**CHAPTER 1**

**1. INTRODUCTION:**

Internet has revolutionized the way we connect and communicate with people. Its helping we connect with people who stay far away and also let us get knowledge of everything that going on in the world.

Vicinity explorer takes advantages of the networks that connect to explore the places that most crucial, the places near us. There are many events occurring nearby the very places that we live, work or spend most of our time in. Many a times we aren’t aware of them as there is few or no application that use the internet to notify the existence of these.

**1.1 Description:**

There are various events, happenings that take place near where we work, or stay or visit frequently. We are oblivious to most of the events that happen in our surroundings as there are few applications that use the internet to notify the about such events. Vicinity explorer takes advantages of networks to explore this crucial area.

The system helps users discover the area that’s most important to him/her (Home, Workplace or any other active location). It creates a geological bound within the users’ location and provides a set of features to the user pertaining to that location. The app is bound to a geographic location and prioritizes depth of content over breath.

1. Local offers, openings and other important events with PORTALS.
2. Sharing books, DVDs and other materials.
3. Discovering places and reviewing them locally.

**1.2 Motivation:**

Musubi is a disinter mediated messaging application for Android devices. Users can join with any of their existing identities (e.g. Facebook, Google, and Stanford) and share words, pictures, and videos with friends. Developers can build applications such as multiplayer games on top of Musubi. All Musubi messages are encrypted using Identity Based Encryption, and no central server stores any message. Personal informatics has been an active area of research in recent years. For example, Aseniero, et al. discuss visualizing daily activities, though not necessarily targeted to students, and focuses on viewing data on a macro level. Our journaling tool examines activities at multiple levels of granularity, though this work could serve as a supplement to ours. [1]

**1.3 Problem Formulation & Methodology Used:**

The system help with improving user understands with the vicinity, improving communication between organization and customers via strong local advertisement and content delivery, sharing with people living near you and in many other aspects pertaining to the location where the user spends most of his time. It will be most useful for people who have just moved in to this place or found a new place of work. The app prioritizes the location and provides services based on it.

Process Model: Waterfall Model:-

The approach that is used to develop the project will be object oriented approach. It is intended to develop the system components as objects and interconnect the system components by message passing between the objects. The implementation of various features of the project involves generation of different classes.

The process model to be used for this project will be the waterfall model as shown in the figure. Using this method, the features of the project will be developed by following the steps of the system development life cycle linearly i.e. requirement gathering, analysis, design, implementation, testing and deployment. For any changes to the system, the same procedure will be followed, but with the use of agility model. In this project the steps of system development life cycle may be skipped or speeded up so that the component design is generated faster to meet the deadlines.

Deployment

Testing

Analysis

Implementation

Design

Requirement Gathering Gathering

Fig.1.3 Phases of Waterfall Model

**1.4 Relevance of the project:**

Any enterprise, franchise when opens a new branch wants to attract the nearby customer. The app allows them to open Portal. A portal is a communication mechanism through which the app will geo-multicast the advertisement and other content to the users nearby. This will improve the chance of the user to respond to the advertisement as he/she lives nearby.

The need for book and DVD sharing is increasing for avid readers and voracious movie watchers. The app allows a system where in a person can share his book or DVDs with persons near him. This helps him to easily meet the other person. E.g. When a person is finished with a book, he can put it on sharing, the app will multicast this to users nearby, any other user interested in the book can trade it with a book in his procession. This will improve the sharing system and help the users save some money.

**1.5 Scope of the project:**

The scope of the system can be described by the services which it provides. The core service the application provides is called as “Portal”. This allows anyone to publish content locally. This is a great way to advertise events, sales etc. The Portal Mechanism is much more general and can be used for geo-multicasting news about events, offers, community formations as well as awareness system for any natural disaster. It can be used by anyone to multicast important messages pertaining to that area.

Another important service that the application provides is called local book sharing. This forms a sharing infrastructure for books or other tangible things. People nearby can trade books, DVDs, comics etc. with the help of the application.

**1.6 Objectives of the project:**

The objective of the project is to allow users to take advantage of the Internet to explore their surroundings. The system will provide an easy-to-access infrastructure, which will help users to track all the events occurring near their home location in real time. As the location is fixed, there is no need for a GPS enabled smart phone. With push notification services, users will always be notified about all the happenings around their location and thus, they will never miss any important event.

This system also provides a great framework for location based advertisement. It allows content to be published locally for greater impact on customers. It is a boost for those who want to restrain their advertising area.

**CHAPTER 2**

**2. REVIEW OF LITERATURE:**

**EXISTING SYSTEM:-**

The college experience is multi-faceted, but not enough has been done in the past to extract personal informatics from daily student life. OmniStanford [1], a smartphone application that automatically logs on-campus location history. Using this information, we are able to introduce two new experience-enhancing features: discovery and journaling. The discovery feature allows students to find people with common characteristics who were at the same location at the same time. The journaling feature gives students the opportunity to associate tasks to locations, enabling personal tracking and recall.

**PEOPLE DISCOVERY:-**

The first application we present for location-based PI is discovering individuals with common personal attributes, such as department affiliation or residence. We present an interface through which a user can select his affiliations, along with an option to allow location sharing with other application users. OmniStanford automatically connects to Musubi to allow communication through existing identities with a common format. We provide a server component for each location that can look up the names and hashed identifiers of people who have matching characteristics. For example, if two new Computer Science students meet at orientation, their presence at the same event will be pushed to each student by the server due to a match on the Computer.

We provide a daily calendar-like interface to list all the check-ins for a day and a drag-and-drop interface for the user to assign a tag to a check-in. The tag is created by user and can represent activities, food, physical exercises, and others. The application will remember the most recent tags so users do not need to type them every time. Once tags are assigned to check-ins, user can choose to visualize them in various ways. Currently we only built one view in D3.js: a pie chart that visualizes user’s daily, weekly, or monthly data. Both the selection interface and the D3 visualization are shown in Figure 1. The pie chart displays the portion of time spent at each location and the portion of time for each activity, which is personally meaningful for user. Aseniero [1] et al. built similar PI tools and we found that our two level pie chart meets most users’ needs. Besides reflection, journaling can also help students manage their time and change their behavior in the future.

**MULTICASTING ALGORITHM:-**

Flooding is probably the simplest multicast routing algorithm. The flooding algorithm can be used to deliver packets to nodes within a location-based multicast group. The multicast flooding algorithm can be implemented as follows: Assume that a node S needs to send a packet to a specific multicast region, a circle in figure. Node S broadcasts the multicast packet to all its neighbors\_ – hereafter, node S will be referred to as the *sender* and nodes D, F, and G as the *multicast group members* (note that in Figure 1 all nodes present in the specified multicast region are, by definition, multicast group members). A node, say B or C, on receiving the packet, compares the specified region’s coordinates with its own location. (We assume that all hosts are able to determine their own location using GPS.) If the location of B is within the specified multicast region, node B will accept the packet. Node B will also broadcast the packet to its neighbors; if it has not received the packet previously (repeated reception of a packet is detected using sequence numbers). If node B is located outside the multicast region and the packet was not received previously, it just broadcasts the packet to its neighbors.[4]

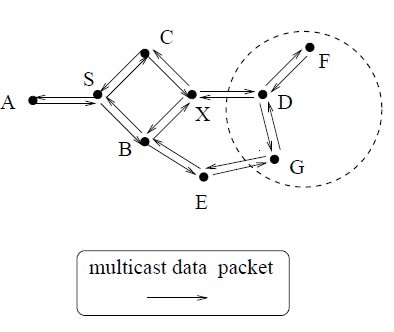


Fig. 2.1 Multicasting of data packets

Yes

No

No

Start

Yes

Redundant nodes exist?

End

*Tk*🡨*Ttrim*

*Tk*🡨*TMST*

*e*(*Tk*)>*e*(*TMST*)?

*TMST*: use the nodes in *Tk* to generate an MST

*Ttrim*: delete the redundant nodes and edges

*Tk*: a tree built by an ant *k*

Fig. 2.2 Flowchart for Multicasting Algorithm

**CHAPTER 3**

**3. SYSTEM STUDY AND ANALYSIS:**

**3.1 Existing System/ Concept:**

The first application present for location-based PI is discovering individuals with common personal attributes, such as department affiliation or residence. We present an interface through which a user can select his affiliations, along with an option to allow location sharing with other application users. OmniStanford automatically connects to Musubi to allow communication through existing identities with a common format. We provide a server component for each location that can look up the names and hashed identifiers of people who have matching characteristics. For example, if two new Computer Science students meet at orientation, their presence at the same event will be pushed to each student by the server due to a match on the Computer. We provide a daily calendar-like interface to list all the check-ins for a day and a drag-and-drop interface for the user to assign a tag to a check-in. The tag is created by user and can represent activities, food, physical exercises, and others. The application will remember the most recent tags so users do not need to type them every time. Once tags are assigned to check-ins, user can choose to visualize them in various ways. The pie chart displays the portion of time spent at each location and the portion of time for each activity, which is personally meaningful for user. Ascenario et al built similar PI tools and we found that our two level pie chart meets most users’ needs. Besides reflection, journaling can also help students manage their time and change their behavior in the future.

**3.2 Proposed System/ Concept:**

The App (Web, Mobile and Desktop) will require the user to input his HOME location, with this location it can give the following services to the user Location Based Friend Search, Food Zones, Help, Marketing Portals, Books and CDs sharing, Location Based News and Offers Roller, Campaign Launcher, Group Formation and Meet Ups, Startups, Jobs and Internships, Real-time Location helps.

The application opens a new world that lies in your geographic bound, a small world that is yet to be explored, studied and experienced. The application is bound to your geographic bounds and prioritizes depth of content over breath.

We use Machine learning to constantly know your behavior and help you find people, places, jobs that are closest to your location which you may not know.

It will give you real-time updates about the happenings around you in the News Roller.

It will help marketing and consumer communication as organizations can easily advertise their products locally according to their needs and can create special zones relevant to their products.

It will feature Issue Clustering where users can write in their queries or any help required for any task, any nearby user can help them and earn points when the job is done. As users are nearby the chance of getting help increases rather than a worldwide or nationwide service.

Food zone will help you find need restaurants and diners near your location and will allow you to give feedback and rating which can be helpful for the owners as well as to others living near you. Again, the application prioritizes location and only shows you the places near you.

Book sharing allows users to share books by creating a virtual library where users can trade books and save money.

Jobs, startups and internships allow the firms and startups to locate nearby employees who satisfy their needs. Nearby jobs save travelling and help both the firm and the employee. Startups will get some benefits by getting local personnel.

Campaign launcher is a special feature which allows NGO and other organization to gather local crowd and start some campaigns for some good cause. This will allow youngsters to be a part of the cause. The communication can be consolidated by the means of the application.

Sometimes when we aren't at home and still want some services such as nearby movies, café or emergency service the app can still provide you with real-time service by obtaining your current location.

