



# Molan IDE

An Online Development Platform

# Contributors



**Progyan  
Bhattacharya**

*Frontend Supervisor  
Web Application Developer*

*progyanb@acm.org  
<https://codeprogyan.me>*



**Prakash Kumar**

*Backend Supervisor*

*prakash.chy.10@gmail.com*



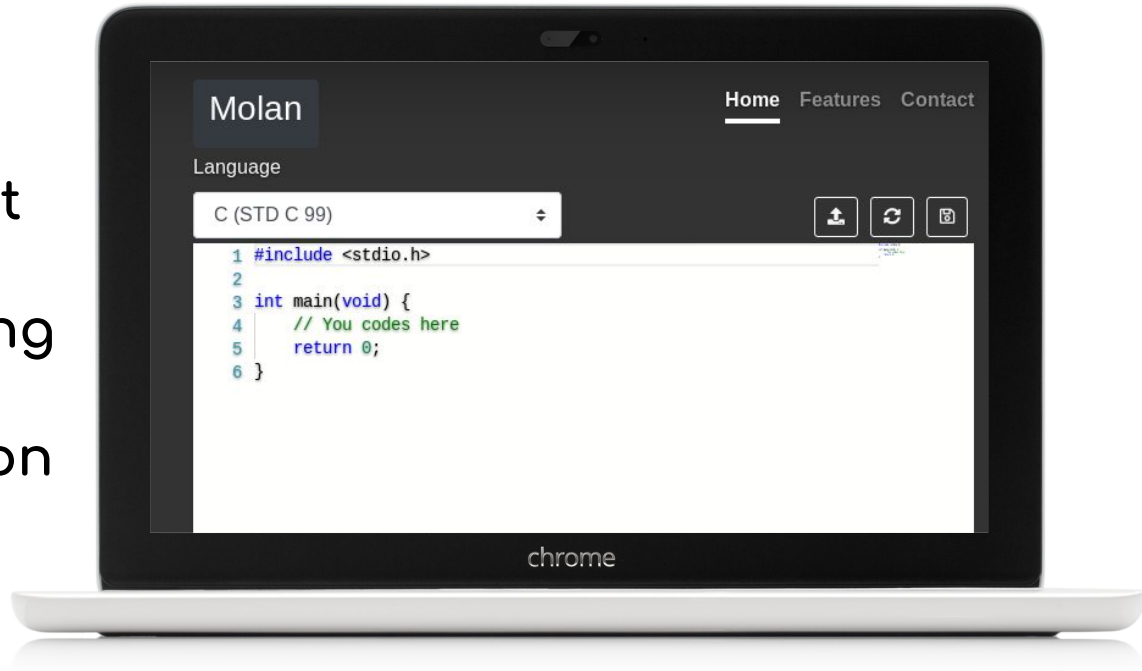
**Deepak Thakur**

*Project Coordinator  
Designer*

*dt19370@gmail.com*

# Product Overview

Molan is a web based Integrated Development Environment with a flexible text editor having multiple language support and deployed on a linux system.



## Premise of Development

There are already a handful number of online IDE available in the market. Ideone(by SPOJ), Runkit(by Stack Exchange), jsFiddle just to name a few. But all of them have embedded a text editor on a static HTML causing the web application to be stagnant and less efficient with user interaction, very much limited compared to desktop equivalent. What we target to build here is a scalable and efficient web IDE using MVC architecture.

# Features

- Interactive Text Editor
- Support for different Programming languages.
- Remote and quick access of data from local storage.
- Offers both user machine and server database Cache
- Minimum boilerplate code as starter.
- Quick reset options.
- Easy Sharing options.

