

ACKNOWLEDGEMENT

Knowledge in itself is a continuous process. At this moment of our substantial enhancement, I rarely find words to express my gratitude towards those who were constantly involved with me.

The completion of any inter disciplinary project depends upon coordination, cooperation and combined efforts of several resources of knowledge, creativity, skill, energy and time. The work being accomplished now, I feel my most sincere urge to recall and knowledge through these lines, trying my best to give full credit wherever it deserves.

I would like to thank our project guide **Mr. Ashish Patel** and & Head of Department Sir **Prof. A. K. Goyal** who advised and gave my moral support through the duration of my project. Without their constant encouragement I could not have been able to achieve what I have.

It's my good fortune that i had support and well wishes of many. I am thankful to all and those names which have been forgotten to acknowledge here but contributions have not gone unnoticed.

With Sincere Regards,

PARTH PATEL
1216BECE31038

ABSTRACT

CodeVilla is place of knowledge sharing and make faster development of your project, We try to make fork concept simple to you, we provide a store of your project information on the platform which also allows you to get github call for your repository details, we provide the thing which allows you to search your repository on the time and save you time to code.

We give also one new feature in this application, in every platform people need documentation, to know how it working?? We provide the thing as like module of the articles. Using articles people can publish their post share their POCS (proof of concept) and they can also able to make their own tutorial post inside, we provide markdown plugin which provide user to make easy and fast interface to write their articles.

TABLE OF CONTENTS

1.	INTRODUCTION.....	1
1.1	INTRODUCTION.....	1
1.2	SCOPE.....	1
1.3	PROJECT SUMMARY.....	2
1.4	OVERVIEW	2
1.5	PROBLEM DEFINITION	2
2	TECHNOLOGY AND LITERATURE REVIEW	3
2.2	BRIEF HISTORY OF WORK DONE	6
3	SYSTEM REQUIREMENTS STUDY.....	7
3.1	HARDWARE REQUIREMENTS.....	7
	4 GB Disk Space.....	7
	Development Side:.....	7
3.2	SOFTWARE REQUIREMENTS.....	7
	Development Side.....	8
	curl	8
3.3	CONSTRAINTS	8
1.	Hardware Limitations	8
2.	Interfaces to Other Applications.....	8
3.	Parallel Operations	8
4.	Higher Order Language Requirements	8
5.	Safety Requirements.....	9
6.	Security Consideration.....	9
4	SYSTEM ANALYSIS.....	10
4.1	STUDY OF CURRENT SYSTEM	10
4.2	PROBLEM AND WEAKNESS OF THE CURRENT SYSTEM	10
4.3	REQUIREMENTS OF NEW SYSTEM	10
4.4	FEASIBILITYSTUDY	10
4.5	FEATURES OF THE SYSTEM	12
4.6	SCOPE OF THE PROPOSED SYSTEM.....	13
5	SYSTEM DESIGN	14

5.1	SYSTEM MODULE DESIGN.....	14
5.2	DATA DICTIONARY	15
5.3	UML DIAGRAMS.....	20
6	SYSTEM LAYOUTS.....	23
7	CONCLUSION.....	31
8	BIBLIOGRAPHY	32



1. INTRODUCTION

1.1 INTRODUCTION

The internet's arrival and its subsequent popularity have made online work today that's much things are in the software development that's require much more affords from Software Engineer or Developer, now today scenario is that every one fight with new stuff of the software development in their regular life, they follows some tutorials and some documentation in their regular stuff, but Github which is really an extreme platform to save over codes, our codes are basically a POC (Proof of concepts) or REPOSITORY, Github is provide free could platform to store their code, 2 year working as freelancer i see this we are getting so many projects which based on research, like authorigestion, intrigration pof different modules like PayPal, strip or more..

We are follows some tutorials of experienced person they are write some blogs and post their code on github, or similar cloud platform, so people can show their code and implement it, but some times of SEO and other problems we have really hard to found their articles and their code.

That's the reason we are come up with an solution which provide you a place where you store your repository details and write articles on it, which can people are able to find easy and made their work smarter and faster.

1.2 SCOPE

Based on a variety of technical indicators, we've developed a self-evolving predictive algorithm. By noting previous success in identifying patterns and trends for an individual Repository and Articles. Individuality is key, each user can customize their own repository list and each stock will have a slightly modified algorithm.

1.3 PROJECT SUMMARY

The CodeVilla System Contains so many features, like data call from github, repository module and articles module we provide full text search to make search essay to user we provide authentication to the user from github so that's the reason we can evaluate our profile faster, codeVilla is a "SASS" base platform which provide really a better functionality to user.

1.4 OVERVIEW

The Web Application is build on "SaSS" platform application which allows users to store their repository information and maintain it and it is also allows as to a platform of write our own functional things or notes as like articles which is supports markdown which provide us a really better and faster platform to write and executes their things on the side ad matter it like

1.5 PROBLEM DEFINITION

Current scenario for trading is like this. People use blogs and wordpress or other platform to write their bogs some makes own but after one technical tutorial we need full code information to start work on that LIKE FORK CONCEPT but some time we didn't get it then after we are gone to github we face one othrt thing also which is this if we are getting any thing we search it but people on github they describe their all things in README.md file they really write less that's the reason we are not get full information of it we try to provide some faster interface to the people to make and manage faster and help other to make their production faster because to day we are live on time of open source.

2 **TECHNOLOGY AND LITERATURE REVIEW**

2.1 **ABOUT TOOLS AND TECHNOLOGY**

1) **Ruby On Rails**

Ruby On Rails is a web development technology where ruby is a programming language and rails is a framework which provides a rich set of api for developing a web-application.

Ruby on Rails, sometimes known as "RoR" or just "Rails," is an open source framework for Web development in Ruby, an object-oriented programming (OOP) language similar to Perl and Python.

The principle difference between Ruby on Rails and other frameworks for development lies in the speed and ease of use that developers working within the environment enjoy. Changes made to applications are immediately applied, avoiding the time consuming steps normally associated with the web development cycle.

According to David Geary, a Java expert, the Ruby-based framework is five to 10 times faster than comparable Java-based frameworks. In a blog posting, Geary predicted that Rails would be widely adopted in the near future.

Rails is made up of several components, beyond Ruby itself, including:

- Active record, an object-relational mapping layer
- Action pack, a manager of controller and view functions
- Action mailer, a handler of email
- Action web services
- Prototype, an implementer of drag and drop and Ajax

functionality

The principle difference between Ruby on Rails and other frameworks for development lies in the speed and ease of use that developers working within the environment enjoy. Changes made to applications are immediately applied, avoiding the time consuming steps normally associated with the web development cycle.

According to David Geary, a Java expert, the Ruby-based framework is five to 10 times faster than comparable Java-based frameworks. In a blog posting, Geary predicted that Rails would be widely adopted in the near future.

Rails is made up of several components, beyond Ruby itself, including:

- Active record, an object-relational mapping layer
- Action pack, a manager of controller and view functions
- Action mailer, a handler of email
- Action web services
- Prototype, an implementer of drag and drop and Ajax

Functionality:

Rails can run on most Web servers that support CGI. The framework also supports MySQL, PostgreSQL, SQLite, SQL Server, DB2 and Oracle. Rails is also an MVC (model, view, controller) framework where all layers are provided by Rails, as opposed to reliance on other, additional frameworks to achieve full MVC support. Invented by David Heinemeier Hanss, Ruby On Rails has been developed as an open-source project, with distributions available through rubyonrails.org.

2) sublime IDE

Sublime Text is a sophisticated text editor for code, markup and prose. You'll love the slick user interface, extraordinary features and amazing performance.

Cross platform

Sublime Text is available for OS X, Windows and Linux. One license is all you need to use Sublime Text on every computer you own, no matter what operating system it uses.

Sublime Text uses a custom UI toolkit, optimized for speed and beauty, while taking advantage of native functionality on each platform.

Plugin API

Sublime Text has a powerful, Python based plugin API. Along with the API, it comes with a built in Python console to interactively experiment in real time.

Customize anything

Key Bindings, Menus, Snippets, Macros, Completions and more - just about everything in Sublime Text is customizable with simple

JSON files. This system gives you flexibility as settings can be specified on a per-file type and per-project basis.

3) Github

GitHub is how people build software. With a community of more than 14 million people, developers can discover, use, and contribute to over 35 million projects using a powerful collaborative development workflow.

Whether using GitHub.com or your own instance of GitHub Enterprise, you can integrate GitHub with third party tools, from project management to continuous deployment, to build software in the way that works best for you.

4) heroku

Heroku (pronounced her-OH-koo) is a cloud application platform – a new way of building and deploying web apps.

Our service lets app developers spend their time on their application code, not managing servers, deployment, ongoing operations, or scaling.

Learn more about our architecture, pricing, and Elements Marketplace.