**3.3 Requirement Analysis: *( write requirements of the project)*** Should follow the SRS format

**3.4 Requirement Specification: *( any specific specification , if any)***

Should follow the SRS format

**3.5 Requirement Validation: *( validity of the input requirements)***

Should follow the SRS format

**3.6 Use-Case Diagrams and description:**



Fig. 3.6 Use Case

|  |  |
| --- | --- |
| **Use Case Name** | **Show Contents to the User** |
| Actors | User. |
| Pre-condition | User must be registered to the system and there must be some existing content in the database. |
| Post-condition | All the content relevant to the user’s location will be displayed to the user’s dashboard. |
| Scenario | 1. User will login with his credentials. 2. System will load all the content pertaining to the user’s location. 3. Each Post can be individually viewed and commented on. |

Fig. 4.1 Use Case

|  |  |
| --- | --- |
| **Use Case Name** | **Posting Contents to a specific location** |
| Actors | Publisher. |
| Pre-condition | Publisher must be registered. |
| Post-condition | The system will publish the content to the specified location. |
| Scenario | 1. Publisher should sign up. 2. Publisher has to create the content for his advertisement. 3. Publisher has to select the location where the content has to be posted. |

Fig. 4.1 Use Case

|  |  |
| --- | --- |
| **Use Case Name** | **Moderate Content** |
| Actors | Moderator. |
| Pre-condition | Moderator has to be registered to the service. |
| Post-condition | Moderator will moderate the content. |
| Scenario | 1. Moderator has to login with his credentials. 2. Moderator dashboard will consist of the posts which are yet to be moderated. 3. Moderator then can allow, deny or comment the post. |

Fig. 4.1 Use Case

**CHAPTER 4**

**4. ANALYSIS MODELING:**

**4.1 Data Modeling:**

**4.1.1 ERD:**

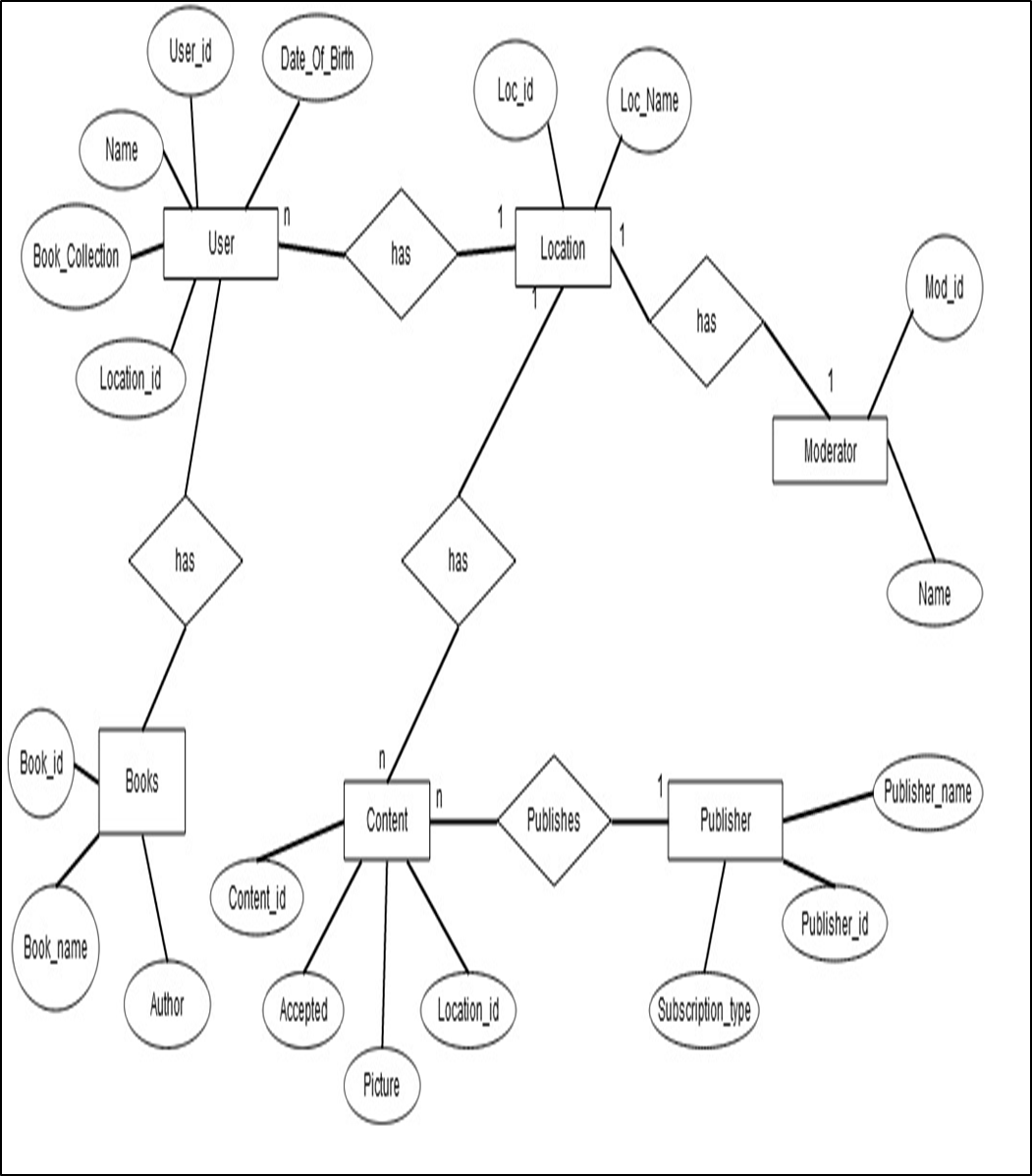


Fig. 4.1.1 Entity Relationship Diagram

**4.1.2 DATA DICTIONARY:-**

User

|  |  |  |  |
| --- | --- | --- | --- |
| User\_email | String | PRIMARY KEY | Unique Identification Number of User. |
| Username | String | NOT NULL | User’s Username. |
| Password | String | NOT NULL | User’s Password. |
| Name | Array | NOT NULL | User’s Full Name |
| Location\_id | Number | NOT NULL | Location of the User. |
| Gender | String | NOT NULL | Gender of the User. |
| Joined\_at | Date | NOT NULL | Joining date of the User. |

Publisher

|  |  |  |  |
| --- | --- | --- | --- |
| Publisher\_id | Number | PRIMARY KEY | Unique Publisher Identification |
| Username | String | NOT NULL | Username for the Publisher. |
| Organizaton\_name | String | NOT NULL | Name of the Organization. |
| Password | String | NOT NULL | Password for the Publisher. |
| Subscription\_type | String | NOT NULL | Monthly, Quarterly or Yearly |
| Subscription\_start | Date | NOT NULL | Start date of Subscription. |
| Subscription\_end | Date | NOT NULL | End date of Subscription. |
| Joined\_at | Date | NOT NULL | Date of Joining. |

Book

|  |  |  |  |
| --- | --- | --- | --- |
| Book\_id | Number | PRIMARY KEY | Unique Book Identification. |
| Book\_name | String | NOT NULL | Name of the Book. |
| Author | String | NOT NULL | Author’s Name. |
| Created\_at | Date | NOT NULL | When the book was shared. |
| Amazon\_link | String |  | Link for Purchasing |
| Bids | Array |  | Bids on the book. |
| Description | String | NOT NULL | Description about the book. |
| Image\_link | String |  | Image of the book. |
| Location | String | NOT NULL | Location for posting. |
| Tags | Array |  | Tags for categorizing |
| Trade\_with | Array | NOT NULL | Book names for trading |
| User | Array | NOT NULL | User details |

Moderator

|  |  |  |  |
| --- | --- | --- | --- |
| Moderator\_email | String | PRIMARY KEY | Unique Moderator Identification |
| Moderator\_name | String | NOT NULL | Name of Moderator |
| Moderator\_password | String | NOT NULL | Moderator Password. |

Post

|  |  |  |  |
| --- | --- | --- | --- |
| Post\_id | Number | PRIMARY KEY | Unique Moderator Identification |
| Body | String | NOT NULL | Text part of content |
| Tags | Array |  | Tags for categorizing. |
| Location | String | NOT NULL | Location for posting the content. |
| Published\_by | String | NOT NULL | Name of the publisher. |
| Published\_at | Date | NOT NULL | Date and Time of content published. |
| Status | Number | NOT NULL | Status for Moderation. |
| User\_comments | String |  | Comments by user. |
| Moderate\_comments | String |  | Comments by Moderator |

Bids

|  |  |  |  |
| --- | --- | --- | --- |
| User | String | NOT NULL | User’s Full name |
| Book\_name | String | NOT NULL | Name of Book |
| Body | String | NOT NULL | Something about the book. |
| Posted\_at | Date | NOT NULL | Date on which user has bided. |

Comments

|  |  |  |  |
| --- | --- | --- | --- |
| Moderator\_email | String | PRIMARY KEY | Unique Moderator Identification |
| Moderator\_name | String | NOT NULL | Name of Moderator |
| Moderator\_password | String | NOT NULL | Moderator Password. |

**4.2 State Diagrams /Activity Diagrams / Class Diagram** *Depending on the type of your project u may include the class diagram****.***