# Technology Involved

NoSQL  
Database

Reactive UI  
Component

REST API  
Service

HTML,  
CSS, JS

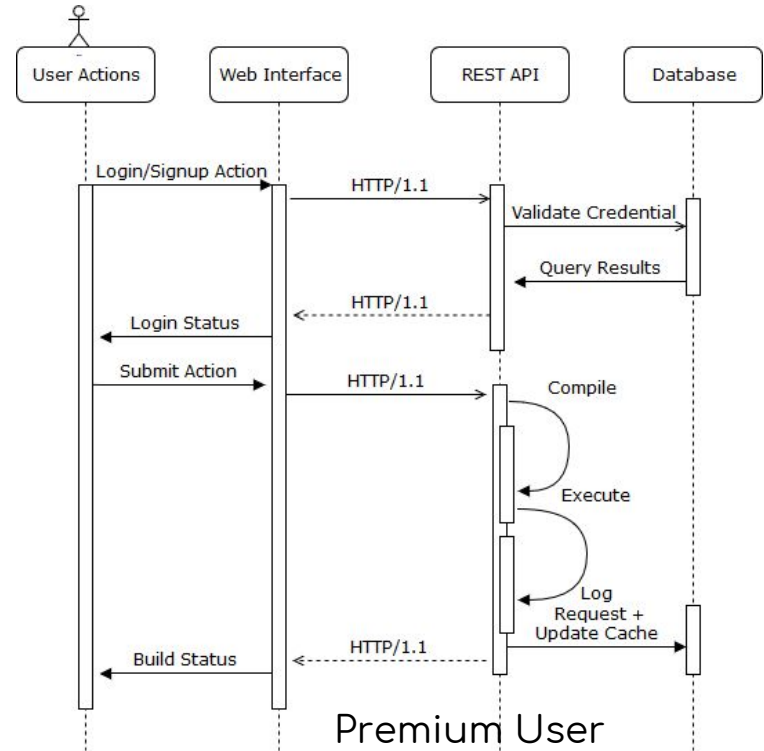
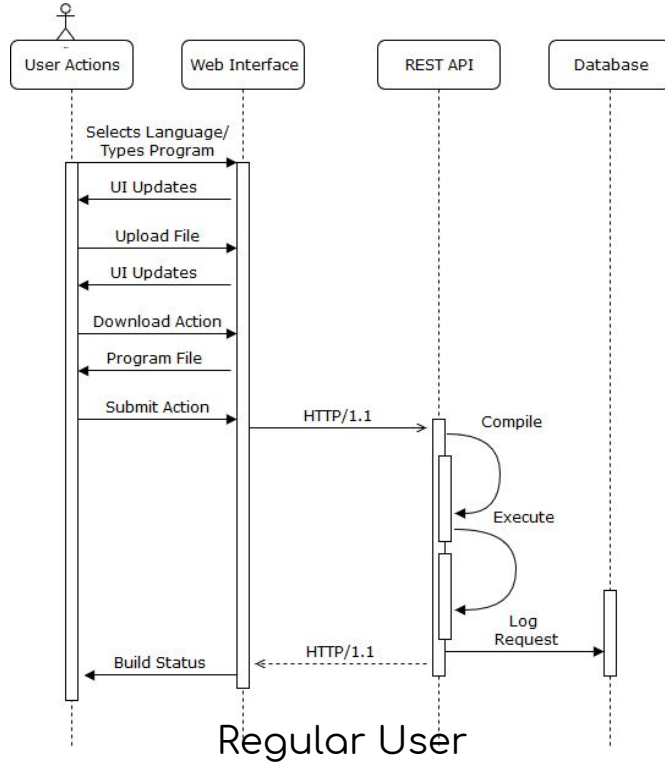
JSON