Heroku was founded in 2007 by Orion Henry, James Lindenbaum, and Adam Wiggins. It was acquired by Salesforce in 2011, and Heroku is now part of the Salesforce App Cloud.

2.2 BRIEF HISTORY OF WORK DONE

The Implementation had three phases.

- **Analysis Phase :**

- First the analysis was performed using the current system which meant knowing what a component does in the system. Then client requirements were generated and they were validated against the possible technology stack which was supposed to use for implementation.

- **Implementation Phase:**

- Implementation phase started when the analysis phase completed and development started to implement most basic functionality such as designing of Login module and integrating Web Services to provide data call from github.
- After basic implementation, yahoo finance APIs were optimized for better performance by caching the results generated by running algorithms and formulas on data supplied by yahoo finance APIs.
- After implementation of web application, ROR a platform.

- **Testing Phase :**

- Testing involved verifying that the charts generated by the library were accurate or not. Also the Web application had a client module which connected the web application with trading modules of the application. So the orders placed were checked to verify the web services were running as they were supposed to.

3 SYSTEM REQUIREMENTS STUDY

3.1 HARDWARE REQUIREMENTS

The selection of hardware is very important in the existence and proper working of any software. When selecting hardware, the size and requirements are also important.

Client Side:

Any Device which have browser and which is support javascript

Server Side:

Generally we use heroku and AWS but if they is not,

Web Server:

1 GB RAM,
4 GB Disk Space

Development Side:

Hardware Requirements:

- Processor: Intel Core i7 3220 CPU @ 3.32GHz
- RAM: 8GB
- Hard Disk Drive: 1 TB

(I am using hp envy m4 1015dx system for this project)

3.2 SOFTWARE REQUIREMENTS

Client Side

- Any Browser

Development Side

Sublime Text 3 (Linux),

PosgreSQL Server (Linux, Windows),

Web Browser

curl

3.3 CONSTRAINTS

1. Hardware Limitations

- The system is hosted on different platforms like Amazon AWS and Google App Engine so the hardware configuration at the host system can be bottleneck for the system performance.

2. Interfaces to Other Applications

- The System uses Ruby On Rails Framework for creating web application.
- The System uses octokit gem to get data from the github.

3. Parallel Operations

- Being a cross platform application allows the users to access the services on different platforms however the system have one login at a time constraint because of the security of the user accounts.

4. Higher Order Language Requirements

- The system uses JQuery 1.1 so its required that Browser atleast support the JQuery version and same goes for android version 2.3.

5. **Safety Requirements**

- The system can be great asset to the traders who use technical analysis for day to day trading and also want to be aware of fundamentals of the script. However the signals generated by the algorithms have high accuracy, but the expected movement in given timeframe cannot be guaranteed as one cannot predict how market will behave at the time. So trades made with the knowledge are high profit probability but does not give guarantee to protect the investment.

6. **Security Consideration**

- The system uses basic authentication system with email as username and password to grant user access to the system. However the only protection is this password and username combination so if user cannot protect the credentials then the trades taken by him have exposure risk. So use of strong password with length 8 or more characters and use of alphanumeric and special characters is encouraged for protection against malicious attacks.

4 SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

- We are not focus on any current system but we need to build some thing like that ehich provide us a really good compact.

4.2 PROBLEM AND WEAKNESS OF THE CURRENT SYSTEM

- We have ,any platform to show us which are open source project running on and which you can contribute we need some thing like which provide a really good platform to codder.

4.3 REQUIREMENTS OF NEW SYSTEM

- Based on a variety of technical indicators, we've developed a self-evolving predictive algorithm. By noting previous success in identifying patterns and trends for an individual stock, our algorithm continually adapts to each stock to make more accurate predictions. Individuality is key, each user can customize their own watch list and each stock will have a slightly modified algorithm. Users will be notified when the algorithm detects a change in the trend of any stock in their personalized watch list, allowing users to maximize their profits.

4.4 FEASIBILITYSTUDY

A Feasibility study is undertaken to determine the probability or possibility of either improving the existing system or developing a completely new system.

The Feasibility Studies are the major objectives of the current system. The estimate roughly the cost of each possible solution to the user problems and identify the solutions that might satisfy the user's needs within its budget and schedule.

The development through estimates of the benefit and drawback of each solution to obtain users and management views on all above procedures and to

obtain a decision from user and from management whether to commit an aliening part of the project.

This phase of the system is for testing system that it is feasible system that is feasible for Operational, Technical or Economical.

If the feasibility system is not successful then further processing for the developing system should be stopped.

- Operational Feasibility
- Technical Feasibility
- Economic Feasibility

1.Operational Feasibility

- This phaseof the system operations performed by proposed system. Here is the important point is that user will be operate the system.
- The important documents will be secure. The system will worked under any condition. The system will allow entering and retrieving of the data only to authorized user to operate the system. It will work any full efficiency and accuracy as used to work any particular computer.
- The system will be configurable and as generic as possible.

2. Technical Feasibility

- The Technical Feasibility is worked for the project is done with the present equipment, manual procedures, existing software technology and available technology hardware.
- In Technical Feasibility requirement is to run system better.
- Internet Connectivity
- System would be expandable and reconfigurable. Also system would guarantee accuracy, reliability and data security.

3. Economic Feasibility

- Economic Feasibility is the systems where the users have economic to this system the agency have to provide the required software.
- In any software, the licensed copy has to provide them and if the system is installed in network then network must be there in organization otherwise it was also large amount of money. So our system is economic feasibility only.
- This is very important phase of our system so must be given proper time and more importance. The cost of the manual system will be reduced. The clerical staff can also be reduced because of the efficient working of the computerized system. The material wastage by the manual system will also be eliminated. The cost of the computerized Admin Management System would not be that much, because of having most of the required hardware and software.

4.5 FEATURES OF THE SYSTEM

- User do not need to open account at our site we use authorigation from github.
- We motivate people to open source and knowledge sharing.
- We provide storeg and list of different repository. And their categories.
- We provide full text search.
- We provide own blog platform and it's sharing with social media profiles and other stuff like vote and comments.

4.6 SCOPE OF THE PROPOSED SYSTEM

- Based on a variety of technical indicators, we've developed a self-evolving predictive algorithm. By this platform we attract students of computer science, some geeks, coder and blogger to write, store their blog and their repository.

5 SYSTEM DESIGN

5.1 SYSTEM MODULE DESIGN

1. Log In

This module consists of custom social authentication

2. Repository

- Intuitive and modern, which show your github existing repos.
- You can make your own crude operation on yours pocs, and other can visit it only.

3. Articles

- User can make CRUDE operation of their files. And make if use of, other can show it share it, vote it, comment it

5.2 DATA DICTIONARY

Articles:

Sr.No	Field	Data type	Constraints
1	Id	Integer	Primary Key
2	title	string	Not Null
3	body	text	Not Null
4	github_profile_id	integer	Foren Key
5	created_at	datetime	Not Null
6	updated_at	datetime	Not Null

Categories:

Sr.No	Field	Data type	Constraints
1	name	string	Not Null
2	created_at	datetime	Not Null
3	updated_at	datetime	Not Null

categories_repositories:

Sr.No	Field	Data type	Constraints
1	category_id	integer	Primary Key
2	repository_id	integer	Not Null

commentator_comments:

Sr.No	Field	Data type	Constraints
1	creator_type	string	Not Null
2	creator_id	integer	Not Null
3	editor_id	Integer	Primary Key
4	editor_type	string	Not Null
5	thread_id	integer	Not Null
6	Body	text	Not Null
7	deleted_at	datetime	Not Null
8	created_at	datetime	Not Null

commentator_subscriptions:

Sr.No	Field	Data type	Constraints
1	subscriber_type	string	Not Null
2	thread_id	integer	Not Null
3	subscriber_id	integer	Primary Key
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null

commontator_threads:

Sr.No	Field	Data type	Constraints
1	commontable_type	string	Not Null
2	commontable_id	integer	Not Null
3	subscriber_id	integer	Primary Key
4	created_at	datetime	Not Null
5	updated_at	datetime	Not Null
6	closed_at	datetime	Not Null
7	closer_type	string	Not Null

Repositories:

Sr.No	Field	Data type	Constraints
1	name	string	Not Null
2	repo_id	integer	Not Null
3	html_url	string	Primary Key
4	github_profile_nickname	string	Not Null
5	fork	boolean	Not Null
6	description	text	Not Null
7	deleted_at	datetime	Not Null
8	created_at	datetime	Not Null
9	has_downloads	boolean	Not Null
10	language	string	Not Null
11	forks_count	integer	Not Null

12	watchers	integer	Not Null
----	----------	---------	----------

Taggings:

Sr.No	Field	Data type	Constraints
1	tag_id	integer	Primary Key
2	taggable_id	integer	Not Null
3	taggable_type	string	Not Null
4	tagger_id	integer	Not Null
5	tagger_type	String	Not Null
6	context	String	Not Null
7	created_at	Datetime	Not Null