Fig. 4.2 Class Diagram

**4.3 Functional Modeling (DFDs *with specifications)***

**4.4 TimeLine Chart*:***

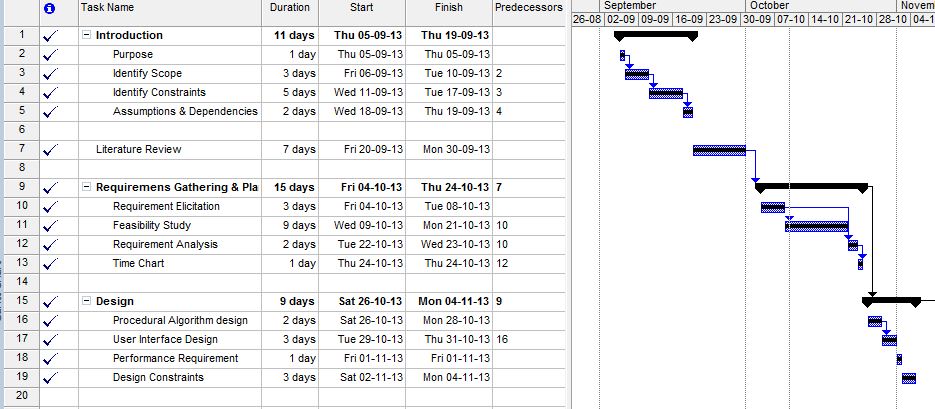


Fig. 4.4.1 Time Line Chart 1

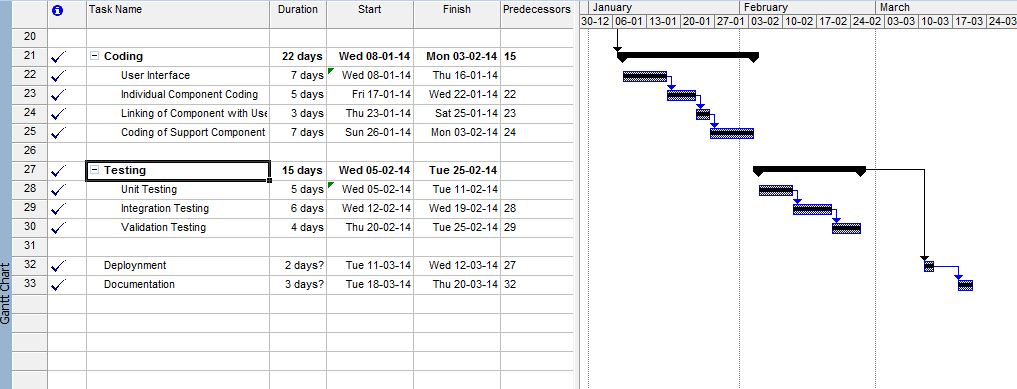


Fig. 4.4.2 Time Line Chart 2

**CHAPTER 5**

**5. DESIGN:**

**5.1 Architectural Design (*Project Flow /architecture* *with description)***

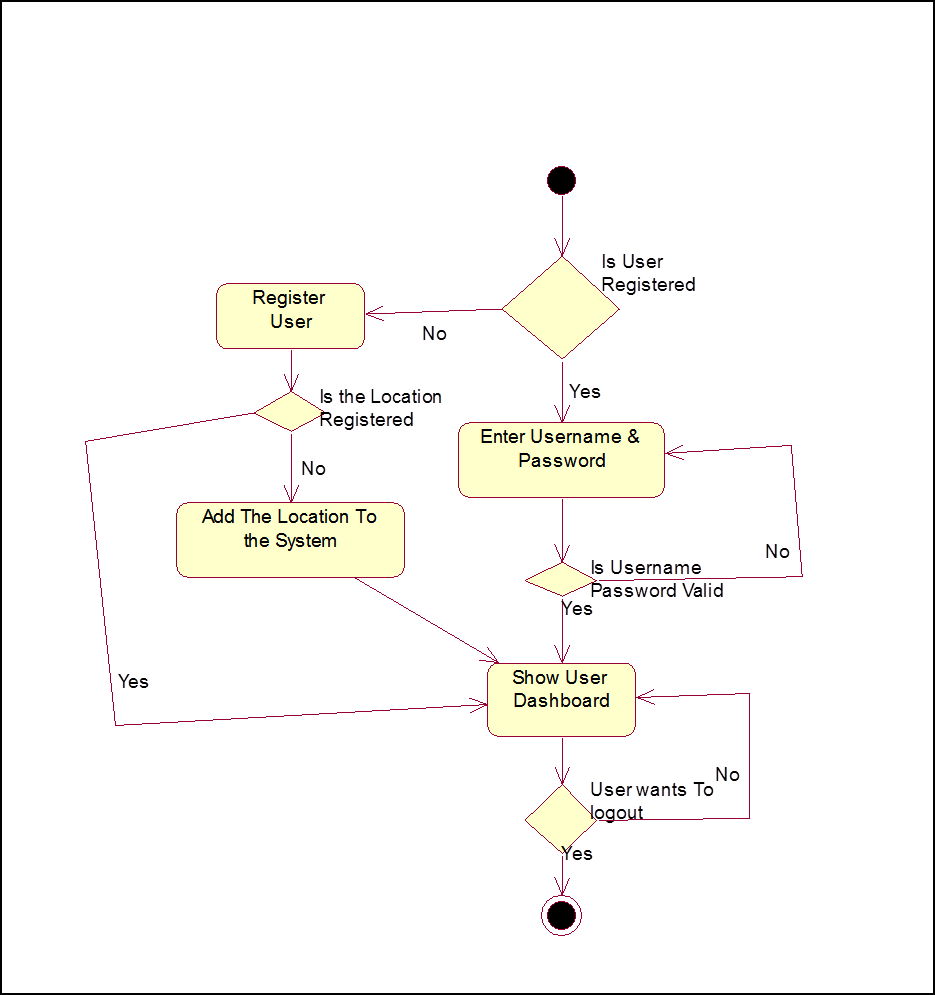


Fig. 5.1 Validating User



Fig. 5.2 Adding Content to the Dashboard



Fig. 5.3 Validating the Content

**5.2 User Interface Design** GUI for your project

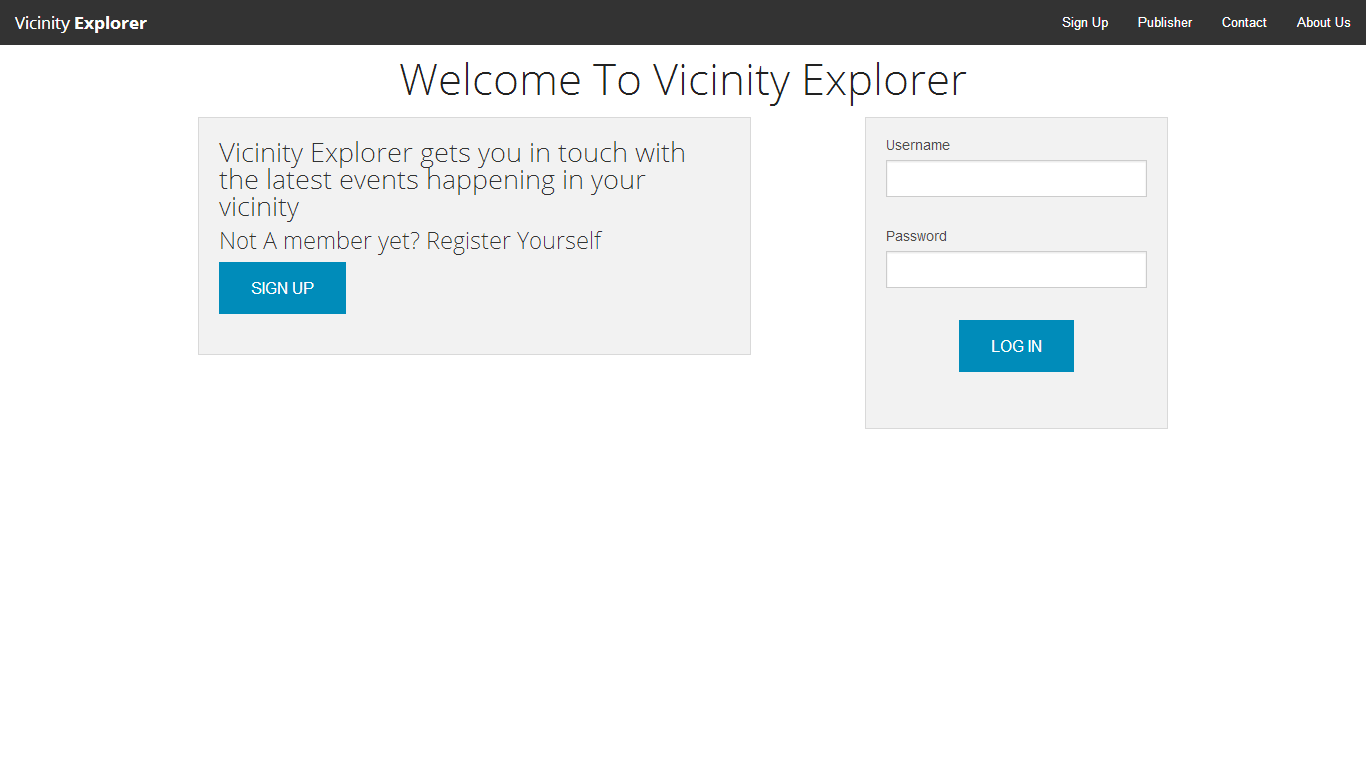


Fig. 5.2.1 User Homepage

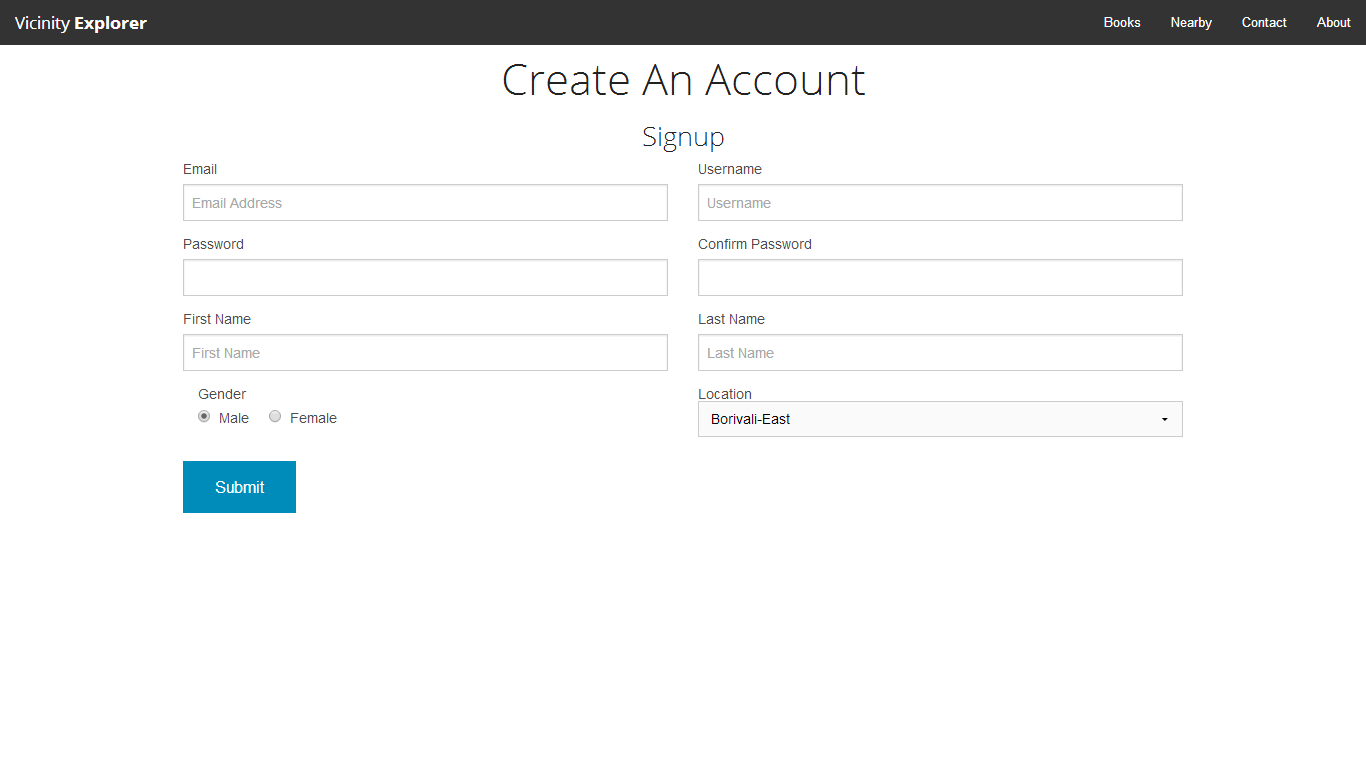


Fig. 5.2.2 User Signup Page

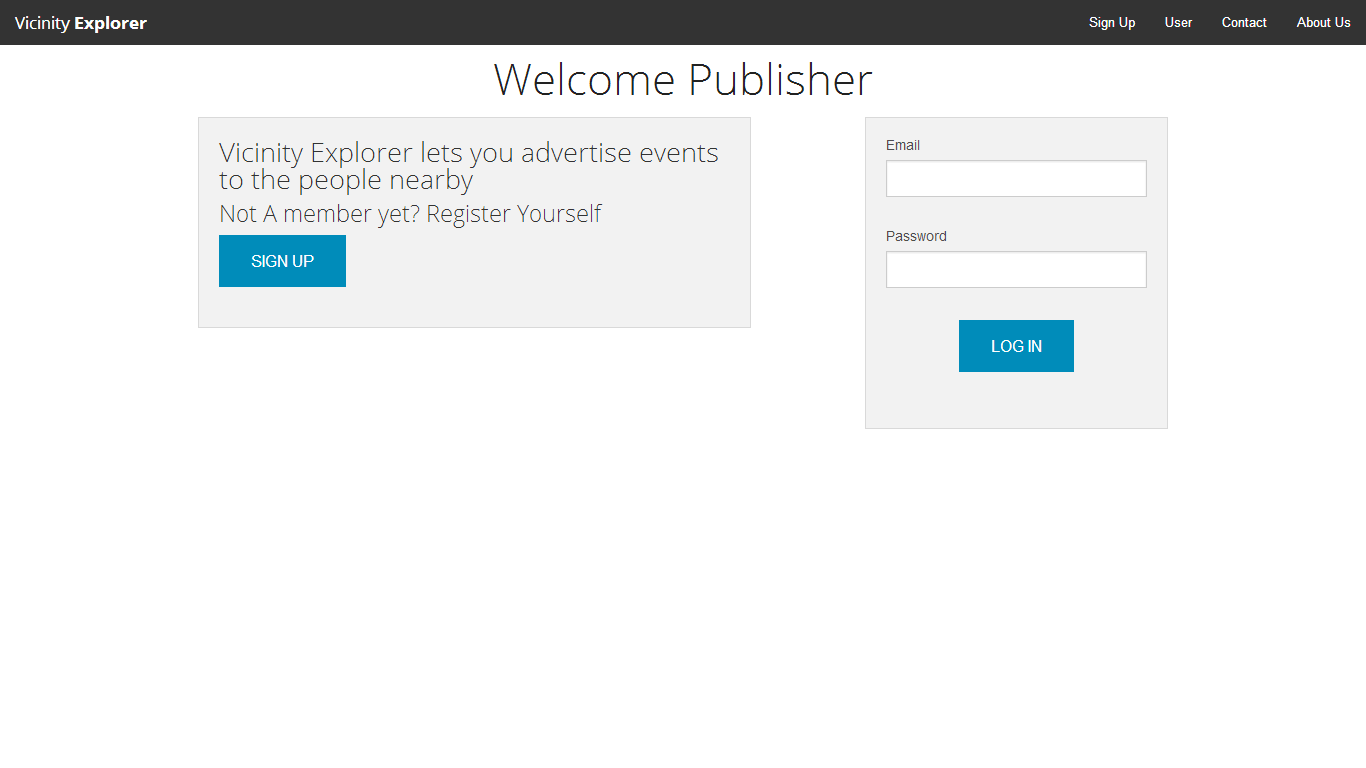


Fig. 5.2.3 Publisher Homepage

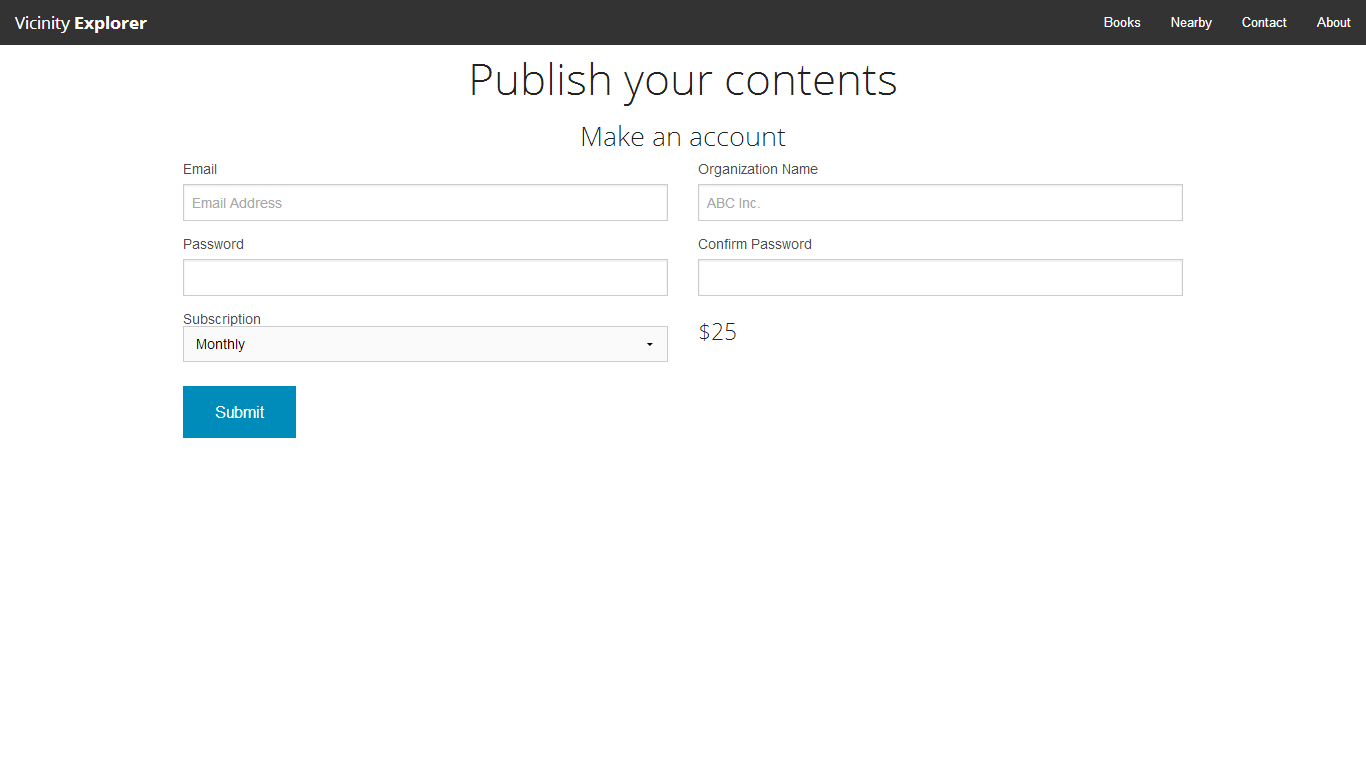


