange.com/>
 - Our recommendation engine provides useful list of questions and their answers discussed on StackOverflow relating to the user's search query
- AWS: <https://aws.amazon.com>
 - The cloud infrastructure that will host our web interface and databases
- Facebook Login: <https://developers.facebook.com/products/login>
 - Login API used for authentication and identity
- Golden Layout: <https://www.golden-layout.com/>
 - Multi-screen layout manager

Non-Functional Requirements

Performance Requirements

Web Interface

- A fluid UI for minimizing and maximizing individual windows within the page that should function smoothly without any appreciable slowdown during UI interactions, <1s lag time
- A fast and efficient compiler that should compile a standard 200 line Java program in under 5 seconds. Maximum time before compilation/running is finished is 5 seconds
- YouTube videos should play smoothly without interruption within the interface
- Downloading code snippets should be quick, files being <=300kb
- A program should be able to be compiled/run while viewing a YouTube video
- The StackOverflow module, YouTube player, and code editor should function concurrently without slowdown

Platform

- All modern web browsers should work, however only Chrome 53+ is specifically supported

API

- The API should be able to handle concurrent users, up to 10 requests per second
- The API response time should be under 10ms
- The uptime of the API should be 99%, meaning at most 1 hour 40 minutes of downtime per week, or 14 min 24 seconds per day

Security Requirements

- As login is abstracted through our integration with Facebook Login, user credentials will be handled through their side, without our need to store and manage user data.

- If we are storing shared code snippets, or archiving chat logs, we will need to ensure we properly encrypt data at rest with the industry-standard SHA-2 encryption scheme.
- We will also ensure we follow proper coding guidelines and avoid leaking private keys unencrypted in our configs setting or hard-coded in any file. We will either use a dedicated secrets management service, or store them securely in the host's environment variables.

Software Quality Attributes

- We will design an interactive UI/UX so that users can use the product more effectively.
- The application can be easily accessed by Chrome browser on most computers.
- The source code will be managed and documented appropriately for software maintenance and new feature implementations.

Use Cases

Use Case 1: Generic Use of Online Code Editor:

Identification: primary function of CodePal, user will launch the webpage and have access to a code editor.

Primary Actor: Generic User

Stakeholders and Interests: None

Preconditions: User has access to a web browser and an internet connection

Post Conditions: User has successfully opened a working development suite

Main Success Scenario:

1. User opens web page
2. User logs in through Facebook
3. Presented with CodePal main UI
4. Can immediately start writing code in a dedicated window

Extensions and Alternative Flows:

- The user's desired language is not supported. There is no alternate flow, except maybe presenting an option that opens up a public poll of desired languages to be integrated

Open Issues: N/A

Use Case 2: Saving code snippets

Identification: users will be able to save code snippets to their account for future use

Primary Actor: Developer

Stakeholders and Interests: None

Preconditions: User has some code written in the CodePal main code editor window

Post Conditions: tab is named with the user's chosen file name and the file is stored in the database

Main Success Scenario

1. User types in some sort of code in the code editor
2. User presses "save", enters a unique filename and presses "save" again
3. Tab changes to chosen name, and saved file appears in saved snippets window

Extensions and Alternative Flows:

- File name is already taken - user is prompted to choose another filename

Open Issues: May need to cap how many snippets a user is allowed to store, or introduce some sort of filtering if the list gets too large

Use Case 3: Restoring code snippet

Identification: User will be able to restore a previously saved code snippet to their current code editor session

Primary Actor: Developer

Stakeholders and Interests: None

Preconditions: User has saved some code snippet per use case 2

Post Conditions: The code editor opens a new tab that corresponds to the chosen file's name and populates the editor with the saved contents

Main Success Scenario

1. User chooses a saved file from the saved snippets window
2. A new tab is created in the code editor with the chosen filename, file contents and language

Extensions and Alternative Flows: N/A

Open Issues: N/A

Use Case 4: YouTube Video View

Identification: users will be able to search for and view YouTube videos through a window embedded within CodePal

Primary Actor: Generic User

Stakeholders and Interests: None

Preconditions: Codepal page is opened, user is logged in

Postconditions: YouTube window within CodePal shows the desired video (generic embedded YouTube video)

Main Success Scenario:

1. User will search for a YouTube video using the CodePal UI,
2. be presented with the results,
3. and be able to select/watch the video

Extensions and Alternative Flows: N/A

Open Issues: N/A

Use Case 5: StackOverflow view

Identification: users will be able to view posts on StackOverflow to help them solve problems or learn new practices.

Primary Actor: Developer

Stakeholders and Interests: None

Preconditions: User is signed in to CodePal

Postconditions: StackOverflow module in CodePal will show a list of related questions

Main Success Scenario:

1. User searches StackOverflow using the CodePal UI,
2. and is presented with relevant post results
3. User clicks on a question
4. And the view of the answer is toggled down

Extensions and Alternative Flows: N/A

Open Issues: Currently implementing a functionality that every time the user compiles on code editor and receives errors. These error messages will be automatically looked up in StackOverflow and CodePal will return a list of questions that are related to the errors.