Users :

Sr.No	Field	Data type	Constraints
1	email	String	Not Null
2	encrypted_password	String	Not Null
3	reset_password_token	String	Not Null
4	reset_password_sent_at	Datetime	Not Null
5	remember_created_at	datetime integer	Not Null
6	sign_in_count	Integer	Not Null
7	current_sign_in_at	Datetime	Not Null
8	last_sign_in_at	Datetime	Not Null
9	current_sign_in_ip	Inet	Not Null
10	last_sign_in_ip	Inet	Not Null
11	confirmation_token	string	Not Null
12	confirmed_at	Datetime	Not Null
13	provider	string	Not Null
14	created_at	Datetime	Not Null
15	updated_at	Datetime	Not Null
16	uid	string	Primary Key

5.3 UML DIAGRAMS

1. Activity Diagram

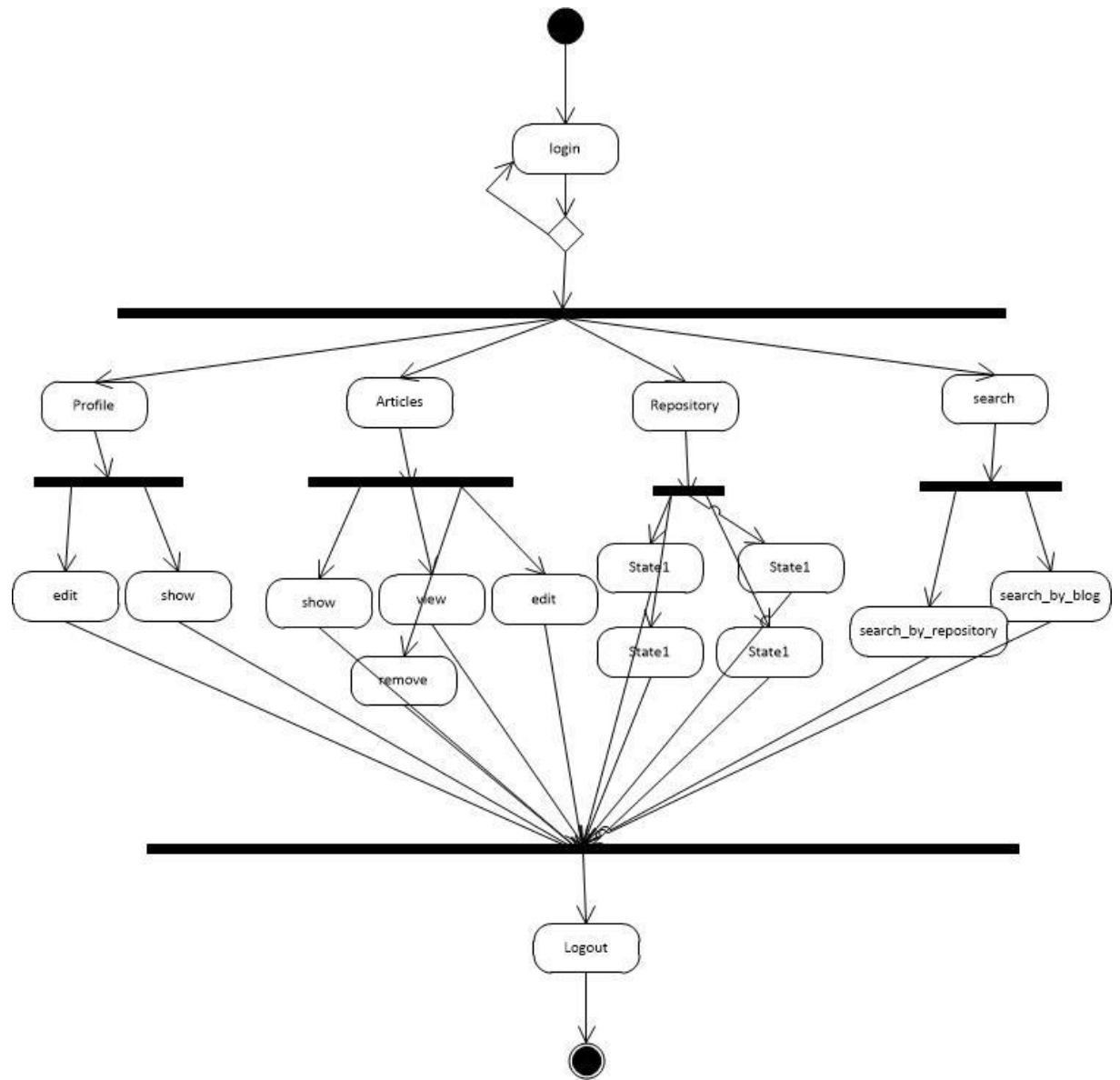


Figure 5.3.1 Activity Diagram of codeVilla

- Use Case Diagram

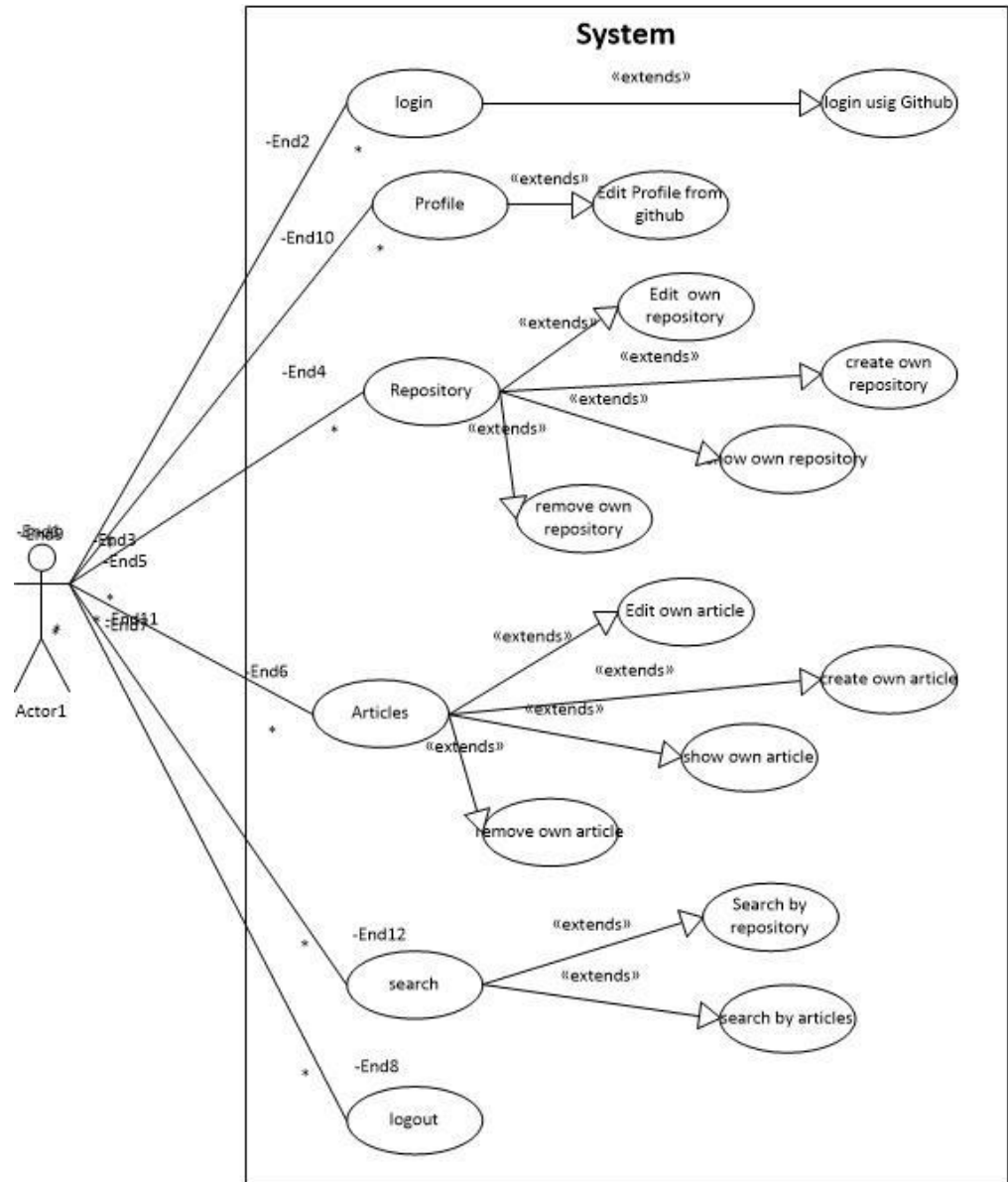


Figure 5.3.2 Use Case Diagram for CodeVilla

- Class Diagram

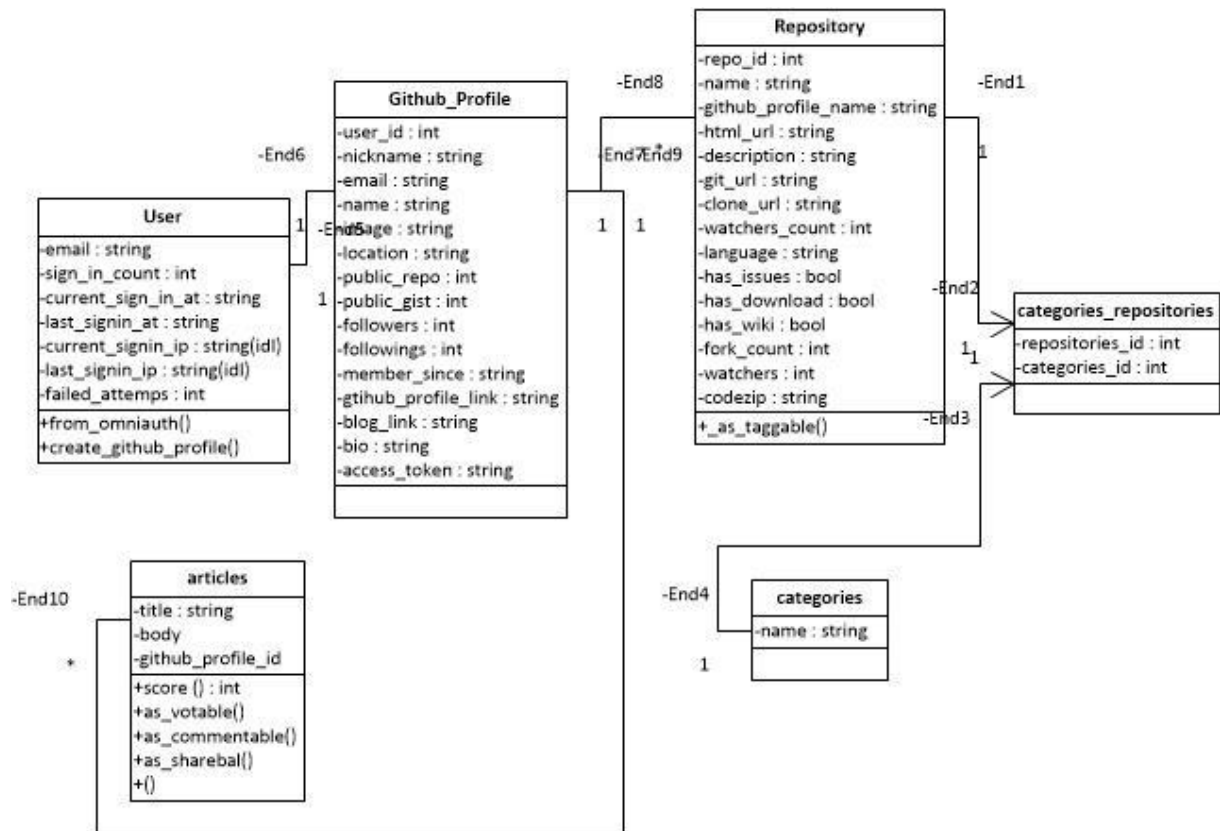


Figure 5.3.3 Class Diagram for CodeVilla

6 SYSTEM LAYOUTS

1. Log In Screen

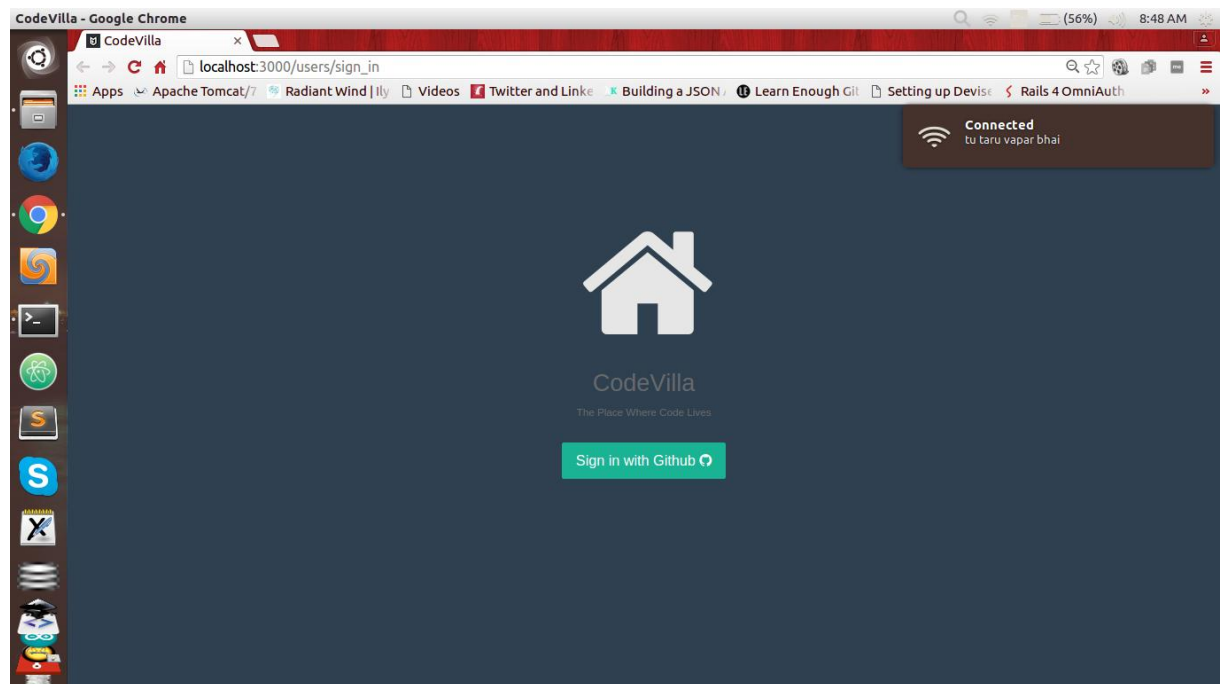


Figure 6.1 Login screen

```
parth@optimus: ~/Parth/Project/CodeVilla
parth@optimus: ~/Parth/Project/CodeVilla
Completed 200 OK in 3073ms (Views: 2868.7ms | ActiveRecord: 167.2ms)

-----
Started GET "/users/auth/github" for 127.0.0.1 at 2016-04-22 08:49:39 +0530
I, [2016-04-22T08:49:39.076113 #3024] INFO -- omniauth: (github) Request phase initiated.

Started GET "/users/auth/github" for 127.0.0.1 at 2016-04-22 08:49:43 +0530
I, [2016-04-22T08:49:44.004316 #3024] INFO -- omniauth: (github) Request phase initiated.