React + Redux + Axios

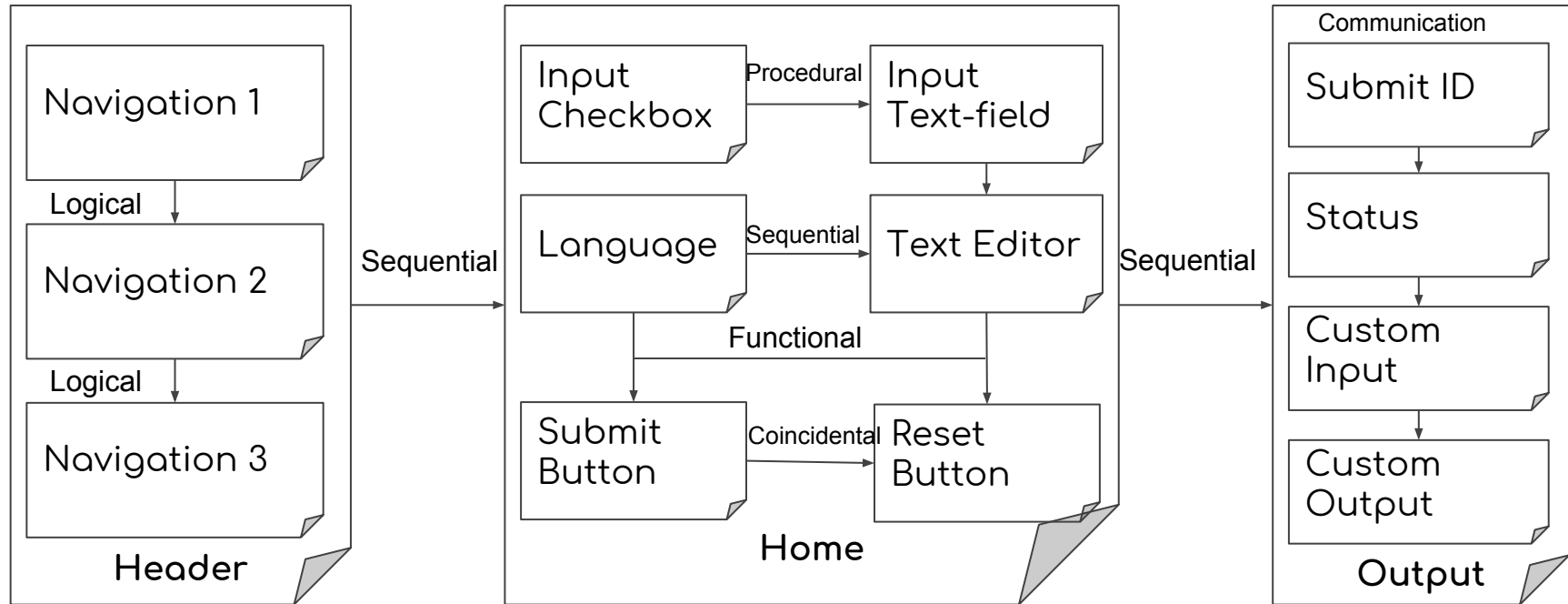
Flask + Python

Bootstrap

# Workflow Design

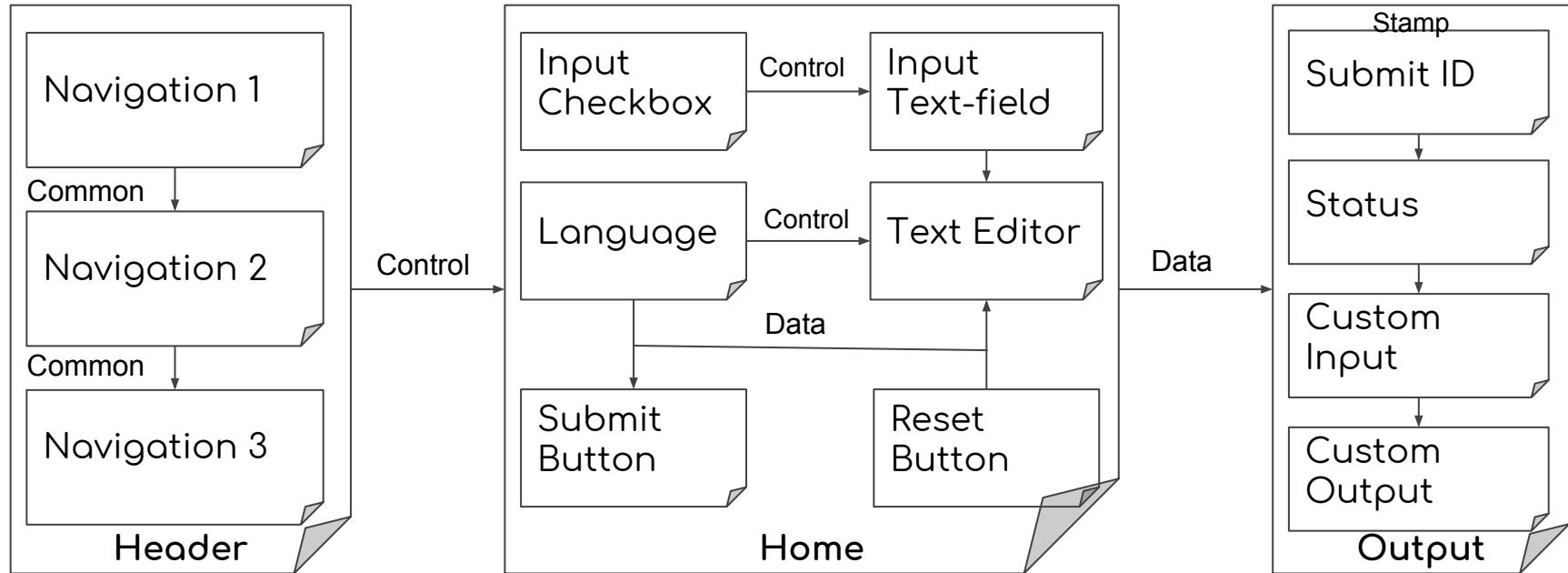


# Component-level Design (*a: Cohesion*)





# Component-level Design (*b: Coupling*)



# Coverage and Tests

## Coding Standard: ES6

```
{
  "parser": "babel-eslint",
  "parserOptions": {
    "ecmaVersion": 6,
    "sourceType": "module",
    "ecmaFeatures": {
      "impliedStrict": true,
      "jsx": true
    },
    "allowImportExportEverywhere": false
  },
  "env": {
    "browser": true,
    "commonjs": true,
    "es6": true
  },
  "plugins": [
    "eslint-plugin-react",
    "mocha"
  ],
  "rules": {
    "semi": ["error", "always"],
    "quotes": ["error", "double"],
    "no-console": 0
  },
  "extends": [
    "eslint:recommended",
    "plugin:react/recommended",
    "mocha/test"
  ]
}
```

```
Testing Checkbox component
  should render a checkbox for custom input
  should call the change handler when checked

Testing Content Management component
  should display 'Home' page as default
  should change content according to props
  should give error message for invalid props

Testing File Button component
  should render a button for file upload
  should trigger file input when clicked

Testing Footer component
  should check for API status
  should render label with API status
  should check for API info
  should render span element with API info on it

Testing Header component
  should render three 'Navlink' element
  should select default Tab
  should call the handler when clicked

Testing Home component
  should render a Language component
  should render three operating buttons
  should render the text editor
  should render a Checkbox component
  should render a EmailBtn if user logged in

Testing Custom Input component
  should render a textarea for input
  should call the change handler when value modified

Testing Language component
  should render a dropdown list of options
  should select default language
  should call the change handler when selected a value

Testing Reload Button component
  should render a button for reload
  should call the reload handler when clicked

Testing Submit Button component
  should render a button for submit
  should call the submit handler when clicked

28 passing (142ms)