Fig. 5.2.4 Publisher Signup Page

Fig. 5.2.4 Publisher Dashboard

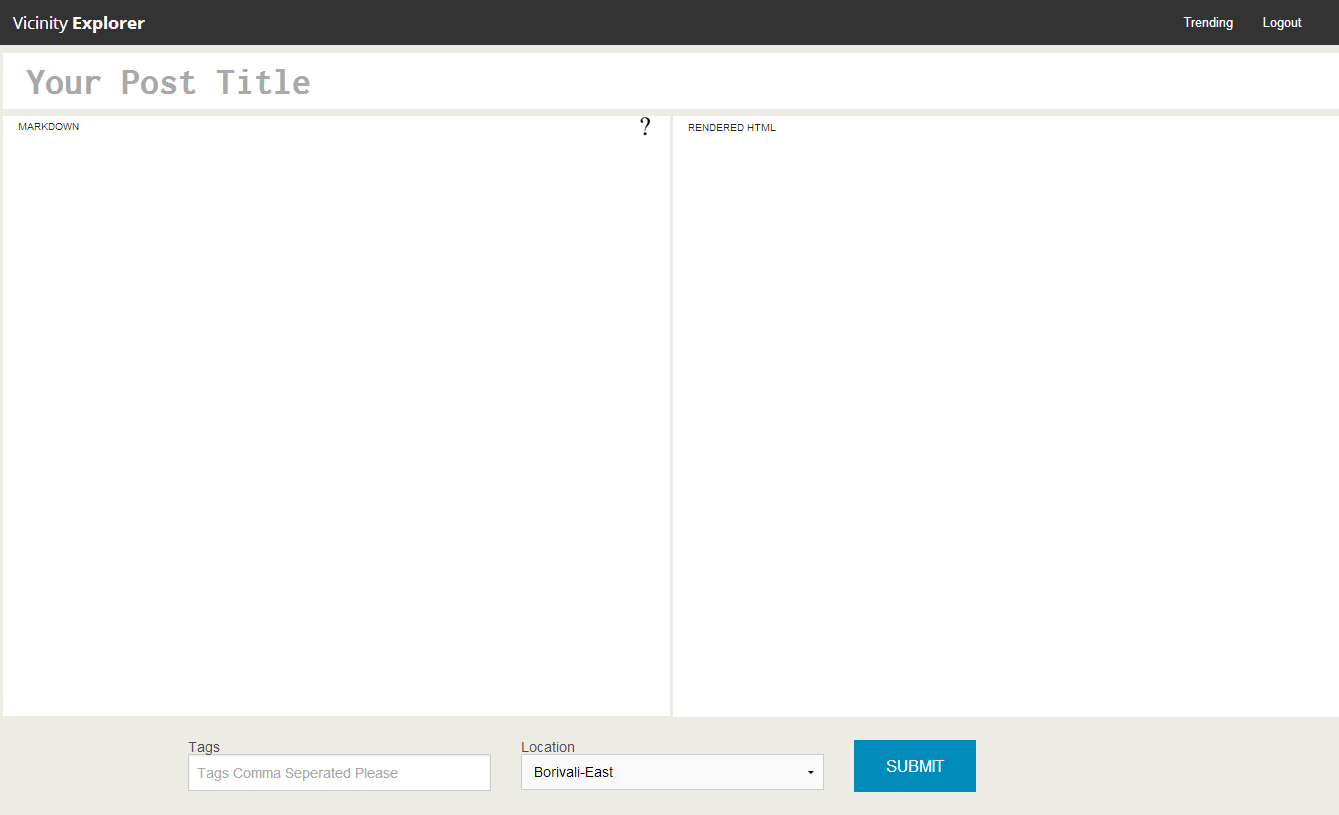


Fig. 5.2.4 Creating new Posts

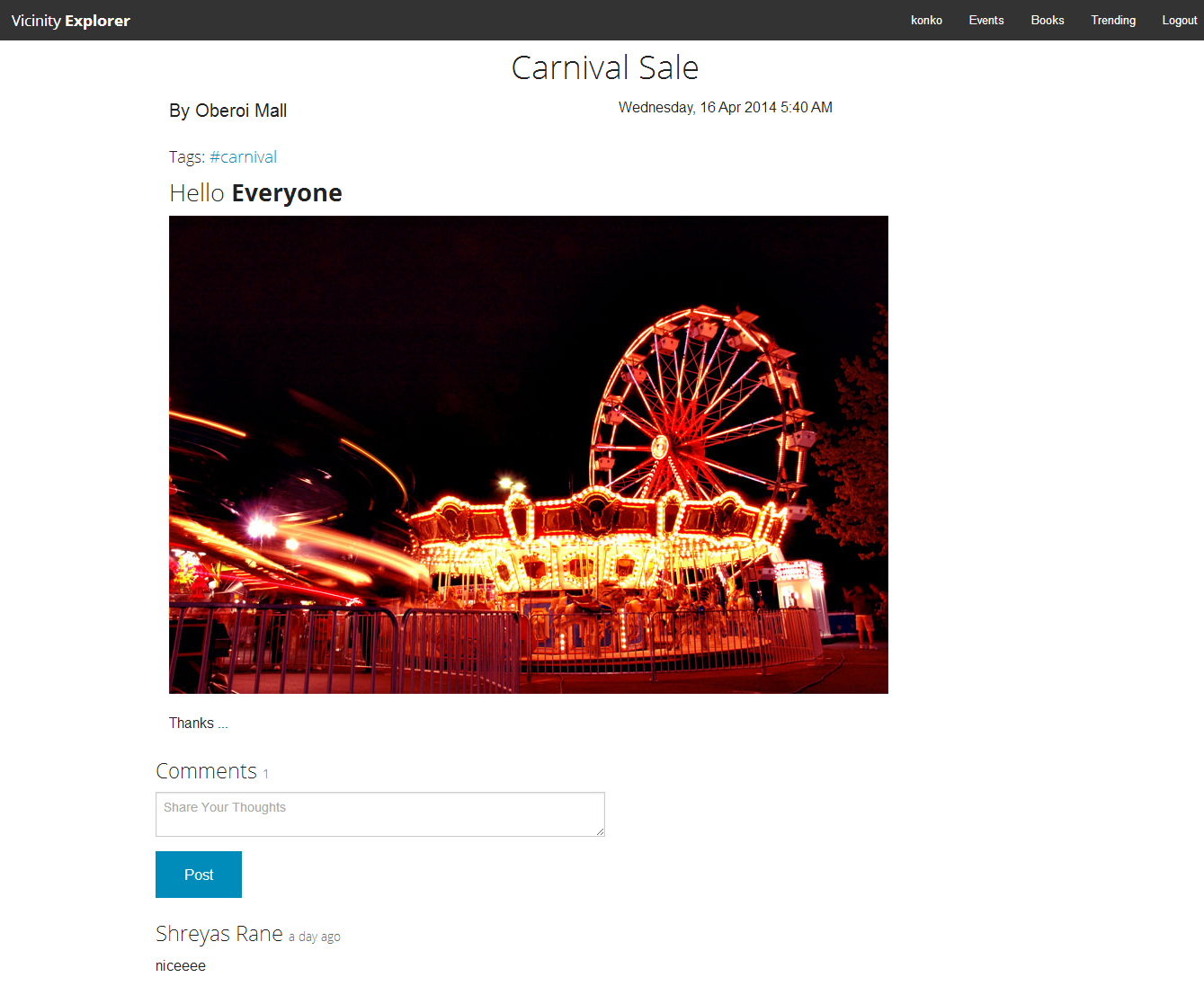
****

Fig. 5.2.4 View of Particular Post

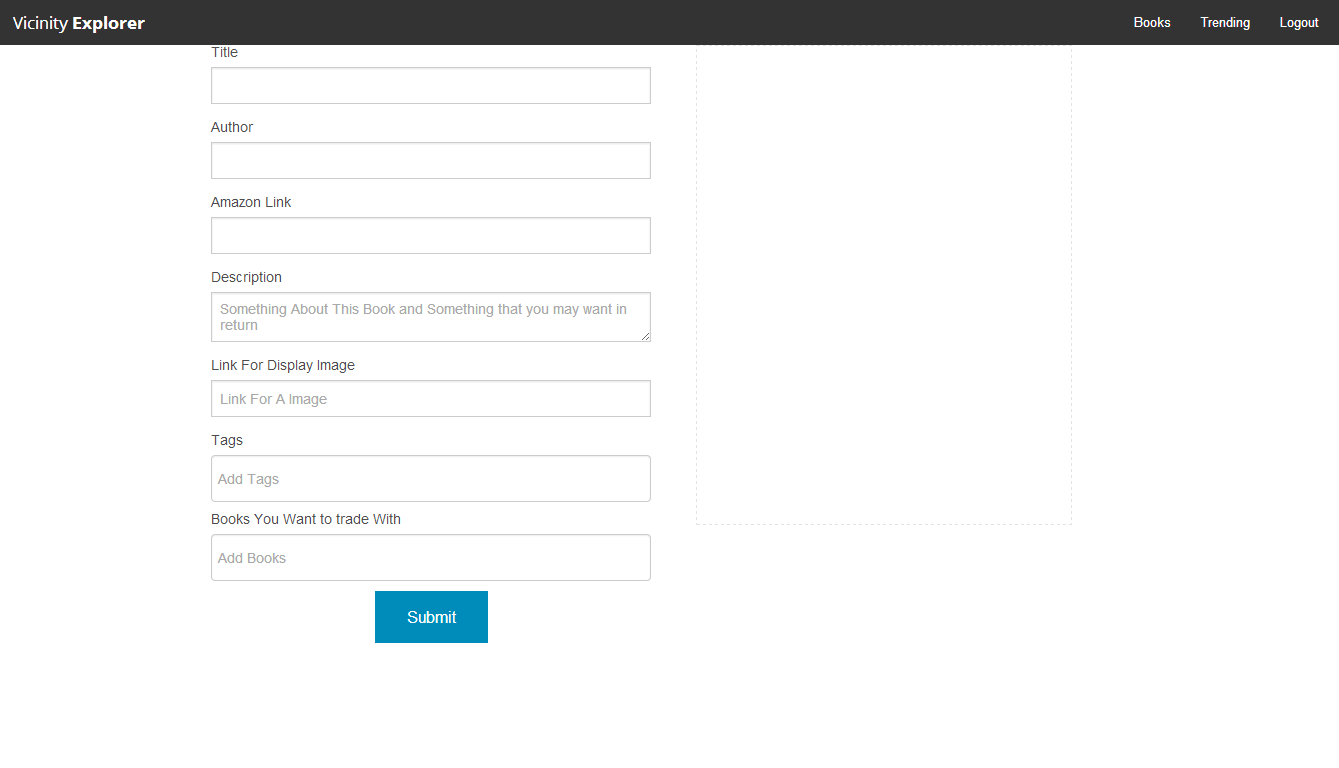
****

Fig. 5.2.4 Creation of Books for Sharing

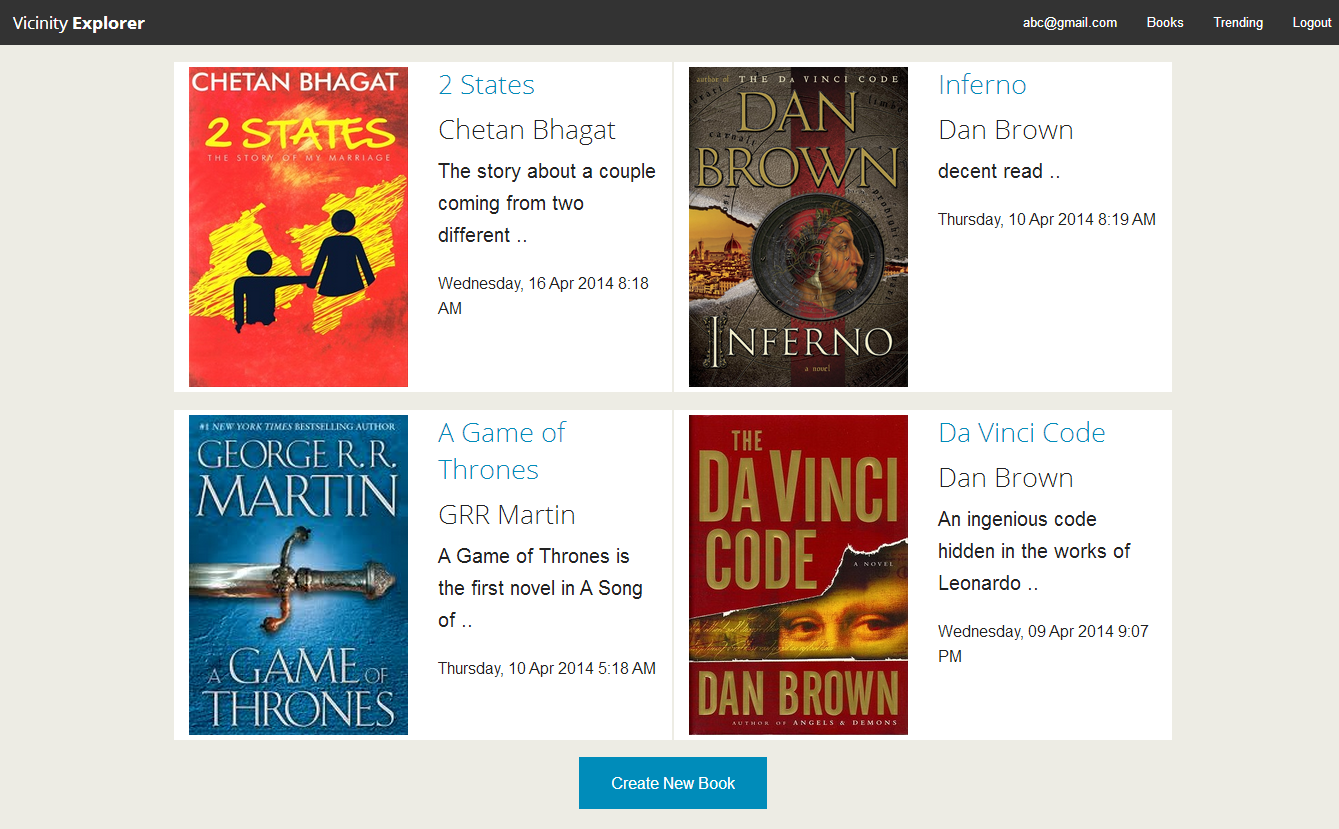
****

Fig. 5.2.4 Book Dashboard

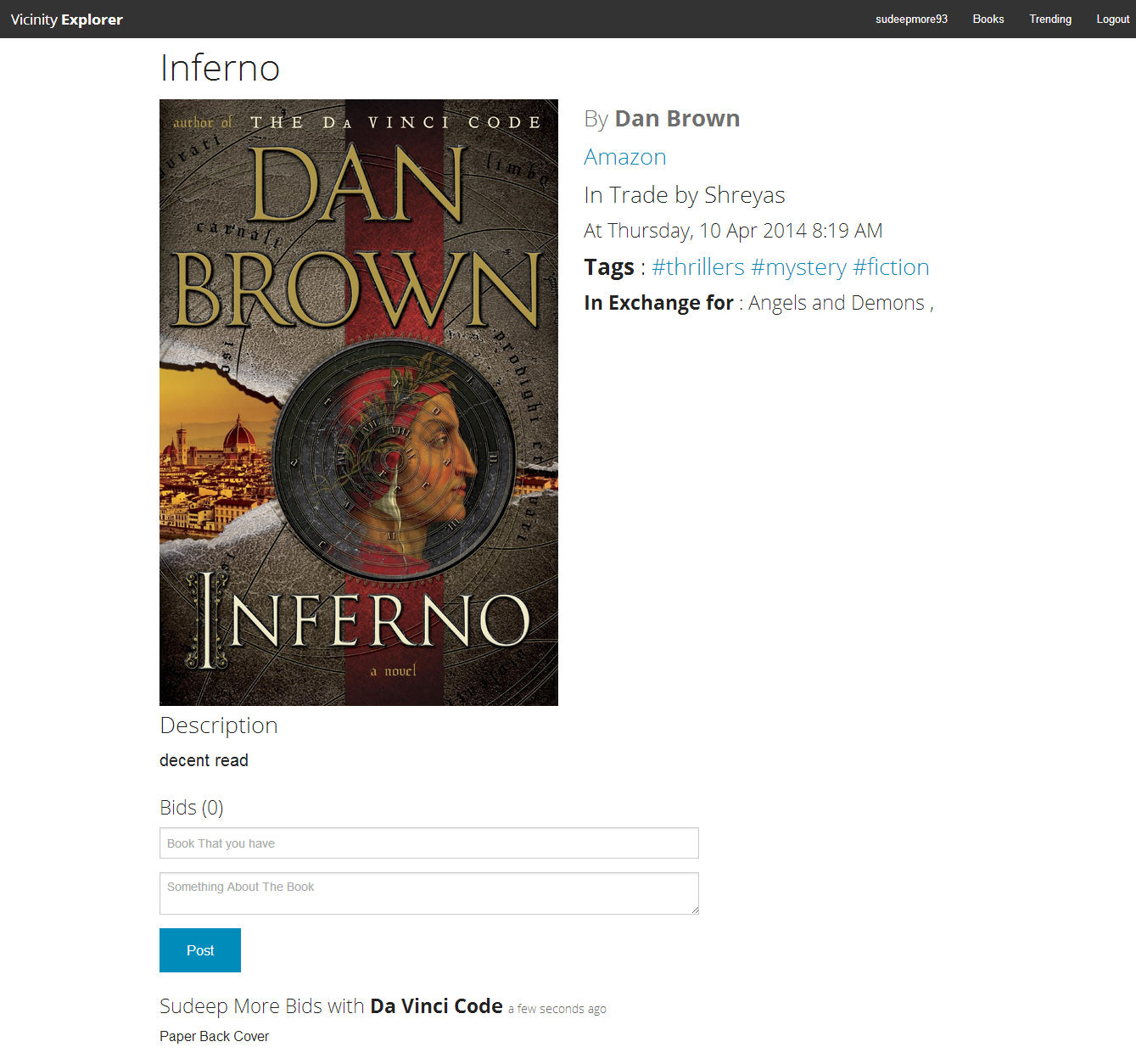
****

Fig. 5.2.4 Book with Bids

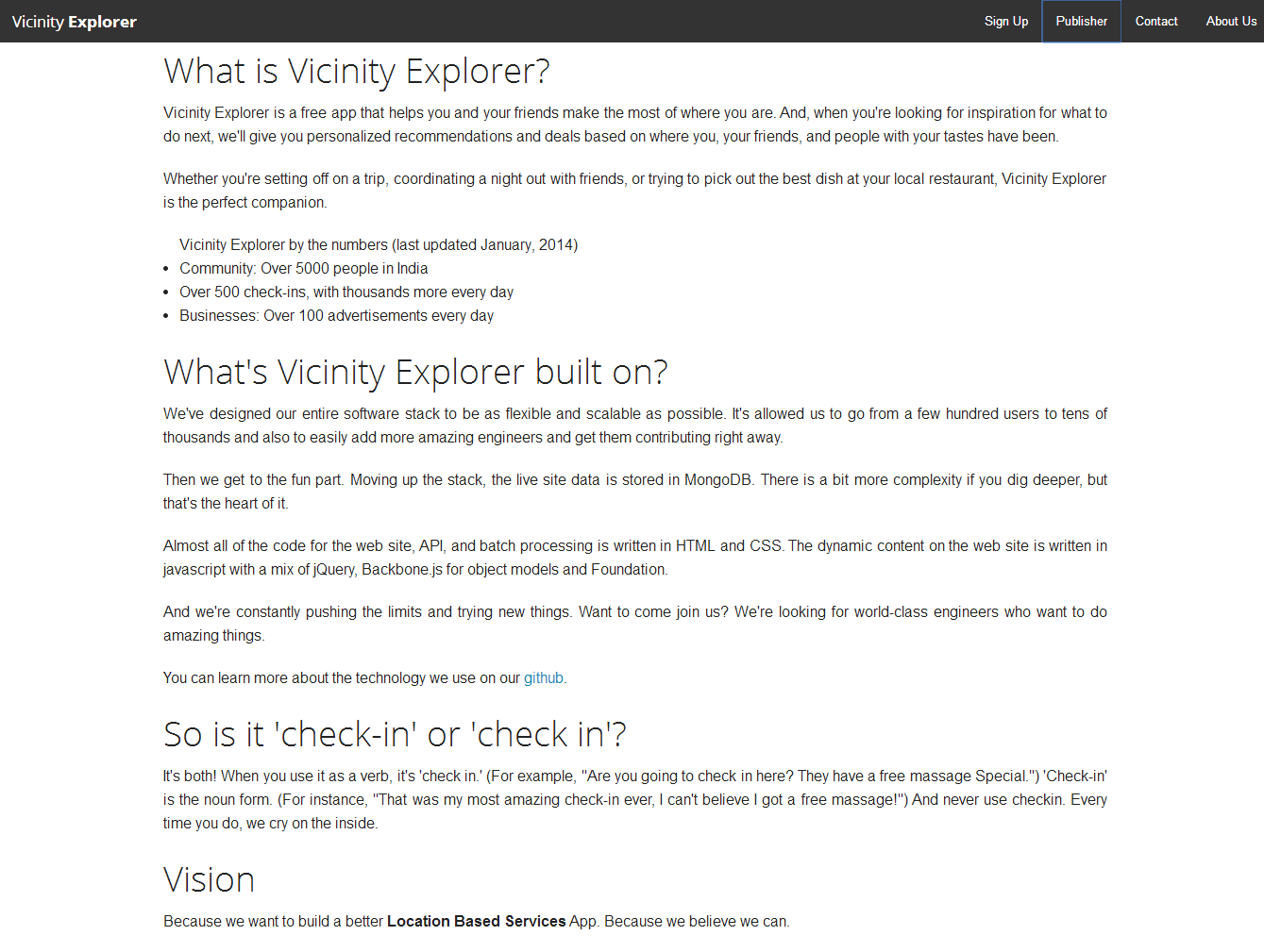
****

Fig. 5.2.4 About Us Page

**CHAPTER 6**

**6. IMPLEMENTATION:**

**6.1 Hardware and Software Used:**

**Hardware Used:**

1. Processor: Inter(R) Core(TM) i5-2450M 2.5GHz.
2. Memory: 8 GB.
3. Hard Disk: 1 TB.

**Software used:**

Client Side:

1. Web Browser: Chrome Version 30.0.1847+, Mozilla Firefox Version 5.0+, Internet Explorer 9+.