Started GET "/users/auth/github/callback?code=673a6b672c34d4a1c519&state=39591e5b29db3ba40553f06ad83edc5bcb3715e5d47d2b43" for 127.0.0.1 at 2016-04-22 08:49:46 +0530
I, [2016-04-22T08:49:46.717997 #3024] INFO -- omniauth: (github) Callback phase initiated.
Processing by CallbacksController#github as HTML
Parameters: {"code"=>"673a6b672c34d4a1c519", "state"=>"39591e5b29db3ba40553f06ad83edc5bcb3715e5d47d2b43"}
User Load (34.8ms) SELECT "users".* FROM "users" WHERE "users"."provider" = $1 AND "users"."uid" = $2 ORDER BY "users"."id" ASC LIMIT 1 [{"provider", "github"}, [{"uid", "8487101"}]]
(0.2ms) BEGIN
SQL (0.5ms) UPDATE "users" SET "last_sign_in_at" = $1, "current_sign_in_at" = $2, "sign_in_count" = $3, "updated_at" = $4 WHERE "users"."id" = $5 [{"last_sign_in_at", "2016-04-20 08:58:43.121133"}, [{"current_sign_in_at", "2016-04-22 03:20:02.019993"}, [{"sign_in_count", 15}, [{"updated_at", "2016-04-22 03:20:02.021793"}, [{"id", 2}]]
(14.4ms) COMMIT
Redirected to http://localhost:3000/
Completed 302 Found in 111ms (ActiveRecord: 50.1ms)

Started GET "/" for 127.0.0.1 at 2016-04-22 08:50:02 +0530
Processing by WelcomeController#index as HTML
User Load (0.5ms) SELECT "users".* FROM "users" WHERE "users"."id" = $1 ORDER BY "users"."id" ASC LIMIT 1 [{"id", 2}]
GithubProfile Load (35.7ms) SELECT "github_profiles".* FROM "github_profiles" WHERE "github_profiles"."user_id" = $1 LIMIT 1 [{"user_id", 2}]
Rendered welcome/index.html.erb within layouts/authen (10.5ms)
Rendered layouts/_navigation.html.erb (2.2ms)
Rendered layouts/_header.html.erb (0.9ms)
Rendered layouts/_footer.html.erb (0.4ms)
Completed 200 OK in 4363ms (Views: 249.0ms | ActiveRecord: 67.2ms)
```