```

```
Checking form reducer
  Other than form submission
    should not mutate for other action
  Form submission
    formReducer: { type: 'FORM_SUBMISSION', payload: {} }
    should update for form submission
    formReducer: { type: 'FORM_SUBMISSION', payload: {} }
    should hold error message in case of empty value
    formReducer: { type: 'FORM_SUBMISSION', payload: { data: { id: 'RandomData' } } }
    should update data property if value exists
    should update data property if value exists

Checking API Info
  API is functioning properly
    should have valid checkInfo action
    should call checkInfo action
    should return a Promise Object
    should resolve 200 OK

Checking info reducer
  Other than status check action
    should not mutate for other actions
  Status check action
    infoReducer: { type: 'API_INFO', payload: {} }
    should update for status check action
    infoReducer: { type: 'API_INFO', payload: {} }
    should hold error message in case of empty value
    infoReducer: { type: 'API_INFO', payload: { data: 'RandomData' } }
    should update status if response is 200 OK

Checking user authentication
  Creating a new user
    should have a valid userAuth function
    should call login action
    should return a Promise Object
    should log in the current user
  Logging out current user
    should call logout action
    should return a Promise Object
    should log out the current user
  Logging in a valid user
    should call login action
    should return a Promise Object
    should log in the current user

Checking login reducer
  Other than login/signup action
    should not mutate for other actions
  Login/signup action
    loginReducer: { type: 'LOGIN_ACTION', payload: {} }
    should update for login
    loginReducer: { type: 'LOGIN_ACTION', payload: {} }
    should hold error message in case of empty value
    loginReducer: { type: 'LOGIN_ACTION', payload: { data: { username: 'RandomData' } } }
    should update user information if value exists
  Logout action
    logoutReducer: { type: 'LOGOUT_ACTION', payload: {} }
    should update for logout
    logoutReducer: { type: 'LOGOUT_ACTION', payload: {} }
    should hold error message in case of empty value
    logoutReducer: { type: 'LOGOUT_ACTION', payload: { data: { loggedIn: false } } }
    should logout user information if response exists

Checking API Status
  API is running
    should have valid checkStatus action
    should call checkStatus action
    should return a Promise Object
    should resolve 200 OK

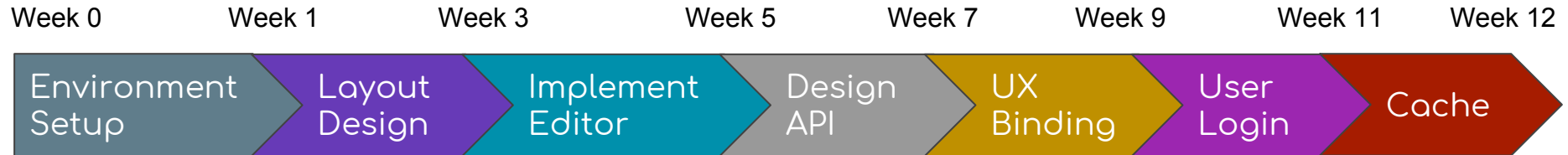
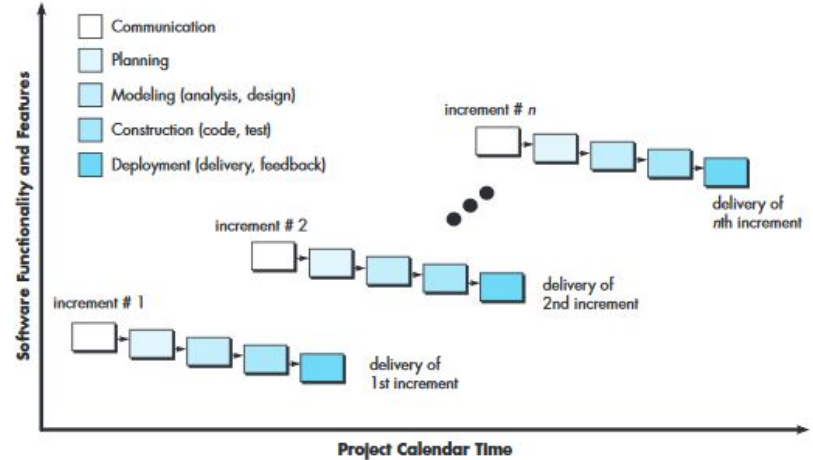
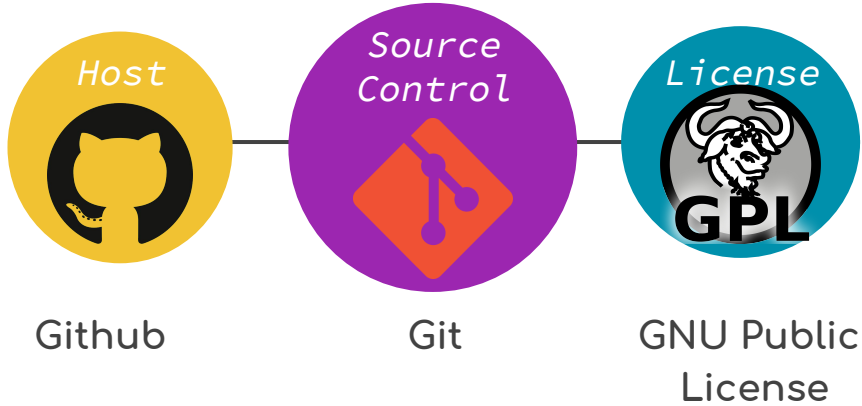
Checking status reducer
  Other than status check action
    should not mutate for other actions
  Status check action
    statusReducer: { type: 'API_STATUS', payload: {} }
    should update for status check action
    statusReducer: { type: 'API_STATUS', payload: {} }
    should hold error message in case of empty value
    statusReducer: { type: 'API_STATUS', payload: { data: 200 } }
    should update status if response is 200 OK

Checking form submission
  Form submission is working properly
    should call submit action
    should return a Promise Object
    should return same id as sent (300ms)
    should print 'Hello World' in output

42 passing (471ms)
```

Blackbox Testing: for UX functionalities  
Whitebox Testing: for UI rendering  
Framework: Mocha, ESLint, Enzyme

# Milestones



# Business Model

Our Product is aimed for Students or self-taught Programmers, developers and Academic Institutions.

- ❑ For Students they can practice their coding skills. And with help of this product they can understand their errors more accurately.
- ❑ For Developer (*Professional*) can verify their design goals, build small functionality and associated tests which will help them to build applications in faster pace.
- ❑ For Institution they can take their examination or any programming challenges using prime features.

# Room for Improvement

Due to short-time development and relatively less experience, the project is still in its beginner phase and demands to be prosper. Following is some few options that can improve the attraction and quality of service offered by the application.

- ❖ Plagiarism Checker: The application when running in examination/competition mode, will check for copied programs with high precision.
- ❖ Scoring Technique: The application should score the participants according to the accuracy, memory usage, execution time and time of submission.

# Thank You

This was IT-603