2. Foundation 5

Middleware:

1. Node.js Version 0.10.24
2. Express.js Version 3.4.7
3. Swig Version 3.0.0

Server Side:

1. Mongo DB Version 2.4.8

**6.2 Algorithms / Methods Used**

**6.3 Working of the project *(by using mentioned algorithms with code)***

**CHAPTER 7**

**7. RESULTS AND DISCUSSIONS: *(final results or outputs)***

**CHAPTER 8**

**8. TESTING:**

**8.1 Test cases:**

**1. FORM VALIDATION:**

|  |  |
| --- | --- |
| Test Name | Login Validation |
| Test Objectives | Check whether all the details entered by the user are valid. |
| Test Configuration | * Opening the webpage in the browser * Loading the required data in the database |
| Procedure | Click the login or sign-in button on the header. |
| Action: | Enter the details. |
| Expected Result | The error message is displayed for the fields which are invalid. |
| Actual Result | The error message is displayed for the fields which are invalid. |
| Pass/Fail | Pass |

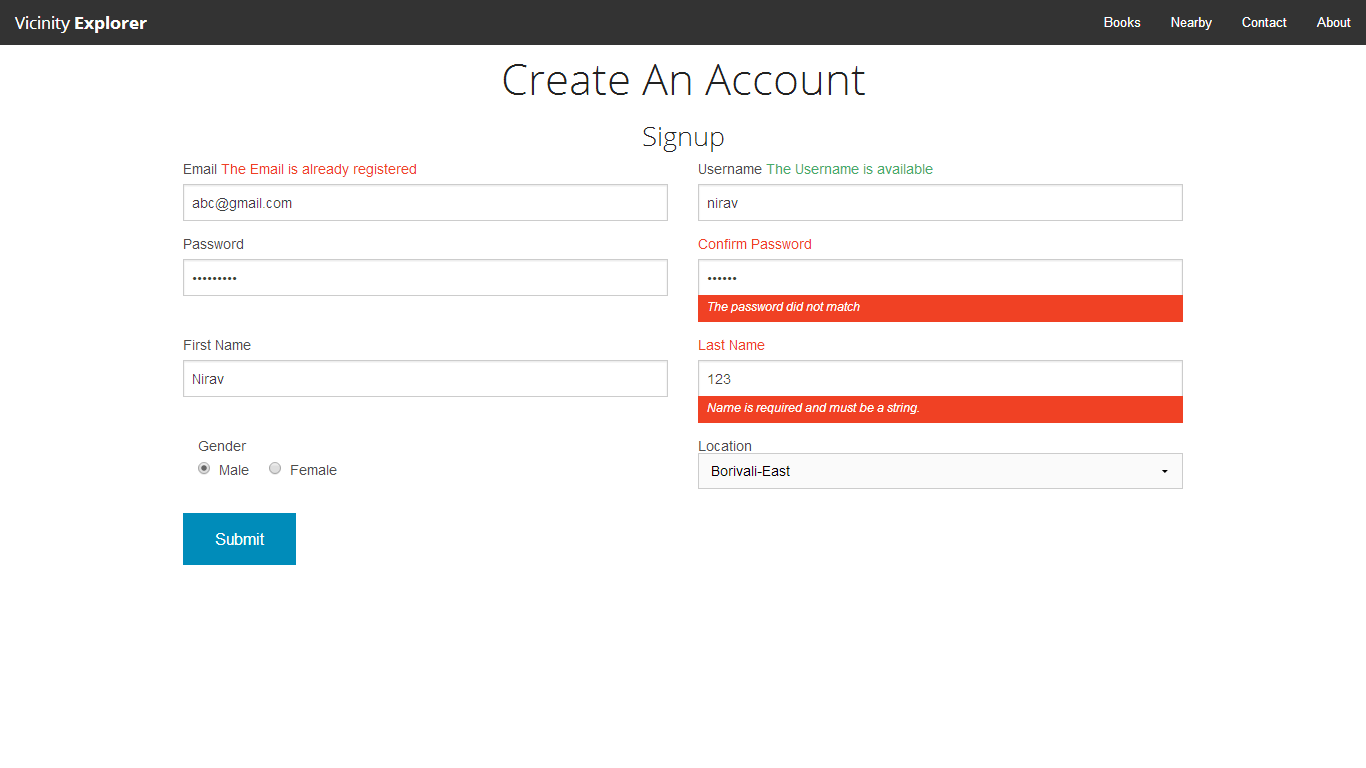


Fig. 5.3 Validating the Content

**2. POST SUBMISSION:**

|  |  |
| --- | --- |
| **Test Name** | **Post Submission** |
| Test Objectives | Check if the post is submitted and saved in the database. |
| Test Configuration | * Login as the Publisher. * Click on “Create New Post” button. |
| Procedure | Insert (Type) the Title, Contents of the post. |
| Action: | Click the submit button. |
| Expected Result | The submitted post should appear on moderator and publisher dashboard. |
| Actual Result | The post appears on moderator and publisher dashboard. |
| Pass/Fail | Pass |



Fig. 5.3 Validating the Content

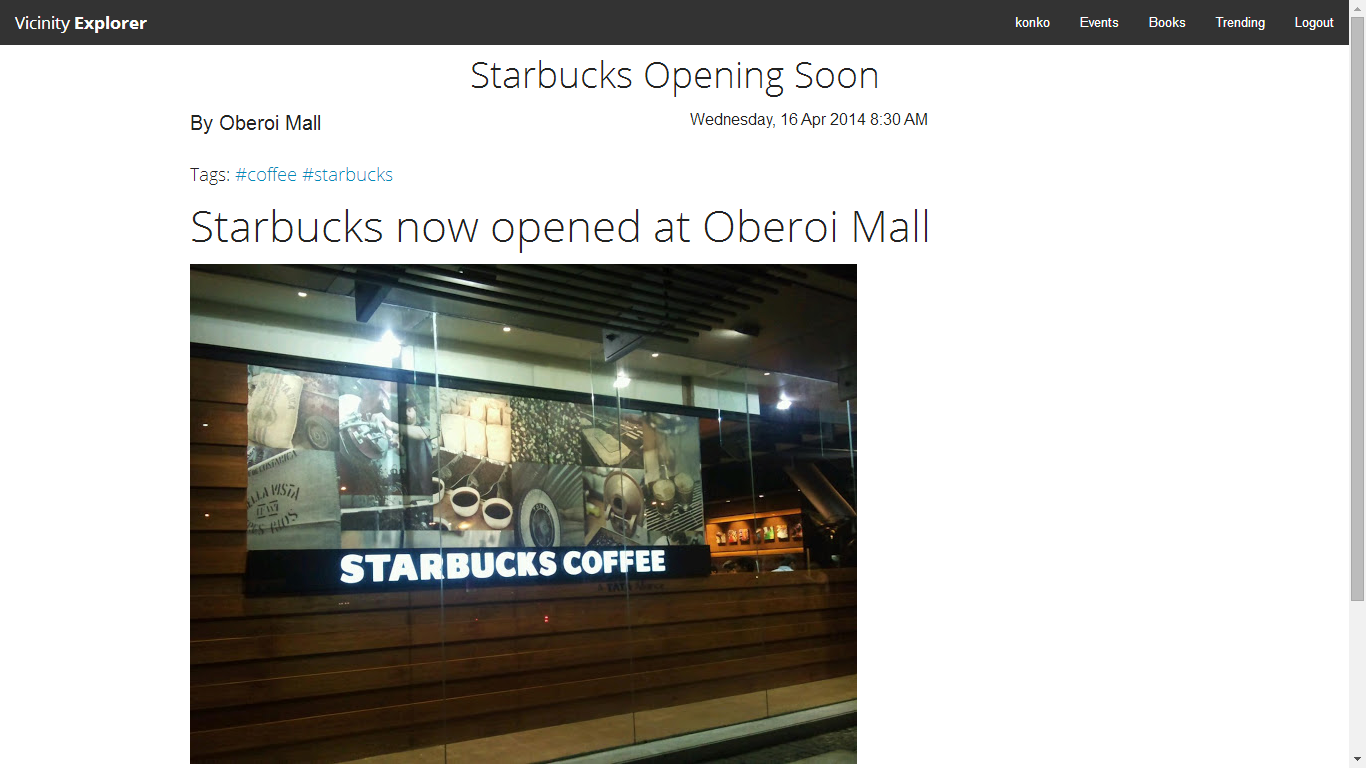


Fig. 5.3 Validating the Content

**3. BOOK SUBMISSION:**

|  |  |
| --- | --- |
| **Test Name** | **Post Submission** |
| Test Objectives | Check if the Book is submitted and saved in the database. |
| Test Configuration | * Login as the Publisher. * Click on “Create New Book” button. |
| Procedure | Insert (Type) the Title, Contents of the book. |
| Action: | Click the submit button. |
| Expected Result | The submitted post should appear on moderator and publisher dashboard. |
| Actual Result | The post appears on moderator and publisher dashboard. |
| Pass/Fail | Pass |

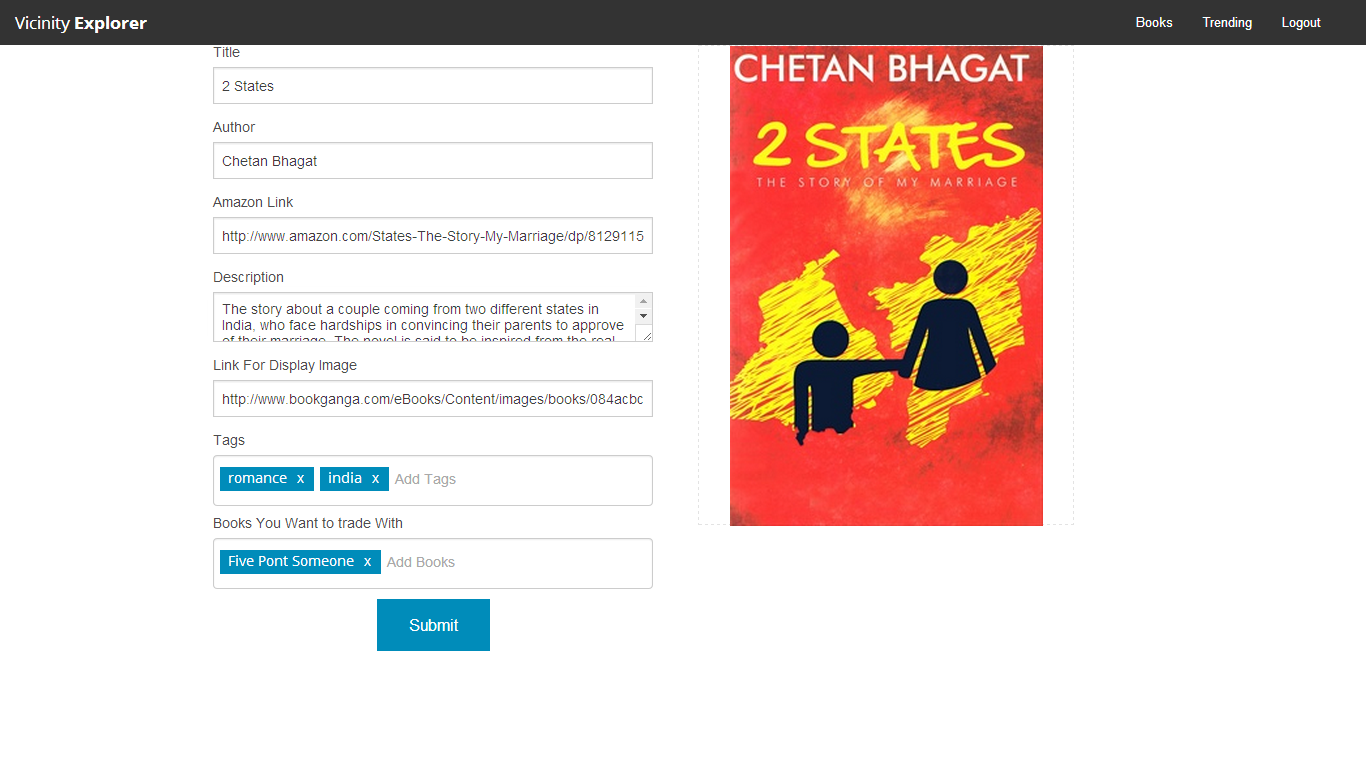


Fig. 5.3 Validating the Content

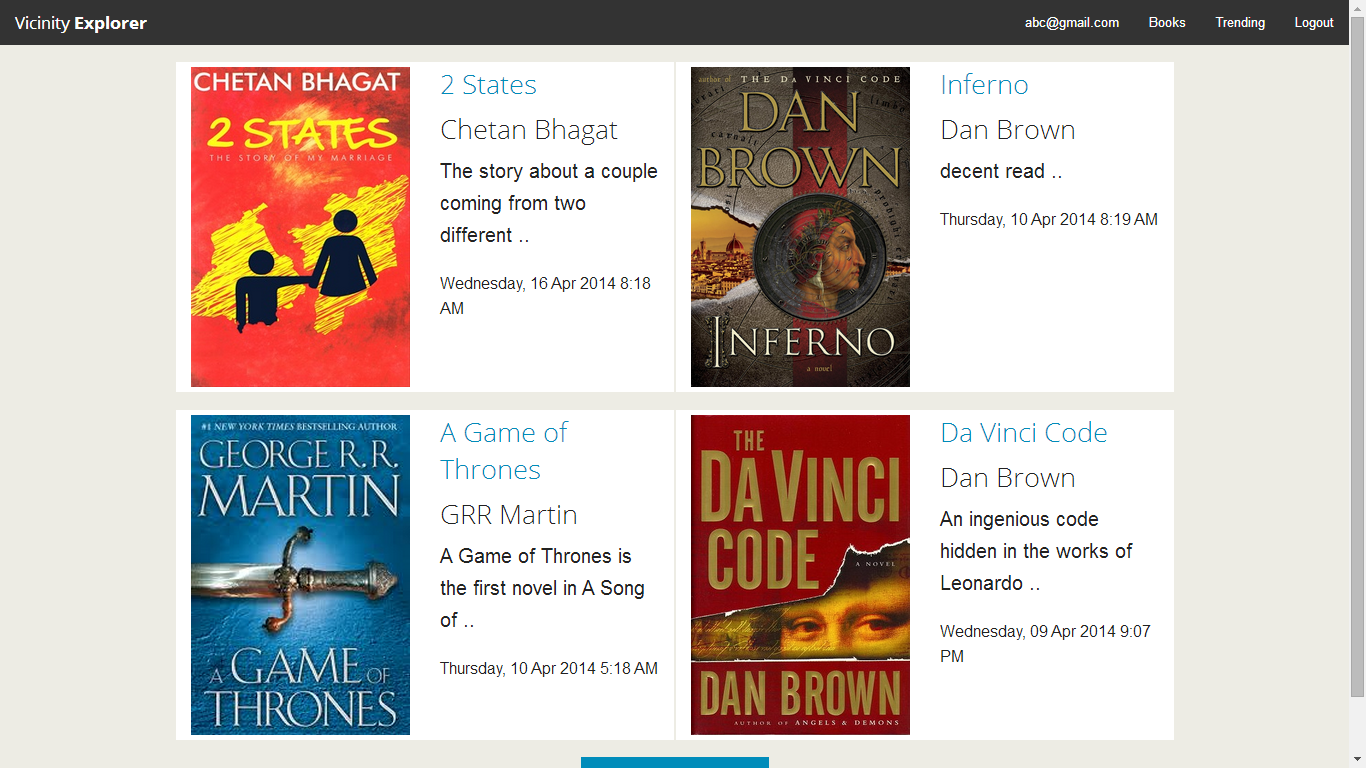


Fig. 5.3 Validating the Content

**4. POST COMMENT:**

|  |  |
| --- | --- |
| **Test Name** | **Post Comment** |
| Test Objectives | Check if the user can comment to a specific post. |
| Test Configuration | * Login as a User. * Click on the specific post to comment. |
| Procedure | Fill up the comment body and click the post button. |
| Action: | Click the Post Button |
| Expected Result | The expected comment body should appear below the post with the user’s name. |
| Actual Result | The comment appears below the post. |
| Pass/Fail | Pass |

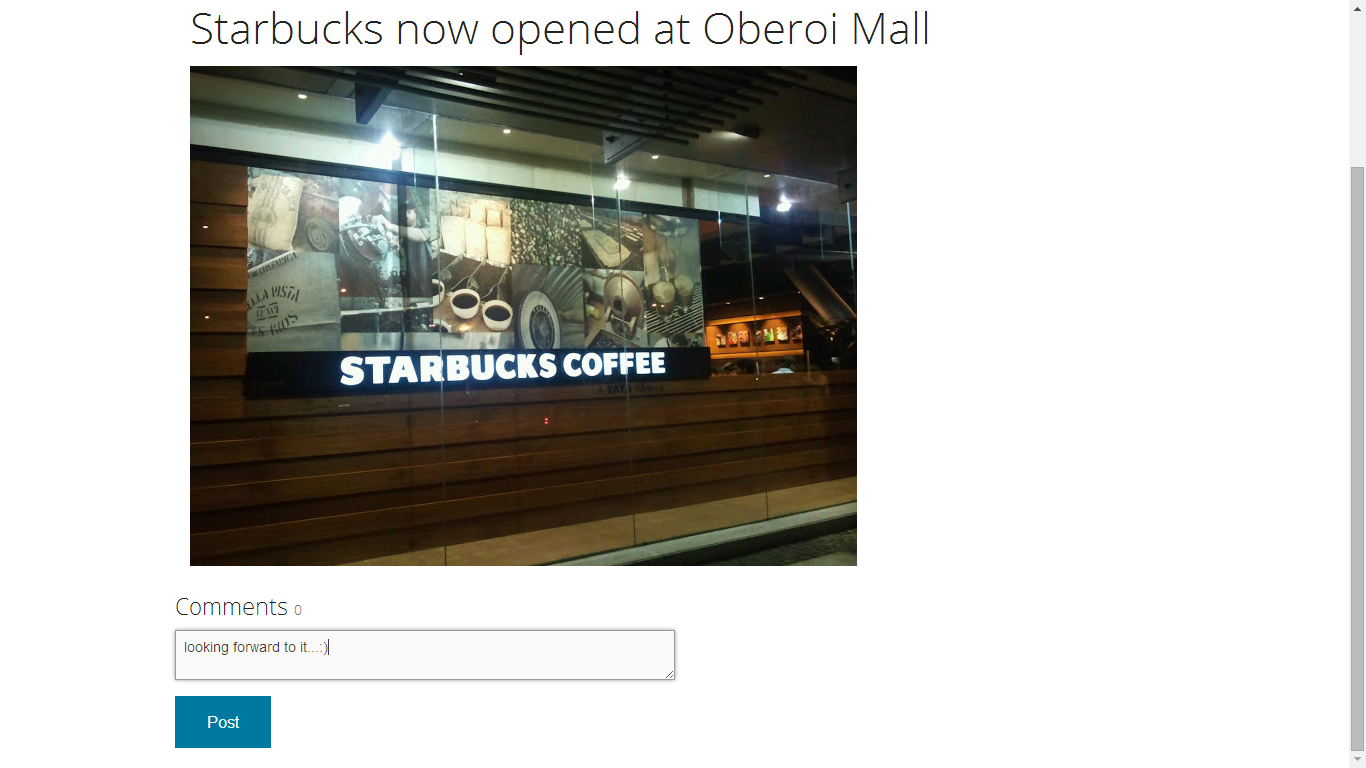


Fig. 5.3 Validating the Content

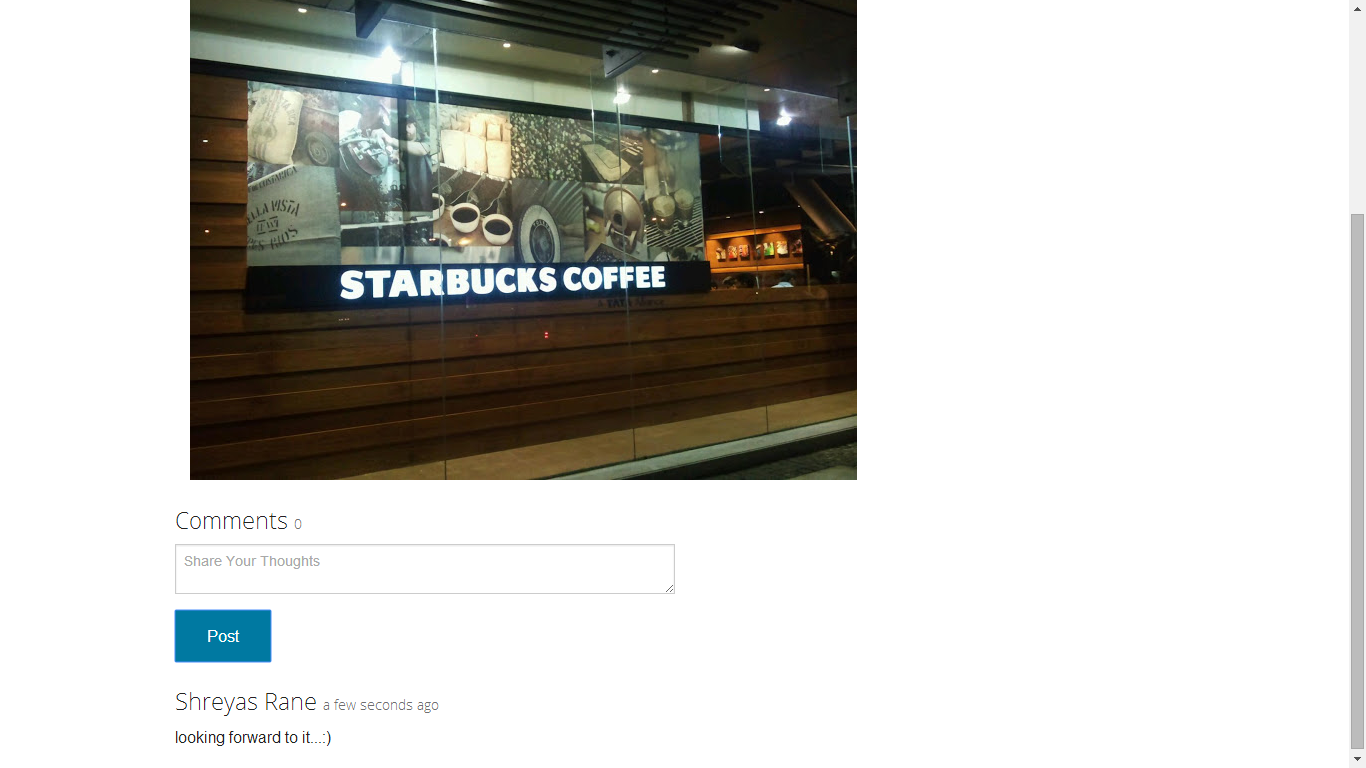


Fig. 5.3 Validating the Content

**5. POST DELETE:**

|  |  |
| --- | --- |
| **Test Name** | **Post Delete** |
| Test Objectives | Check if the post is getting deleted from the database. |
| Test Configuration | * Login as a Publisher. * Click on the specific post to be deleted. |
| Procedure | Check if the post is required for future. |
| Action: | Click the Delete Post Button. |
| Expected Result | Post should not appear on any of the dashboard. |
| Actual Result | The post gets deleted from the database. |
| Pass/Fail | Pass |

**6. POST MODERATION:**

|  |  |
| --- | --- |
| **Test Name** | **Post Moderation** |
| Test Objectives | Check if the moderator can Accept/Deny the post. |
| Test Configuration | * Login as a Moderator. * Click on the specific post for moderation. |
| Procedure | Check if the post satisfies the required specification. |
| Action: | Click on the Accept/Deny Button. |
| Expected Result | Post should not appear on user dashboard after denied. |
| Actual Result | The post doesn’t appear on the user dashboard. |
| Pass/Fail | Pass |

**8.2 Type of Testing used**

**BLACK BOX TESTING:**

Black box testing treats the system as a “black-box”, so it doesn’t explicitly use Knowledge of the internal structure or code. Or in other words the Test engineer need not know the internal working of the “Black box” or application. Main focus in black box testing is on functionality of the system as a whole. The term ‘behavioral testing’ is also used for black box testing and white box testing is also sometimes called ‘structural testing’. Behavioral test design is slightly different from black-box test design because the use of internal knowledge isn’t strictly forbidden, but it’s still discouraged. Each testing method has its own advantages and disadvantages. There are some bugs that cannot be found using only black box or only white box. Majority of the applications are tested by black box testing method. We need to cover majority of test cases so that most of the bugs will get discovered by black-box testing. Black box testing occurs throughout the software development and testing life cycle i.e. in Unit, Integration, System, Acceptance and regression testing stages.

**BLACK BOX TESTING TECHNIQUES:**

**1. Graph Based Testing Methods:**

Each and every application is buildup of some objects. All such objects are identified and graph is prepared. From this object graph each object relationship is identified and test cases written accordingly to discover the errors.

**2. Error Guessing:**

This is purely based on previous experience and judgment of tester. Error Guessing is the art of guessing where errors can be hidden. For this technique there are no specific tools, writing the test cases that cover all the application paths.

**3. Boundary Value Analysis:**

Many systems have tendency to fail on boundary. So testing boundary values of application is important. Boundary Value Analysis (BVA) is a test Functional Testing technique where the extreme boundary values are chosen. Boundary values include maximum, minimum, just inside/outside boundaries, typical values, and error values.

**Advantages of Black Box Testing are as follows:**

1. Tester can be non-technical.
2. Used to verify contradictions in actual system and the specifications.
3. Test cases can be designed as soon as the functional specifications are complete

**Disadvantages of Black Box Testing are as follows:**

1. The test inputs needs to be from large sample space.
2. It is difficult to identify all possible inputs in limited testing time. So writing test cases is slow and difficult.
3. Chances of having unidentified paths during this testing.

**WHITE BOX TESTING:**

White box testing (WBT) is also called Structural or Glass box testing. White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all internal components have been adequately exercised. White box testing is needed so that all independent paths within a module have been exercised at least once, all logical decisions verified on their true and false values, all loops executed at their boundaries and within their operational bounds internal data structures validity.

**WHITE BOX TESTING TECHNIQUES:**

**1. Basis Path Testing:**

Each independent path in the code is taken for testing.

**2. Data Flow Testing (DFT):**

In this approach you track the specific variables through each possible calculation, thus defining the set of intermediate paths through the code. DFT tends to reflect dependencies but it is mainly through sequences of data manipulation. In short each data variable is tracked and its use is verified. This approach tends to uncover bugs like variables used but not initialize, or declared but not used, and so on.

**3. Path Testing:**

Path testing is where all possible paths through the code are defined and covered. It is a time consuming task.

**4. Loop Testing:**

These strategies relate to testing single loops, concatenated loops, and nested loops. Independent and dependent code loops and values are tested by this approach.

**Advantages of White Box Testing are as follows:**

1. As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.
2. Forces test developer to reason carefully about implementation
3. It helps in removing the extra lines of code, which can bring in hidden defects.

**Disadvantages of White Box Testing are as follows:**

1. It is nearly impossible to look into every bit of code to find out hidden errors, which may create problems, resulting in failure of the application.
2. Miss cases omitted in the code.
3. Very few white-box tests can be done without modifying the program, changing values to force different execution paths, or to generate a full range of inputs to test a particular function.

**CHAPTER 9**

**9. CONCLUSION:**

**CHAPTER 10**

**10. FUTURE SCOPE:**

**Issue Clustering:**

**Government Portal:**

This is a government Portal

**Review Places:**

**Find People near You:**

**Meet ups and social Hubs:**

**Travel Guide:**

**APPENDIX**

**GLOSSARY**

**BIBILIOGRAPHY**

**REFERENCES**

1. Cain Evans and Philip Moore and Andrew Thomas, “An Intelligent Mobile Advertising System (iMAS) Location-Based Advertising to Individuals and Business”. IEEE 2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems
2. KanakBiscuitwala and Steve Fan “OmniStanford: Location-Based

Discovery and Journaling for College Students” UIST’12, October 7-10, 2012, Cambridge, MA, USA.

1. Cain Evans and Philip Moore and Andrew Thomas, “An Intelligent Mobile Advertising System (iMAS) Location-Based Advertising to Individuals and Business”. IEEE 2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems
2. KanakBiscuitwala and Steve Fan “OmniStanford: Location-Based

Discovery and Journaling for College Students” UIST’12, October 7-10, 2012, Cambridge, MA, USA.

1. Dodson, B., Vo, I., Purtell, T., Lam, M.S. “Musubi: Disintermediated Interactive Social Feeds for Mobile Devices”. In Proceedings of the 21st International Conference on World Wide Web. ACM, 2012.
2. Young-BaeKo and Nitin H. Vaidya ”Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms”

1. I.F.Akyildiz, S. M. Joseph, and Y.-B. Lin. “Movement based location update and selective paging for PCS networks”. IEEE/ACM Trans. on Networking, 4(4):94–104, 1996.
2. Aseniero, B.A., Carpendale, S., Tang, A. “Deep Personalization in Tools for Reflection” In CHI 2012 Workshop on Personal Informatics in Practice: Improving Quality of Life through Data, Austin, TX.