Figure 6.1.1 login log

2. Dashboard

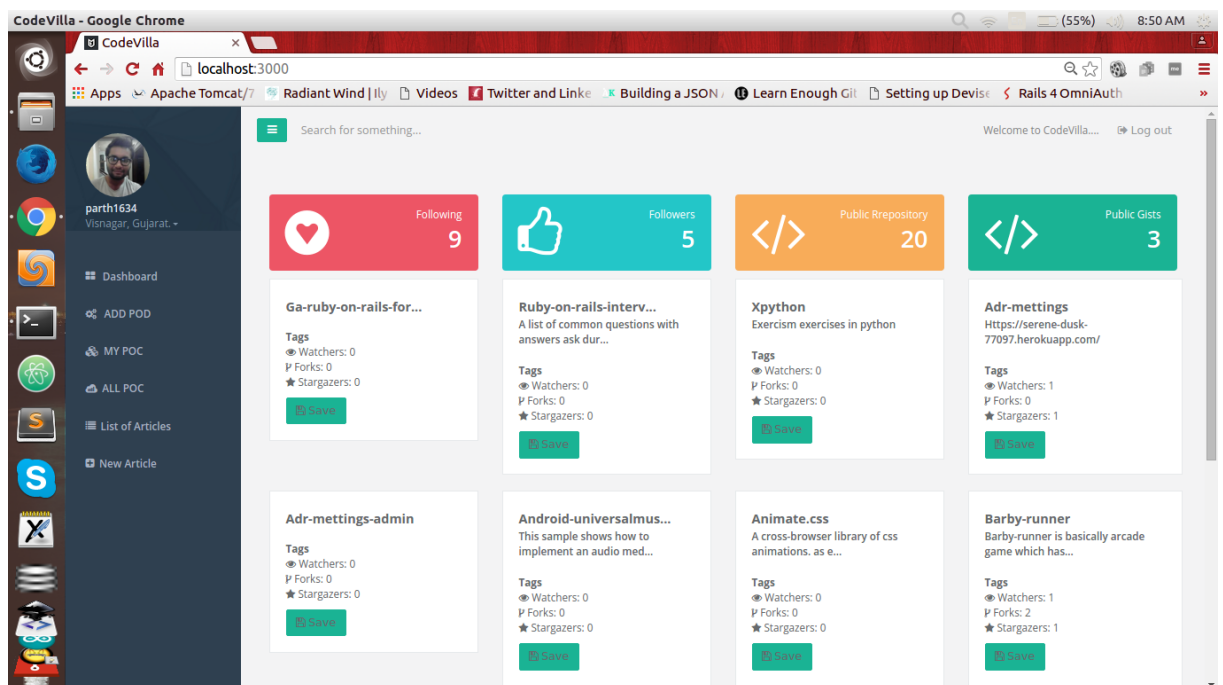
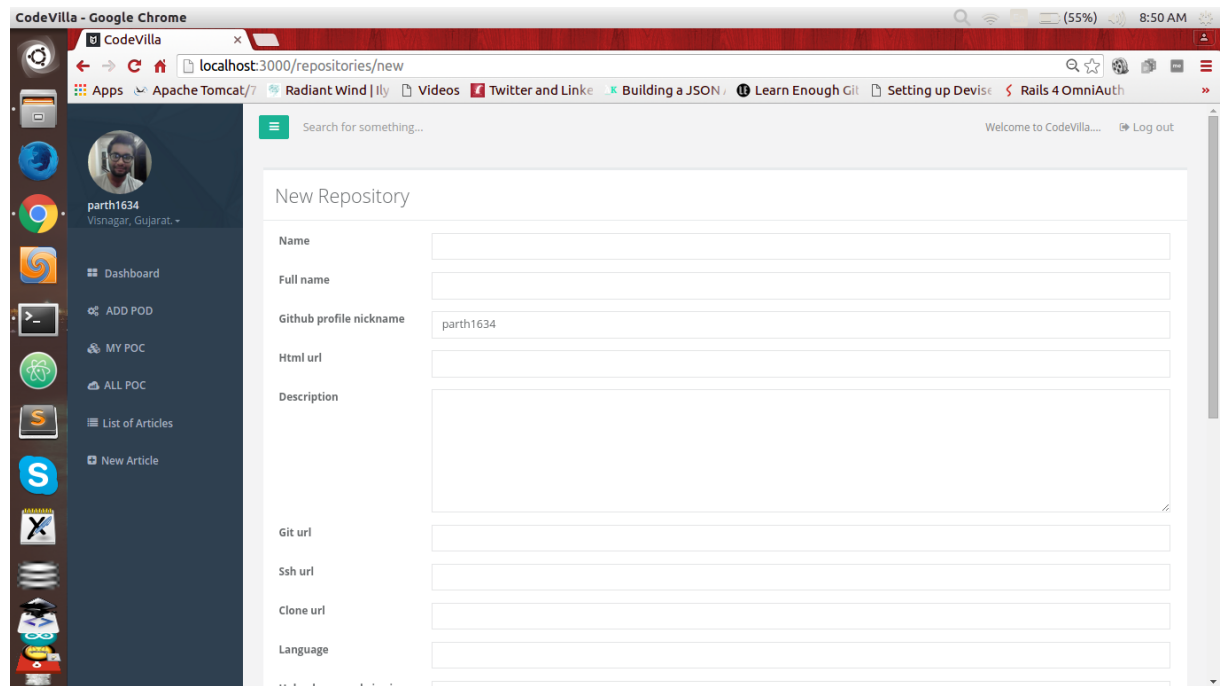


Figure 6.2 Dashboard of codevilla

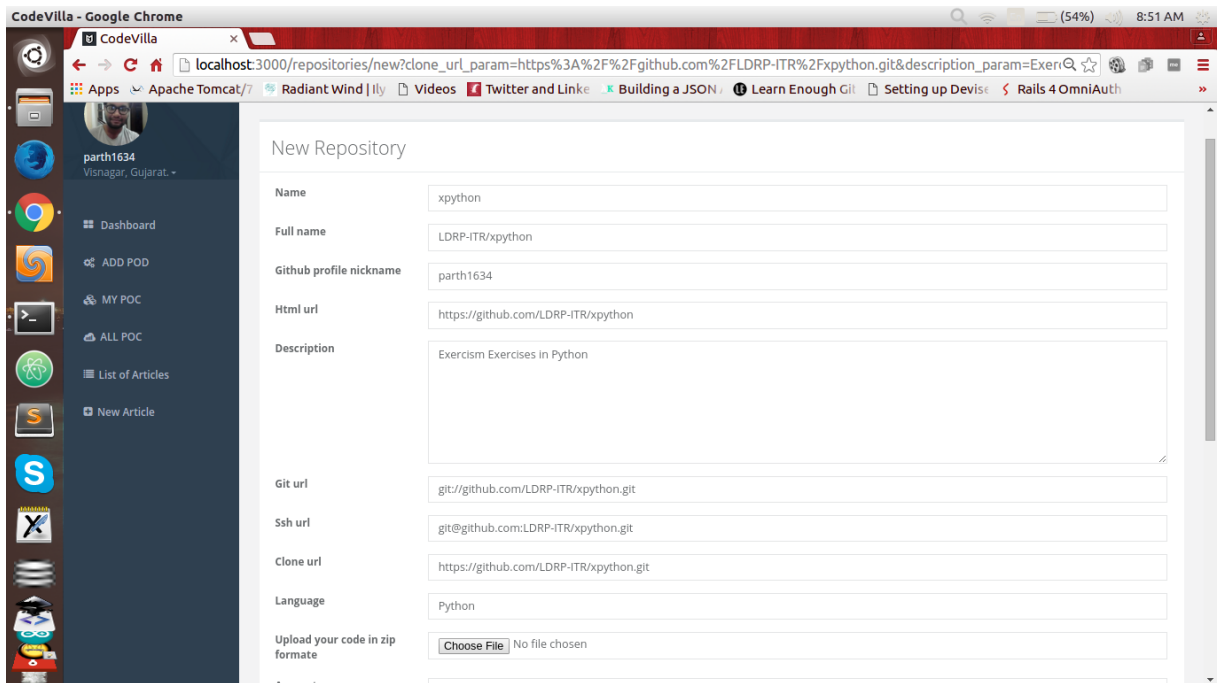
3. Repository form (auto& manual)



The screenshot shows the 'New Repository' form in the CodeVilla application. The form is located at the URL `localhost:3000/repositories/new`. The left sidebar shows the user profile for 'parth1634' and a navigation menu with options like Dashboard, ADD POD, MY POC, ALL POC, List of Articles, and New Article. The form fields are as follows:

Field	Value
Name	
Full name	
Github profile nickname	parth1634
Html url	
Description	
Git url	
Ssh url	
Clone url	
Language	

Figure 6.3 Repository registration form



The screenshot shows the 'New Repository' form with the following fields populated:

Field	Value
Name	xpython
Full name	LDRP-ITR/xpython
Github profile nickname	parth1634
Html url	https://github.com/LDRP-ITR/xpython
Description	Exercism Exercises in Python
Git url	git://github.com/LDRP-ITR/xpython.git
Ssh url	git@github.com:LDRP-ITR/xpython.git
Clone url	https://github.com/LDRP-ITR/xpython.git
Language	Python
Upload your code in zip format	<input type="button" value="Choose File"/> No file chosen

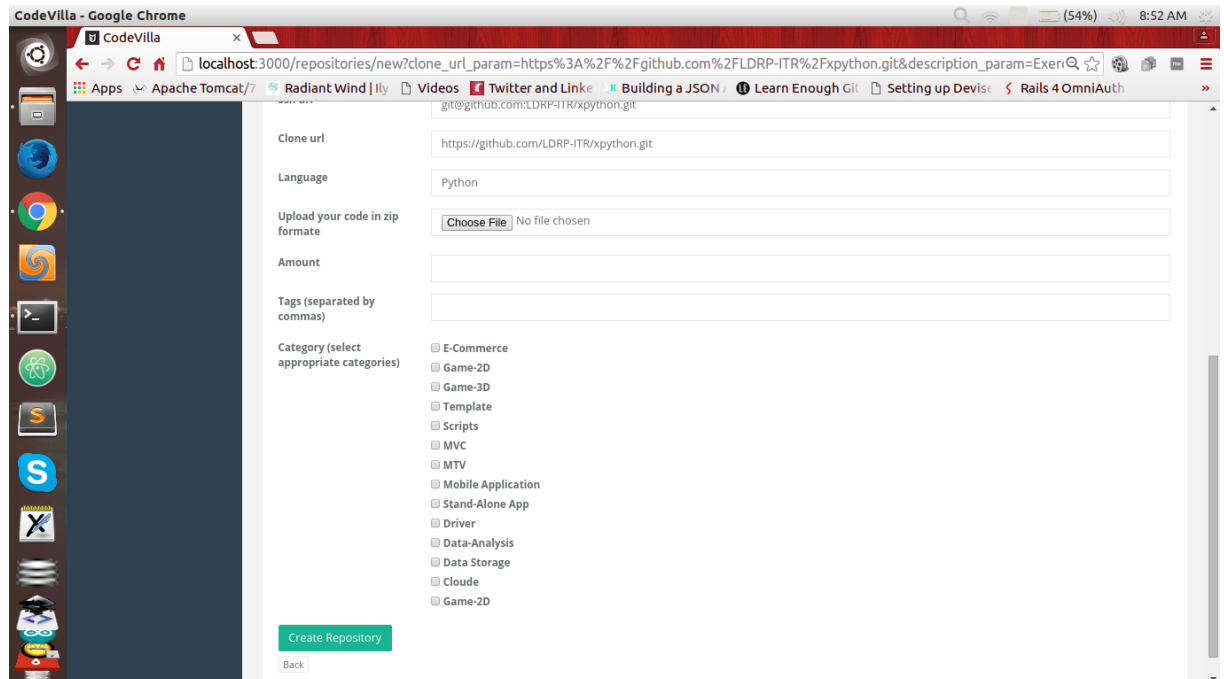


Figure 6.3.1 Auto filled form from github

4. User Repository list

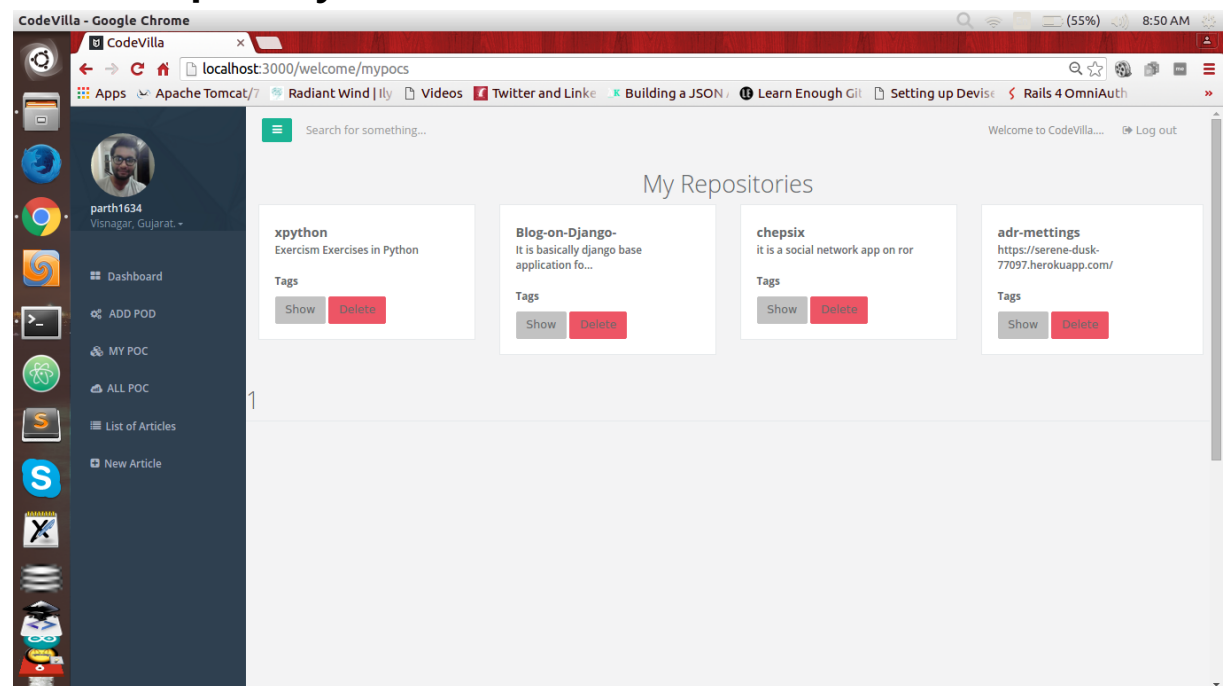


Figure 6.4 Users Repository list

5. All repository list

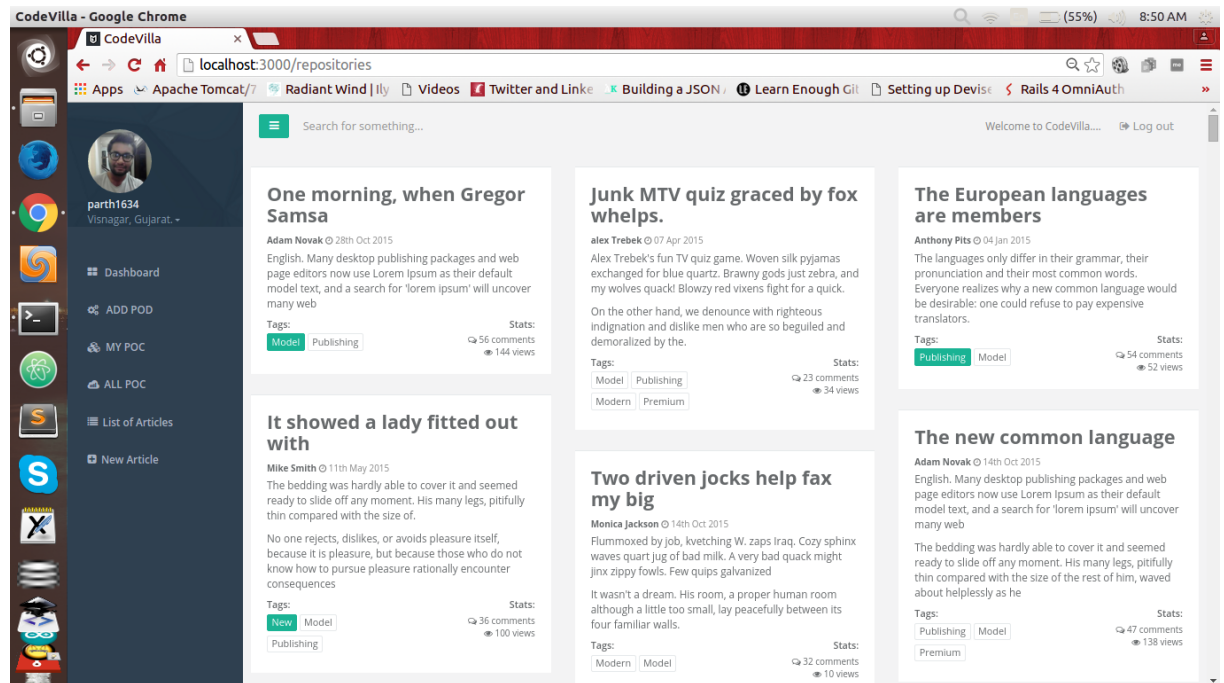


Figure 6.5 All repository list

6. Articles list

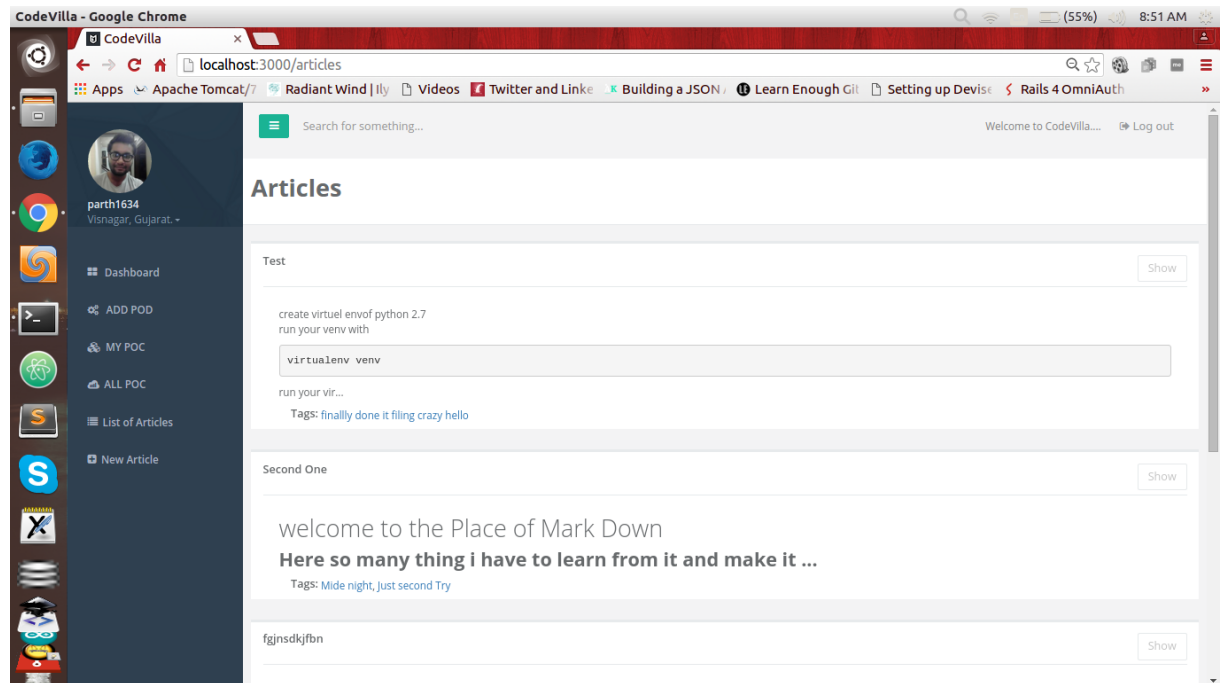


Figure 6.6 all articles list

7. New Article form

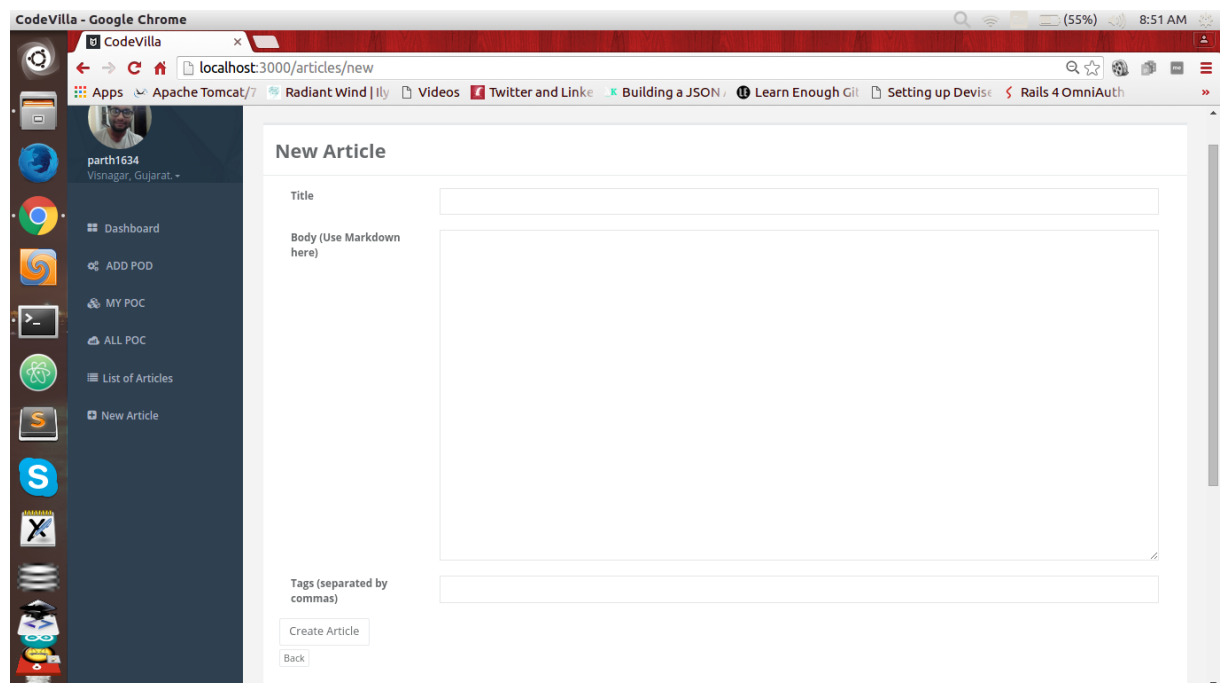


Figure 6.7 new article form with markdown

8. Articles show

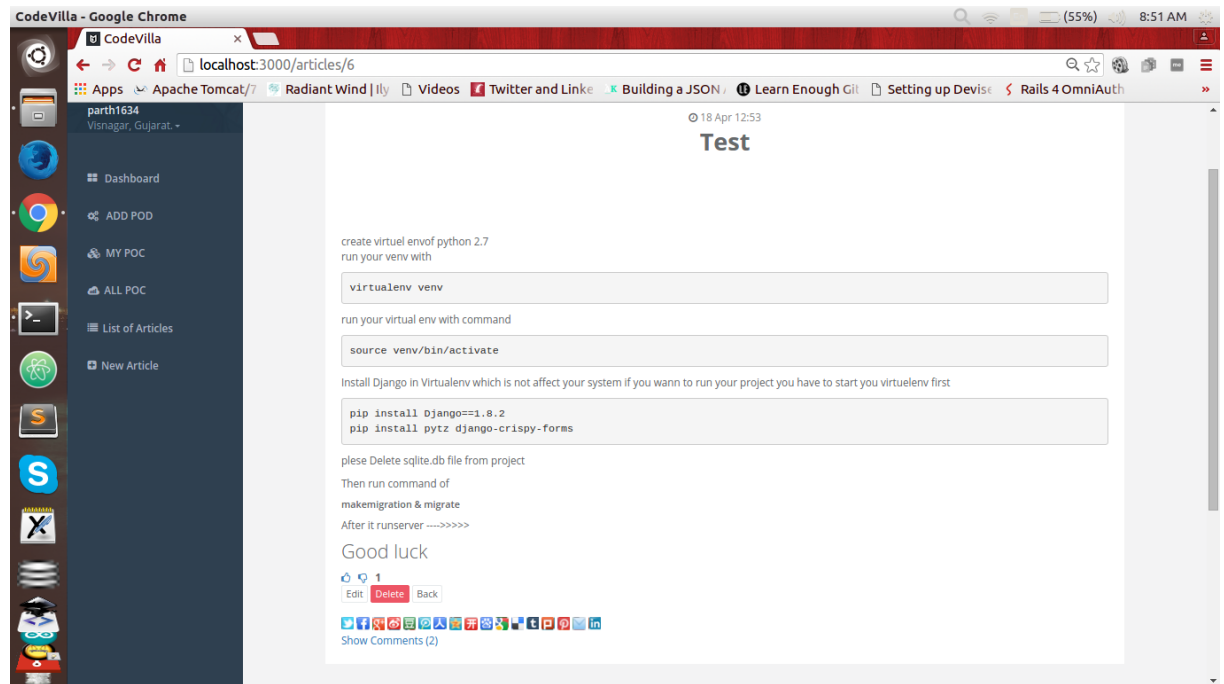


Figure 6.8 Articles show

9. Edit Articles form

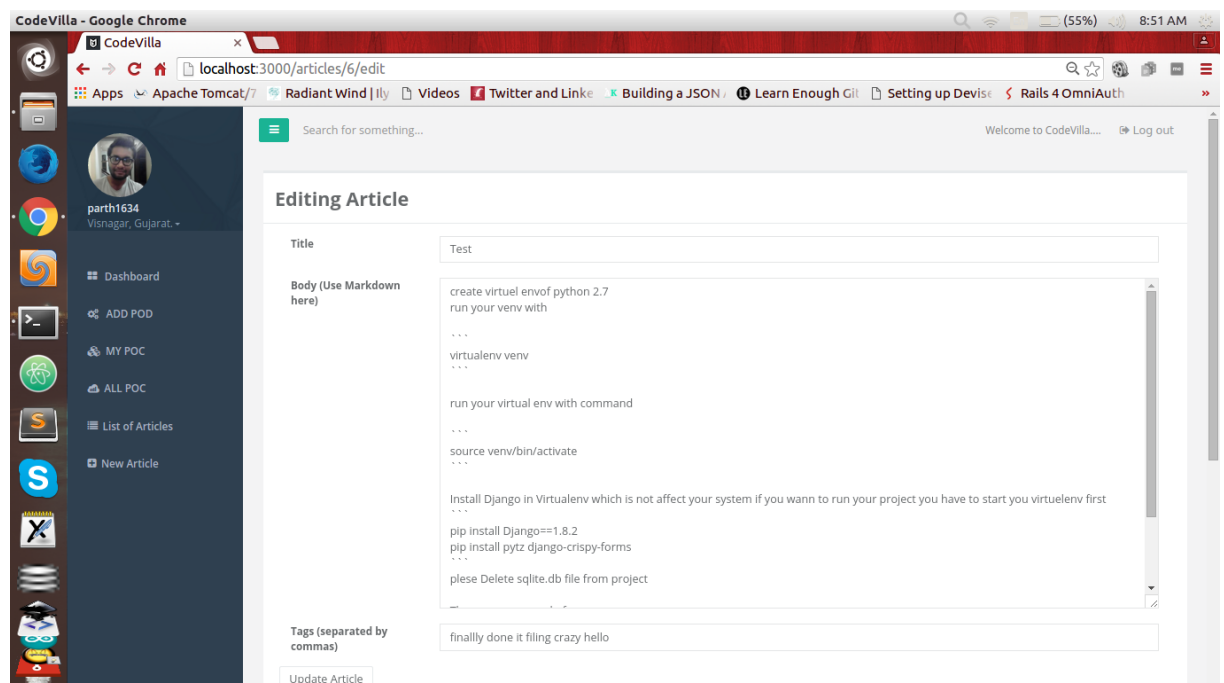


Figure 6.9 articles edit form

10. search result

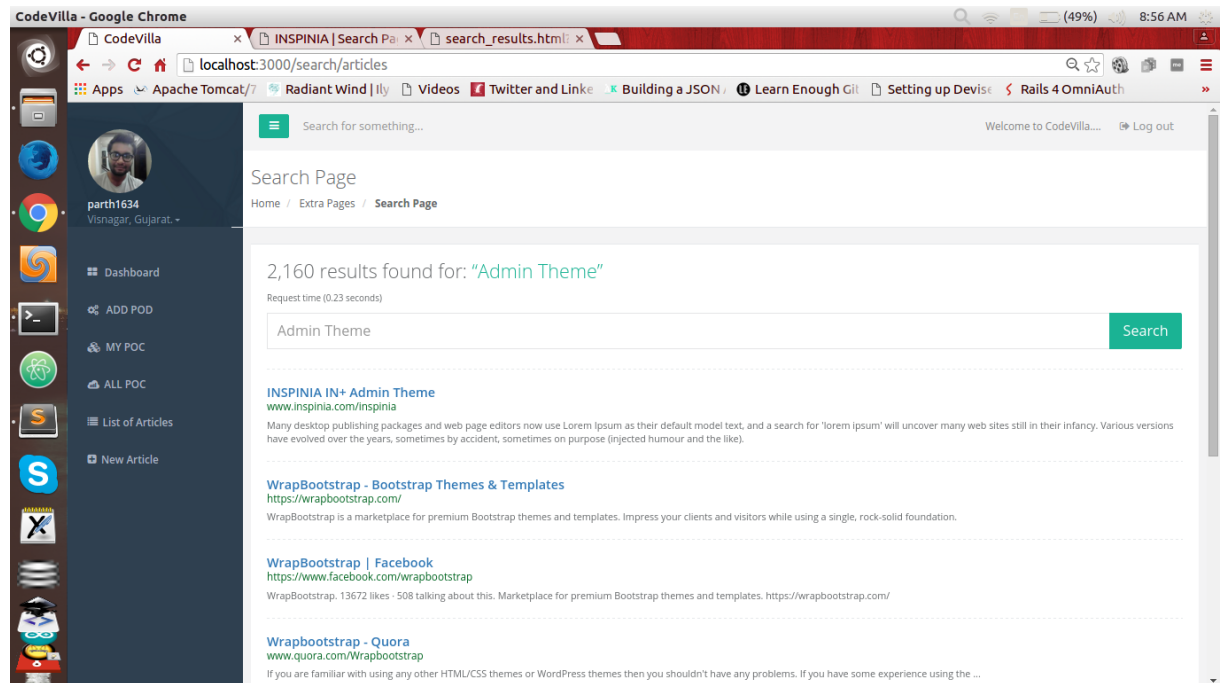


Figure 6.10.1 serching repository

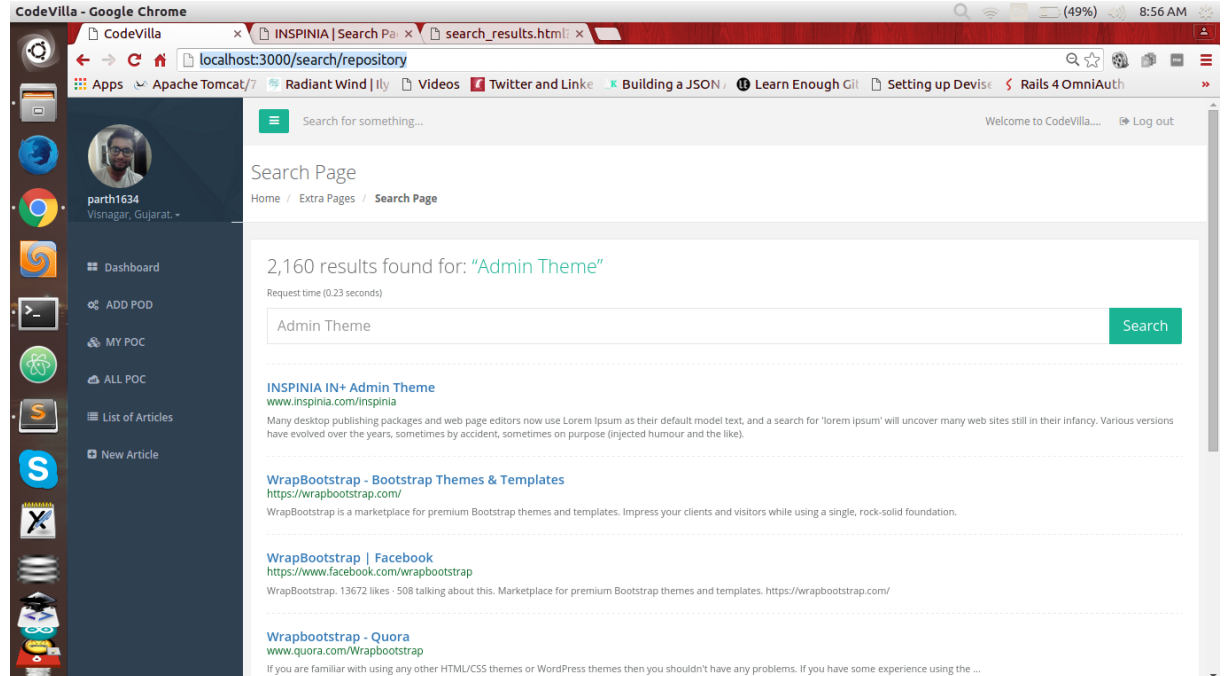


Figure 6.10.2 seareching articles

7 CONCLUSION

Now we are provide a interface to user which they can, store their pocs and their articles peoples can able to see their posts and their data towards to it, it is a beautiful platform to store your records, to help people make process of your development faster, just utilize the fork concept towards to user to faster development of your project.

8 **BIBLIOGRAPHY**

- <http://bootstrap-tutorial.bootstraptor.com/>
- <http://getbootstrap.com/getting-started/>
- <http://www.tutorialspoint.com/jquery/index.htm>
- <http://creately.com/> (for UML Diagram Drawing)
- <http://railsguide.com>