TaskVibe Tech Stack Study Guide

TaskVibe Documentation Team

May 2025

Contents

1 Introduction

TaskVibe is a Haskell-based daily routine and scheduling web application developed in GitHub Codespaces, with a frontend hosted on GitHub Pages and documentation generated using Sphinx. This study guide provides an overview of the technologies in TaskVibe's tech stack, their roles, setup instructions, and resources for learning. It is designed for developers contributing to or learning from the project.

2 Haskell

2.1 Role in TaskVibe

Haskell is the primary programming language for TaskVibe's backend, powering the API and business logic using the Scotty web framework. Its functional programming paradigm ensures robust, type-safe code for task and schedule management.

2.2 Setup in Codespaces

Install Haskell and Stack in GitHub Codespaces:

```
sudo apt update
sudo apt install -y haskell-platform
curl -sSL https://get.haskellstack.org/ | sh
```

Initialize a new Haskell project:

```
stack new TaskVibe simple
cd TaskVibe
stack setup
```

2.3 Learning Resources

- Official Haskell Documentation
- Learn You a Haskell for Great Good!
- Real World Haskell

3 Stack

3.1 Role in TaskVibe

Stack is the build tool for TaskVibe’s Haskell backend, managing dependencies and builds. It uses ‘stack.yaml‘ and ‘TaskVibe.cabal‘ to configure the project.

3.2 Setup

Stack is installed with Haskell (see above). Configure ‘stack.yaml‘:

```
resolver: lts-22.0
packages:
- .
extra-deps:
- scotty-0.12
```

Build the project:

```
stack build
```

3.3 Learning Resources

- Stack Documentation
- FP Complete’s Stack Guide

4 Scotty

4.1 Role in TaskVibe

Scotty is a lightweight Haskell web framework used for TaskVibe’s API, handling routes for tasks and schedules (e.g., `GET /tasks`, `POST /schedules`).

4.2 Setup

Add Scotty to ‘TaskVibe.cabal‘:

```
build-depends: base, scotty, aeson
```

Example route in `Main.hs`:

```
module Main where
import Web.Scotty
main = scotty 3000 $ do
  get ”/tasks” $ text ”List␣of␣tasks”
```

4.3 Learning Resources

- Scotty Hackage
- Scotty GitHub

5 Tailwind CSS

5.1 Role in TaskVibe

Tailwind CSS styles TaskVibe’s frontend, providing a responsive, utility-first design for the GitHub Pages-hosted UI.

5.2 Setup

Install Tailwind CSS:

```
npm install -D tailwindcss
npx tailwindcss init
```

Configure `tailwind.config.js` and generate `frontend/css/styles.css`.

5.3 Learning Resources

- Tailwind CSS Documentation
- Tailwind UI

6 Sphinx and MyST-Parser

6.1 Role in TaskVibe

Sphinx generates TaskVibe’s documentation, hosted on GitHub Pages. MyST-Parser enables Markdown support, integrating `README.md` and other Markdown files.

6.2 Setup

Install Sphinx and MyST-Parser: