DYNAMIC TRAFFIC CONTROL AND MANAGEMENT SYSTEM

PROJECT REPORT

Submitted in partial fulfillment of the requirements

for the degree of

Bachelor of Technology

in

Computer Science & Engineering

Submitted by

Rohit Roushan (1605147)
Rahul Pandey (1605142)

Under the Guidance of

Prof. Anil Kumar Swain

School of Computer Engineering
Kalinga Institute of Industrial Technology
Deemed to be University, Bhubaneswar

17 APR 2020

A PROJECT REPORT

ON

DYNAMIC TRAFFIC CONTROL AND MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science & Engineering

Submitted by

Rohit Roushan
1605147

Rahul Pandey
1605142

Under the Guidance of

Prof. Anil Kumar Swain

School of Computer Engineering

Kalinga Institute of Industrial Technology
Deemed to be University, Bhubaneswar

17 APRIL 2020

School of Computer Engineering

Kalinga Institute of Industrial Technology
Deemed to be University, Bhubaneswar

CERTIFICATE

*This is to certify that the project report entitled "**Dynamic Traffic Control and Management System"** has been carried out by **Rohit Roushan (1605147), Rahul Pandey (1605142)** in partial fulfillment of the award of degree of **Bachelor of Technology in Computer Science & Engineering** from **School of Computer Engineering, Kalinga Institute of Industrial Technology, Deemed to be University, Bhubaneswar** during the academic year 2017-2018 under my supervision.*

Prof. Anil Kumar Swain
Computer Sc. & Engineering

Examiner 1

Examiner 2

Date: 17 APR, 2020

Place: KIIT Bhubaneswar

DECLARATION

We, Rahul Pandey (1605142), Rohit Roushan (1605147), hereby declare that the matter embodied in this project report is original and has not been submitted for the award of any other degree to any other university.

Rohit Roushan (1605147)

Rahul Pandey (1605142)

ACKNOWLEDGEMENTS

We would like to thank Prof. Anil Kumar Swain, School of Computer Engineering, KIIT University, Bhubaneswar for his continued drive for better quality in everything that allowed us to carry out this project work.

We would also like to take the opportunity to thank the director, School of Computer Engineering, KIIT University, Bhubaneswar for providing us with the resources and the help and co-operation in this project work.

Lastly, we would like to express our gratitude to our Parents and all the Professors, Lecturers, and friends for their co-operation, constructive criticism and valuable suggestions during the preparation and successful completion of this project report.

Rohit Roushan (1605147)
Rahul Pandey (1605142)

ABSTRACT

Automated traffic control systems have been now-a days used as an alternative to the existing traditional traffic systems.In order to generalize data and automatize human behavior,image processing combined with machine learning(ml) has been largely used in various research field.

The paper focuses on the model that sends sms /mail to the person breaking the traffic light rules.This system further adapts the traffic signal timer according to the random traffic density using image processing(OpenCV) and machine learning algorithms.To validate the model prototype, video data sets are used to test the model based on background subtraction filter (BackgroundSubtractorMOG2) and Convolutional neural network.Results show 100% precision rate and recall of 65%.

KEYWORDS:

- ✓ Automated traffic light detection
- ✓ Computer vision
- ✓ Image processing
- ✓ Machine learning
- ✓ Convolutional Neural Network

Contents

Declaration……... ii

Acknowledgments…… iii

Abstract ……...…… iv

List of figures …….. vii

1 Introduction ....……... 1

1.1 About Project..…… 1

1.2 Background of the project ……..…… 1

1.3 Modules...……... 2

1.4 Definitions, Acronyms and Abbreviations……. 2

2 Literature Survey ……. 3

3 Software Requirement Analysis…….…….. 4

3.1 Software Requirement ……..……...… 4

3.2 Hardware Requirement……... 4

3.3 Product Perspective …… 5

3.4 Product Feature ……..…… 5

3.5 Assumptions …….. 6

4 Requirement Analysis ……. 7

4.1 Functional Requirements ……. **7**

4.2 Non-Functional Requirements ……. 10

5 Technological Stack.…….. 11

5.1 Python ……..... 11

5.2 OpenCV……..….. ……..…….. 12

5.3 Yolo . …...…….…….…….…….. 13

5.4 Convolutional Neural Network (CNN)…….. 14

5.5 MySQL ……. 15
5.6 WAY2SMS …….. 16
5.7 Docker ……..……. 17

6 System Design……...…….. 18
6.1 Control Flow Graph ……..……. 18
6.2 Sequence Diagram…….. 19

7 System Testing ……... 20

8 Project Planning……. 21

9 Implementation ……...……...……... 22

10 Screenshots ...……. 24
10.1 Screenshots of project implementation …… **24**
10.2 Screenshots of project codes …… **29**

11 Conclusion …….. 43

12 Future Scope …….. 44

13 References …….. 45

14 Appendix ……. 46

List of Figures

6.1 State Chart Graph ……..……. 18

6.2 Sequence Diagram……..…….. 19

10.1 Screenshots of project implementation 24

10.1.1 Implementation of Sig.py 24

10.1.2 Implementation of Yolo.py 25

10.1.3 Implementation of Capture.py 26

10.1.4 Implementation of Plate_segment.py 27

10.1.5 Implementation of Licence_no.py 28

10.2 Screenshots of project codes 29

10.2.1 Project codes of Sig.py 29

10.2.2 Project codes of Yolo.py 30

10.2.3 Project codes of Counter.py 32

10.2.4 Project codes of Motion.py 33

10.2.5 Project codes of Capture_car.py 35

10.2.6 Project codes of Plate_segment.py 37

10.2.7 Project codes of Licence_no.py 39

10.2.8 Project codes of Mail.py 40

Chapter 1.

INTRODUCTION

1.1 About Project

Internpreneur – An Online Job portal for the Students and Entrepreneurs is a web application built in PHP and MySQL. It provides fresher students having no prior Work Experience, ability to register to this application and search for internships, manage their projects and accounts. Each candidate will have an account with their own home page.

On the other hand, Entrepreneurs those are willing to publish the Internships for their company to students, can register to the portal and get their own account created and can post Internships to portal's database. Registered Entrepreneurs can add or remove internships and these internships can be seen by various students and they can contact the concern person for the job.

Main aim of this web application is to make a user friendly platform where, students can search internships easily and is accessible to everyone who are interested.

1.2 Purpose

The Purpose of the application is to provide Internships to College Students who do not have prior work experience, to submit their Resumes and apply for Internships, where Entrepreneurs can select best Candidates from available candidate profile.

1.3 Modules

The main users in the project can be categorized into three modules as follow:

1. Admin
2. Entrepreneurs
3. Students

1.4 Definitions, Acronyms and Abbreviations

1. Admin – The authorized user who controls all the network.
2. Entrepreneurs – The user who has a company / business and creates vacancies and provides Internships.
3. Students – The user who applies for Internships.

Chapter – 2

REQUIREMENT ANALYSIS

2.1 Functional Requirements

In this section we will briefly discuss different functional requirements of different users.

2.1.1 Admin

1. Manage Students and Entrepreneurs profiles.
2. Add new Services, Related to Students or Entrepreneurs.
3. Provide Roles and Authentication to Users.
4. Ability to change the look and feel of the site by changing template.
5. Most Recent Internship positions to be displayed in homepage.
6. Search Internship Offers based on keywords, categories, locations etc.
7. Location management console to add/edit/remove locations for which offers can be posted.
8. Admin can Add/Edit/Delete/Approve/Disapprove any offer.
9. Search Students and Entrepreneurs on the basis of Keyword, name, Email-address, Country, etc.
10. Setup multiple admin with file access privileges

2.1.2 Entrepreneur

1. Ability to Search resume by keywords, category, days since posted.
2. Post Offers with detailed description (Title, Company name, Company URL, Location, Stipend, Work type, Description, Required skills...)
3. Ability to create and edit a company profile.
4. Ability to add a Company Logo.
5. Retrieve passwords reset by automated email.
6. Entrepreneur Registration
7. Information on number of vacancy viewings and applications
8. Option of including company profile and logo.
9. Update profile at any time.
10. View Submitted Student Resume.
11. Download Resume or application submitted by student.

2.1.3 Students

1. Apply for an Internship at the click of a button.
2. Resume Posting.
3. Student Registration.
4. Login and update profile, including forgotten password reminders.
5. View Application History from profile.
6. Powerful online job search tool.
7. Search offers by location, qualification, categories, company, location, experience and keywords.
8. Add multiple list of qualification and experience.
9. Ability to add and submit Photos to profile.
10. Ability to add Cover Letter when apply for job.

2.2 Non-Functional Requirements

1. Secure access of confidential data (user's details). SSL can be used.
2. 24 X 7 availability.
3. Better component design to get better performance at peak time.
4. Flexible service based architecture will be highly desirable for future extension

2.3 Hardware Requirements

- Processor : Intel Pentium And Above Version.
- Speed : 1.60 GHz
- Hard Disk : 10GB And Above
- Key Board : Standard

2.4 Software Requirements

- Operating System : Windows 7 or above.
- Data Base : MYSQL, PHPMYADMIN
- Tools : HTML, CSS, PHP, JQUERY
- Server : WAMP, LAMP, XAMP or any Linux Hosting Server

Chapter 3

TECHNOLOGIES STACK

The server side language used in designing this project is PHP (Hypertext Preprocessor). Along with MySQL to develop the databases. HTML, jQuery and CSS were mainly used for the User Interface.

3.1 PHP (HYPERTEXT PREPROCESSOR)

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP stands for Hypertext Preprocessor.

Its flexibility and relatively small learning curve (especially for programmers who have a background in C, Java, or Perl) make it one of the most popular scripting languages around.

PHP code may be embedded into HTML code, or it can be used in combination with various Web template systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications.

PHP's popularity continues to increase as businesses, and individuals everywhere embrace it as an alternative to Microsoft's ASP language and realize that PHP's benefits most certainly outweigh the costs. According to Netcraft, PHP code can now be found in approximately 16 million Web sites.

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

3.2 MYSQL

MySQL is the database construct that enables PHP and Web Server to work together to access and display data in a readable format to a browser. It is a Structured Query Language server designed for heavy loads and processing of complex queries. As a relational database system, MySQL allows many different tables to be joined together for maximum efficiency and speed.

The popular features of this program are as follows:

- Multi-platform operation
- Numerous column types cover virtually every type of data
- Group functions for mathematical calculations and sorting
- Commands that allow information about the databases to be easily and succinctly shown to the administrator
- Function names that do not affect table or column names
- A password and user verification system for added security
- Up to 32 indexes per table permitted; this feature has been successfully implemented at levels of 60,000 tables and 5,000,000,000 rows.

MySQL is the perfect choice for providing data via the Internet because of its ability to handle heavy loads and its advanced security measures.

3.3 HTML

Hyper Text Markup Language (HTML) is the main mark-up language for creating web pages and other information that can be displayed in a web browser.

HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags, known as empty elements, are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tags, comments and other types of text-based content. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behaviour of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material.

```
<!-- Bar -->
<div id="bar"> </div>
<!-- /Bar -->
<!-- Header -->
<div id="header"> </div>
<!-- /Header -->
<!-- Search -->
<div id="search"> </div>
<!-- /Search -->
<!-- Content -->
<div id="content">
  <div class="inner">
    <!-- Content Inner -->
    <div class="content-inner">
      <!-- Content Center -->
      <div class="content-center frontpage"> </div>
      <!-- /Content Center -->

      <!-- Content Right -->
      <div class="content-right"> </div>
      <!-- /Content Right -->
    </div>
    <!-- /Content Inner -->
  </div>
</div>
<!-- /Content -->

<!-- Footer -->
<div id="footer"> </div>
<!-- /Footer -->

<!-- Copyright -->
<div id="copyright"> </div>
<!-- /Copyright -->

```

3.4 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a mark-up language. It's most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.

CSS is designed primarily to enable the separation of document content (written in HTML or a similar mark-up language) from document presentation, including elements such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for table-less web design). It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed.

3.5 JQUERY

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. jQuery is the most popular JavaScript library in use today, with installation on 65% of the top 10 million highest-trafficked sites on the Web. It is free, open-source software licensed under the MIT License.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and web applications.

The set of jQuery core features—DOM element selections, traversal and manipulation—enabled by its selector engine, created a new "programming style", fusing algorithms and DOM data structures. This style influenced the architecture of other JavaScript frameworks like YUI v3 and Dojo, later stimulating the creation of the standard Selectors API.

3.6 WAMP SERVER

WAMPSERVER is a Windows Web development environment for Apache, MySQL databases and PHP.

WAMP is an archetypal model of web service solution stacks, named as an acronym of the names of its original four open-source components: the Windows operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language. The WAMP components are largely interchangeable and not limited to the original selection. As a solution stack, WAMP is suitable for building dynamic web sites and web applications.

Since its creation, the WAMP model has been adapted to other componentry, though typically consisting of free and open-source software.

3.7 MD5 ENCRYPTION

The MD5 message-digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.

Description :

```
string md5 ( string $str [, bool $raw_output = false ] )
```

Parameters :

str

The string.

raw_output

If the optional raw_output is set to TRUE, then the md5 digest is instead returned in raw binary format with a length of 16.

Return Values :

Returns the hash as a 32-character hexadecimal number.

Chapter 4

SYSTEM DESIGN

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Object-oriented analysis and design (OOAD) methods are becoming the most widely used methods for computer system design. The UML has become the standard language used in Object-oriented analysis and design. It is widely used for modeling software systems and is increasingly used for high designing non-software systems and organizations.

4.1 Use Case Diagram

- The use case concept was introduced by Ivar Jacobson in the object oriented software engineering (OOSE) method.
- A use-case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and the use cases, and generalization among the cases.
- Use case diagrams show how users interact with the system.
- Use case diagrams describe what a system does from the standpoint of an external observer. The emphasis is on what a system does rather than how.
- Used during requirements elicitation to represent external behaviour.
- Actors represent roles, that is, a type of user of the system.
- Use cases represent a sequence of interaction for a type of functionality.
- The use case model is the set of all use cases. It is a complete description of the functionality of the system and its environment.

Actors

- An actor models an external entity which communicates with the system
- User
- External system
- Physical environment
- An actor has a unique name and an optional description.
- Examples:
    - Passenger: A person in the train
    - GPS satellite: Provides the system with GPS coordinates

Use Cases

- A use case represents a class of functionality provided by the as an event flow.

A use case consists of :

Actors:

1. Admin
2. Student
3. Entrepreneur

Admin Use Cases

1. Add / View Internships
2. Add Location
3. Delete Student / Entrepreneur
4. View Student / Entrepreneur Profile
5. Approve Internship / Company

Student Use Cases

1. Create Profile
2. Upload Resume
3. Apply For Internship
4. Job Alert
5. Latest Internship Offers
6. Search Internship Offers

Entrepreneur Use cases

1. Add Company
2. Post A Job / Internship Offer
3. Add Department
4. View Companies
5. Search Student
6. Check Applied Resumes

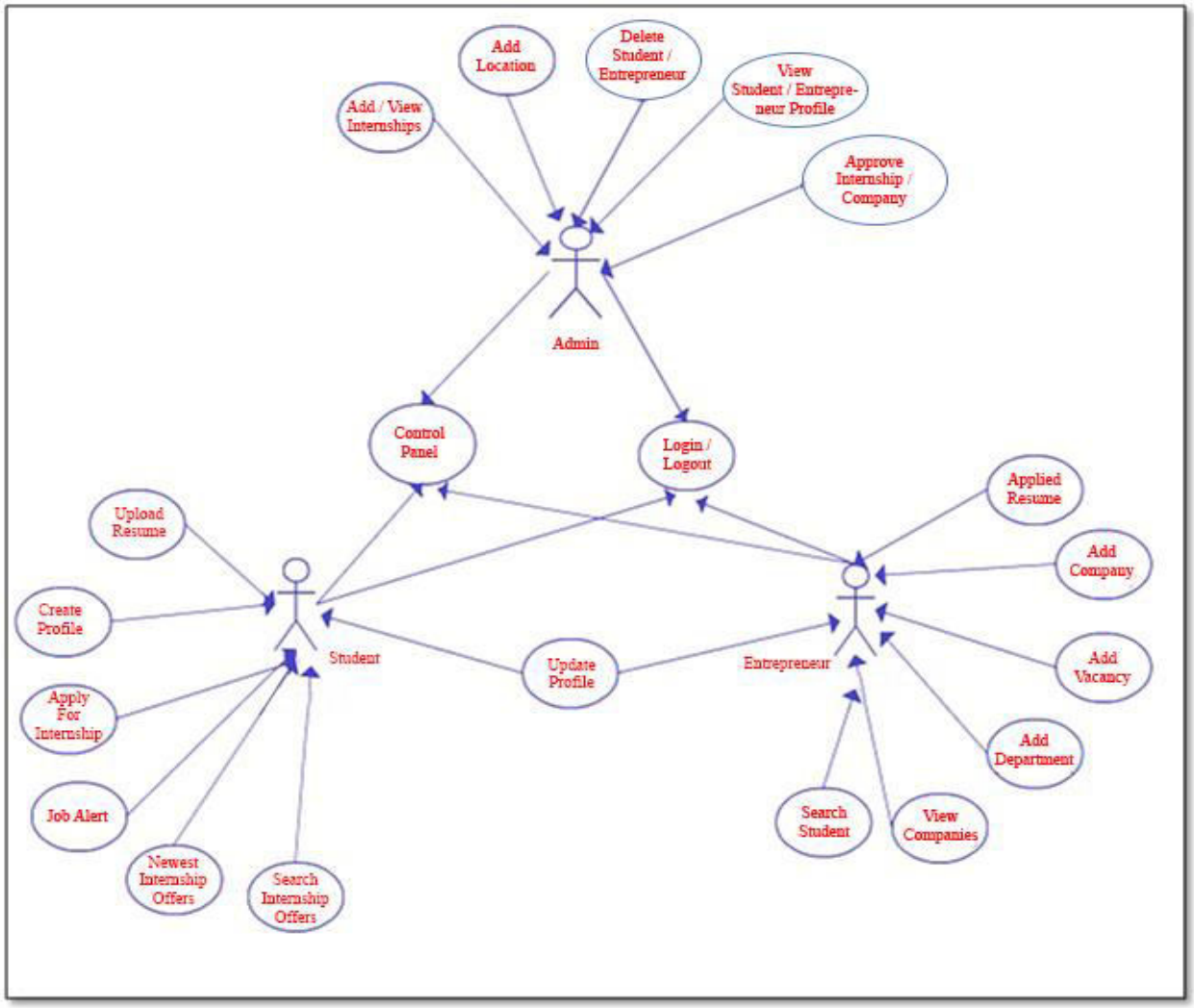

Figure 4.1

4.2 Sequence Diagram

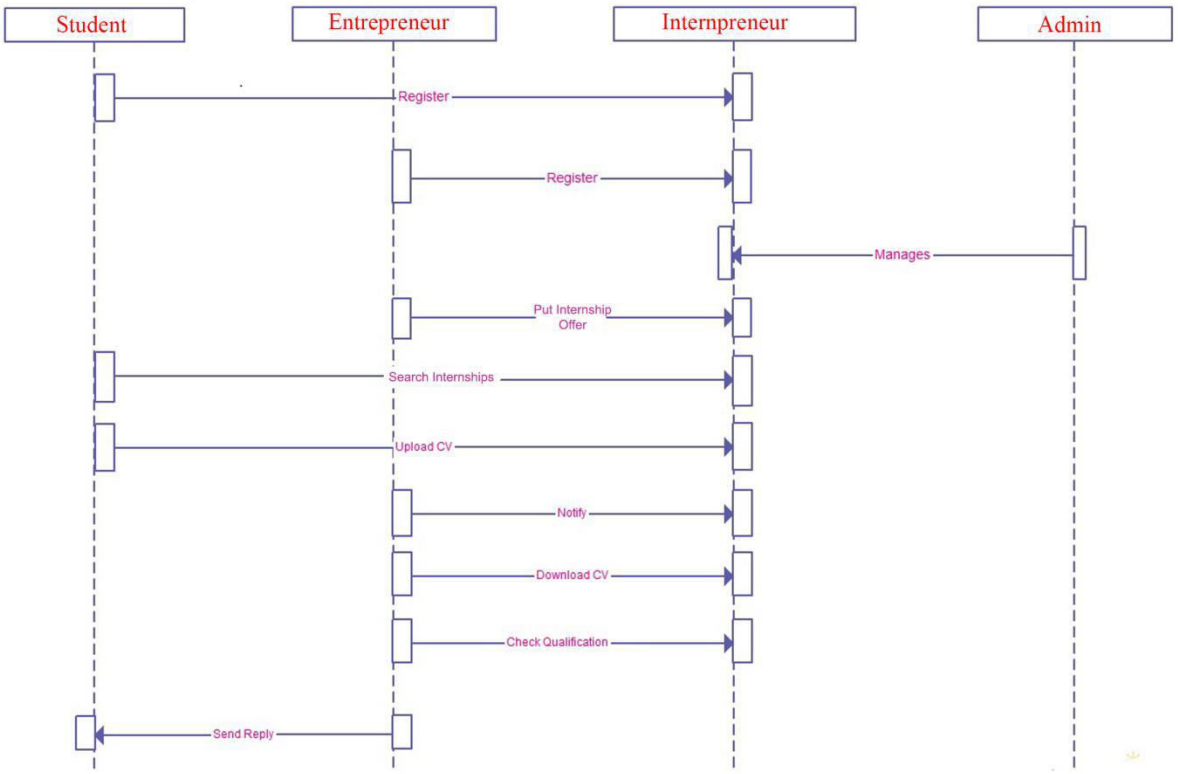

Figure 4.2

4.3 Data Flow Diagram

4.3.1 Level 0

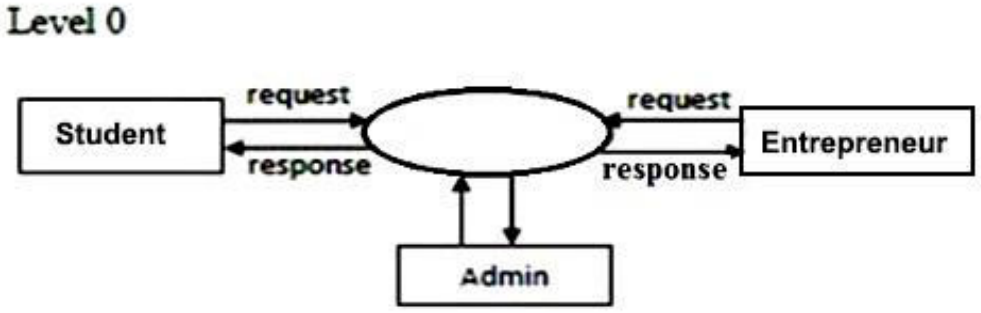

Figure 4.3.1

4.3.2 Level 1

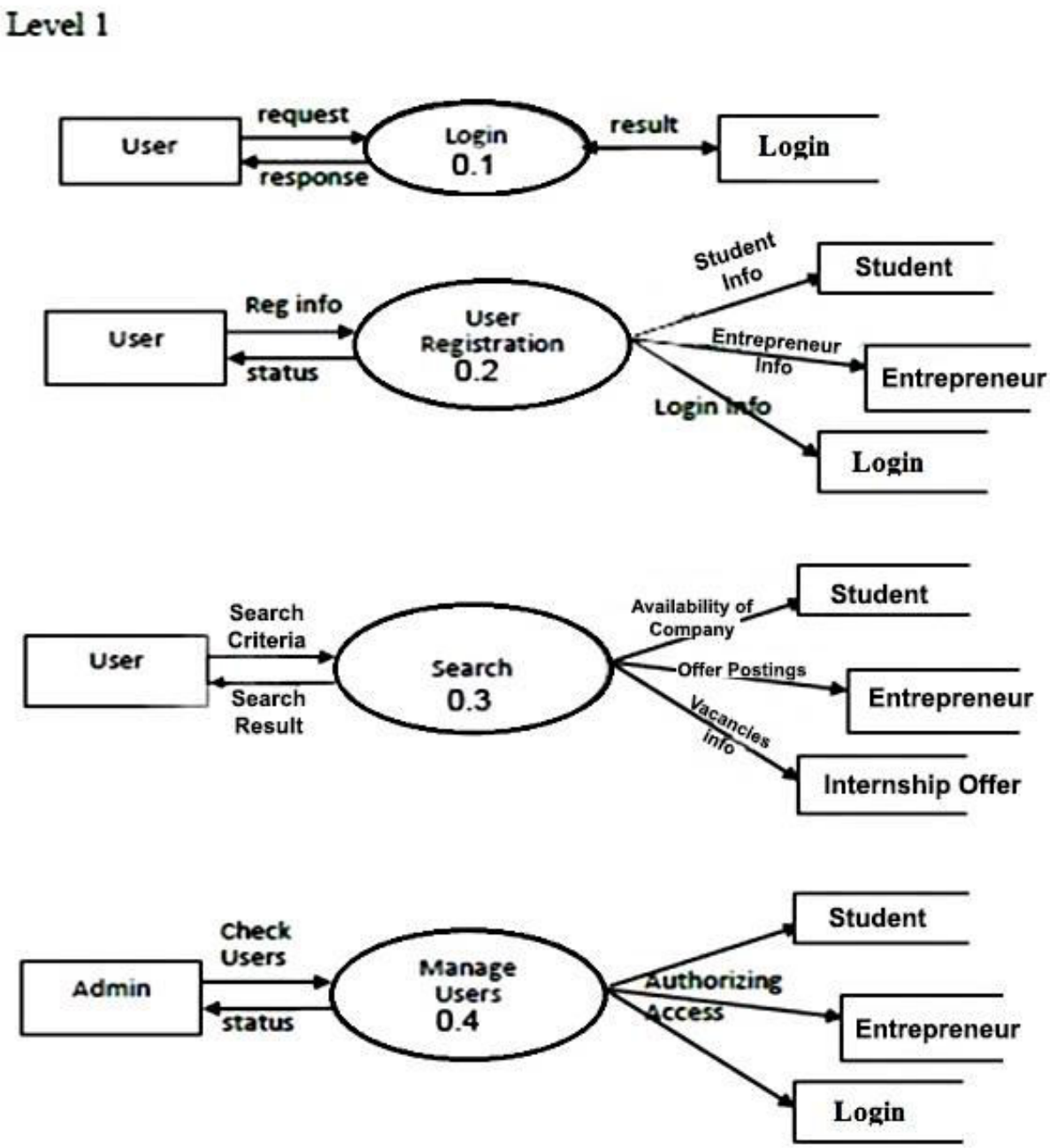

Figure 4.3.2

4.3.3 Level 2

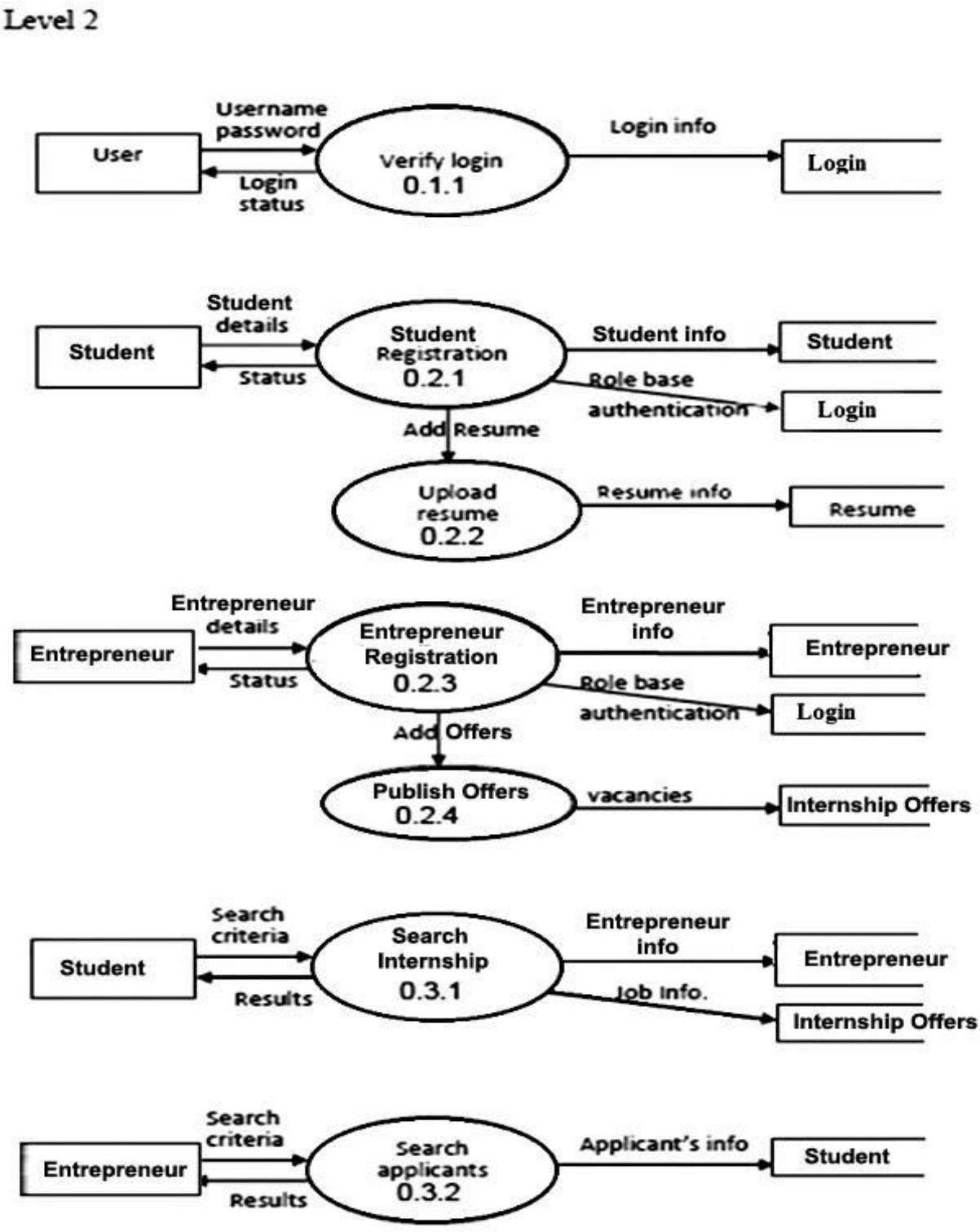

Figure 4.3.3

Chapter 5

DATABASE

5.1 Database Tables

5.1.1 in_admin

Column	Type	Null	Default
admin_id	bigint(20)	No	
sb_admin_name	varchar(255)	Yes	*NULL*
sb_pwd	varchar(255)	Yes	*NULL*
sb_email	varchar(255)	No	

5.1.2 in_applications

Column	Type	Null	Default
app_id	bigint(20)	No	
job_id	bigint(20)	No	0
resume_id	bigint(20)	No	0
cover_id	bigint(20)	No	0
sb_cand_id	bigint(20)	No	0
sbtemp	timestamp	No	CURRENT_TIMESTAMP
sb_applied_on	timestamp	No	0000-00-00 00:00:00

5.1.3 in_candidate

Column	Type	Null	Default
sb_cand_id	int(100)	No	
username	varchar(255)	No	
password	varchar(255)	No	
firstname	varchar(255)	No	
lastname	varchar(255)	No	
email	varchar(255)	No	
dob	date	No	
address	varchar(255)	No	
city	varchar(255)	No	
mobile	bigint(255)	No	
suspended	tinyint(2)	Yes	*NULL*

5.1.4 in_categories

Column	Type	Null	Default
cat_id	bigint(20)	No	
sb_cat_name	varchar(255)	Yes	*NULL*

5.1.5 in_companies

Column	Type	Null	Default
sb_company_id	bigint(20)	No	
sb_entrp_id	bigint(20)	No	0
sb_name	varchar(255)	Yes	*NULL*
sb_type	bigint(20)	Yes	*NULL*
sb_location	varchar(255)	Yes	*NULL*
sb_industry	bigint(20)	Yes	0
sb_no_of_emps	varchar(255)	Yes	*NULL*
sb_no_of_officies	varchar(255)	Yes	*NULL*
sb_profile	longtext	Yes	*NULL*
sb_logo	varchar(255)	Yes	*NULL*
sb_website	varchar(255)	Yes	*NULL*
sb_viewed	bigint(20)	No	0
sb_approved	varchar(255)	Yes	no
tempdate	timestamp	No	CURRENT_TIMESTAMP
sb_posted_on	timestamp	No	0000-00-00 00:00:00
sb_show_profile	varchar(255)	No	no

5.1.6 in_cover_letters

Column	Type	Null	Default
cover_id	bigint(20)	No	
sb_cand_id	bigint(20)	Yes	*NULL*
sb_title	varchar(255)	Yes	*NULL*
sb_contents	longtext	Yes	*NULL*
sb_approved	varchar(255)	Yes	*NULL*

5.1.7 in_education

Column	Type	Null	Default
edu_id	bigint(20)	No	
resume_id	bigint(20)	No	0
sb_school	varchar(255)	Yes	*NULL*
sb_end_month	int(11)	Yes	*NULL*
sb_end_year	int(11)	Yes	*NULL*
sb_degree	varchar(255)	Yes	*NULL*
sb_city	varchar(255)	Yes	*NULL*
sb_state	varchar(255)	Yes	*NULL*
sb_country	bigint(20)	Yes	*NULL*
sb_desc	longtext	Yes	*NULL*

5.1.8 in_entrepreneur

Column	Type	Null	Default
sb_entrp_id	int(100)	No	
username	varchar(255)	No	
password	varchar(255)	No	
firstname	varchar(255)	No	
lastname	varchar(255)	No	
email	varchar(255)	No	
dob	date	No	
address	varchar(255)	No	
city	varchar(255)	No	
mobile	bigint(255)	No	
suspended	tinyint(2)	Yes	*NULL*

5.1.9 in_experience

Column	Type	Null	Default
sb_cand_id	int(10)	No	
experience	varchar(255)	No	
sb_duration	int(3)	No	

5.1.10 in_industries

Column	Type	Null	Default
ind_id	bigint(20)	No	
sb_name	varchar(255)	Yes	*NULL*

5.1.11 in_jobs

Column	Type	Null	Default
job_id	bigint(20)	No	
sb_entrp_id	bigint(20)	No	0
sb_title	varchar(255)	Yes	*NULL*
sb_role	varchar(255)	Yes	*NULL*
sb_company_id	bigint(20)	No	0
sb_vacancies	bigint(20)	No	0
sb_posted_on	timestamp	No	CURRENT_TIMESTAMP
sb_experience	varchar(255)	Yes	*NULL*
sb_education	varchar(255)	Yes	*NULL*
sb_min_salary	decimal(15,2)	No	0.00
sb_max_salary	decimal(15,2)	No	0.00
sb_description	longtext	Yes	*NULL*
sb_approved	varchar(255)	Yes	*NULL*
job_type_id	bigint(20)	No	0
sb_job_status	bigint(20)	No	0
sb_show_profile1	varchar(255)	Yes	*NULL*
sb_front_featured	varchar(255)	No	no
sb_featured	varchar(255)	No	no
sb_highlight	varchar(255)	No	no
cat_id	int(11)	No	0
ind_id	int(11)	No	0
loc_id	int(11)	No	0

5.1.12 in_job_type

Column	Type	Null	Default
job_type_id	int(3)	No	
sb_job_name	varchar(255)	No	

5.1.13 in_locations

Column	Type	Null	Default
loc_id	bigint(20)	No	
sb_loc_name	varchar(255)	Yes	*NULL*

5.1.14 in_resumes

Column	Type	Null	Default
resume_id	bigint(20)	No	
sb_cand_id	bigint(20)	No	0
sb_name	varchar(255)	No	
sb_contact	varchar(255)	No	
sb_objective	longtext	Yes	*NULL*
sb_education	varchar(255)	No	
sb_job_type	int(11)	Yes	*NULL*
sb_internship	varchar(255)	No	
sb_curricular	int(255)	No	
sb_achievement	int(255)	No	
sb_firstname	varchar(255)	Yes	*NULL*
sb_lastname	varchar(255)	Yes	*NULL*
sb_city	varchar(255)	Yes	*NULL*
sb_state	varchar(255)	Yes	*NULL*
sb_zip	varchar(255)	Yes	*NULL*
sb_country	bigint(20)	No	-1
sb_mobile	varchar(255)	Yes	*NULL*
sb_email	varchar(255)	Yes	
sb_posted_on	timestamp	No	CURRENT_TIMESTAMP
sb_approved	varchar(255)	Yes	*NULL*
sb_img_url	varchar(255)	Yes	*NULL*

5.2 E-R Diagram

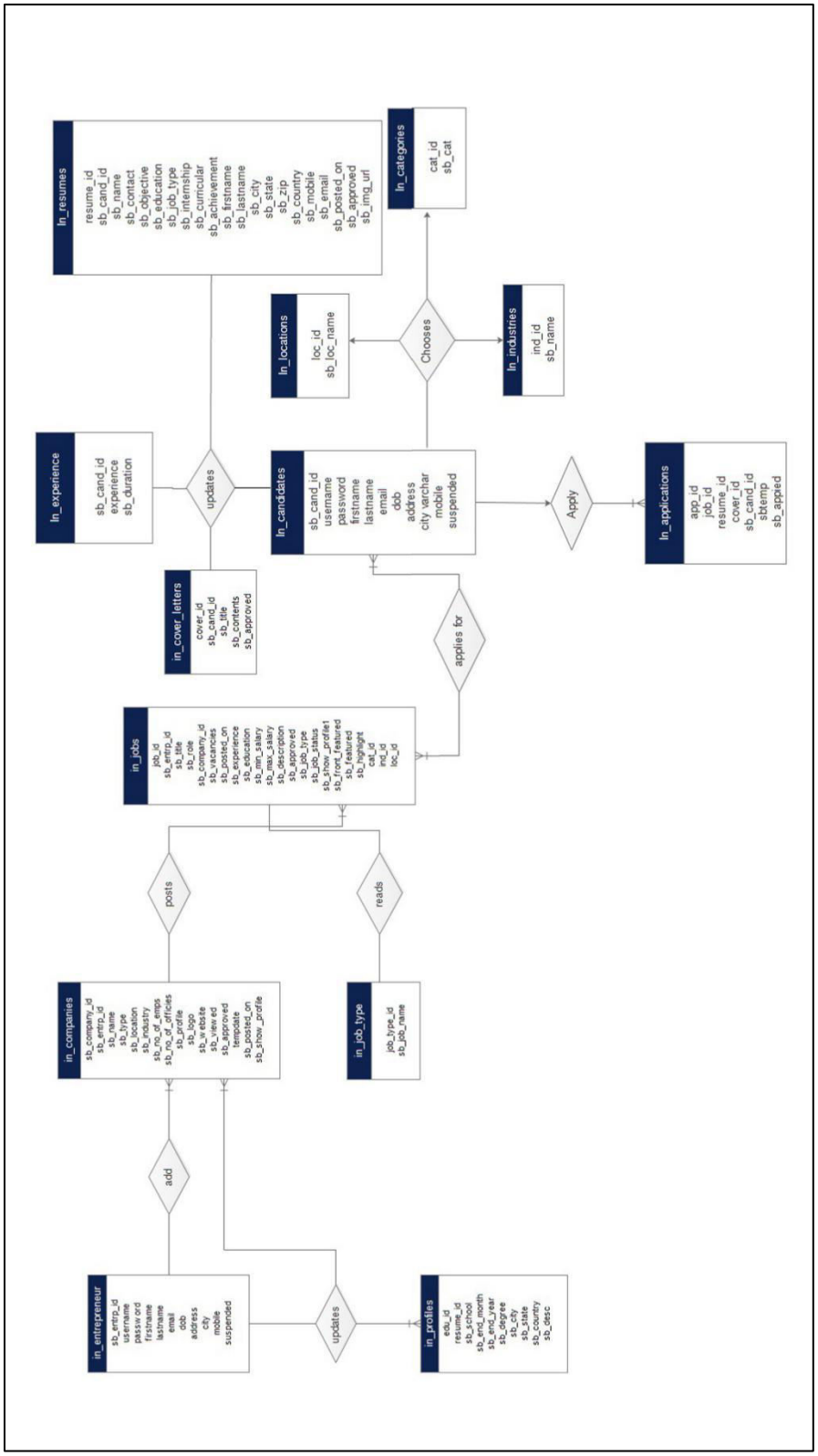

Figure 5.2

Chapter 6

IMPLEMENTATIONS

6.1 Screenshots

6.1.1 Administrator Login

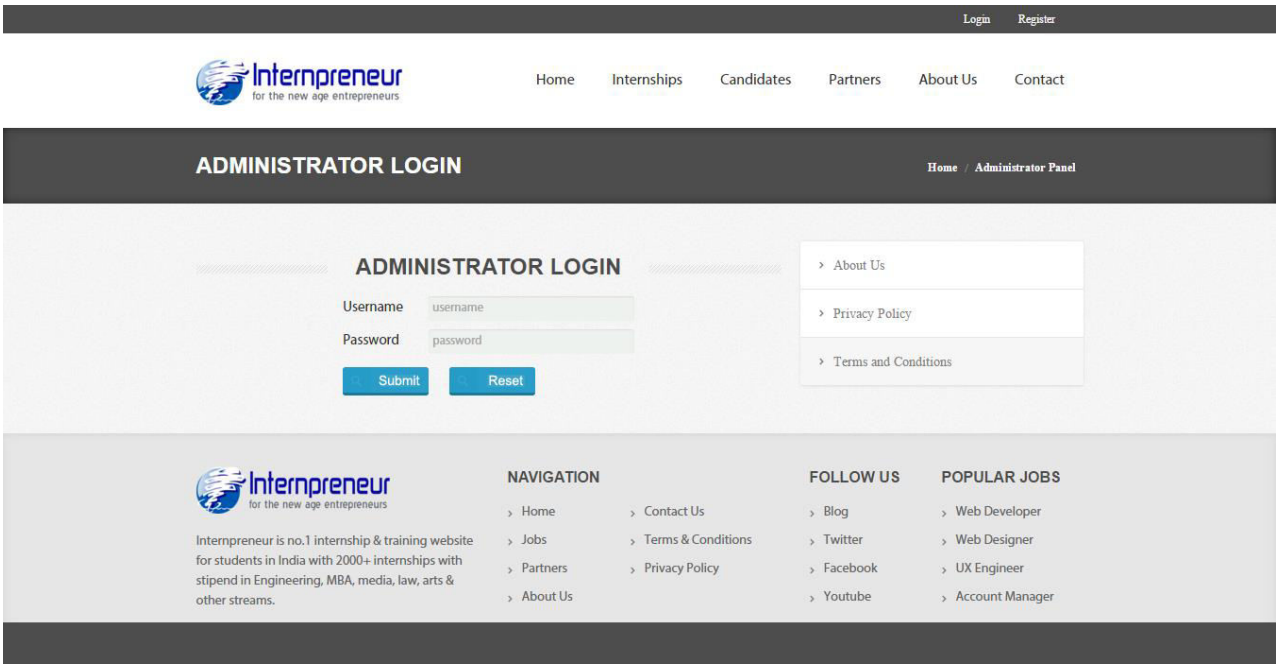

Figure 6.1.1 : Administrator Login (localhost/Internpreneur/admin/index.php)

6.1.2 Homepage

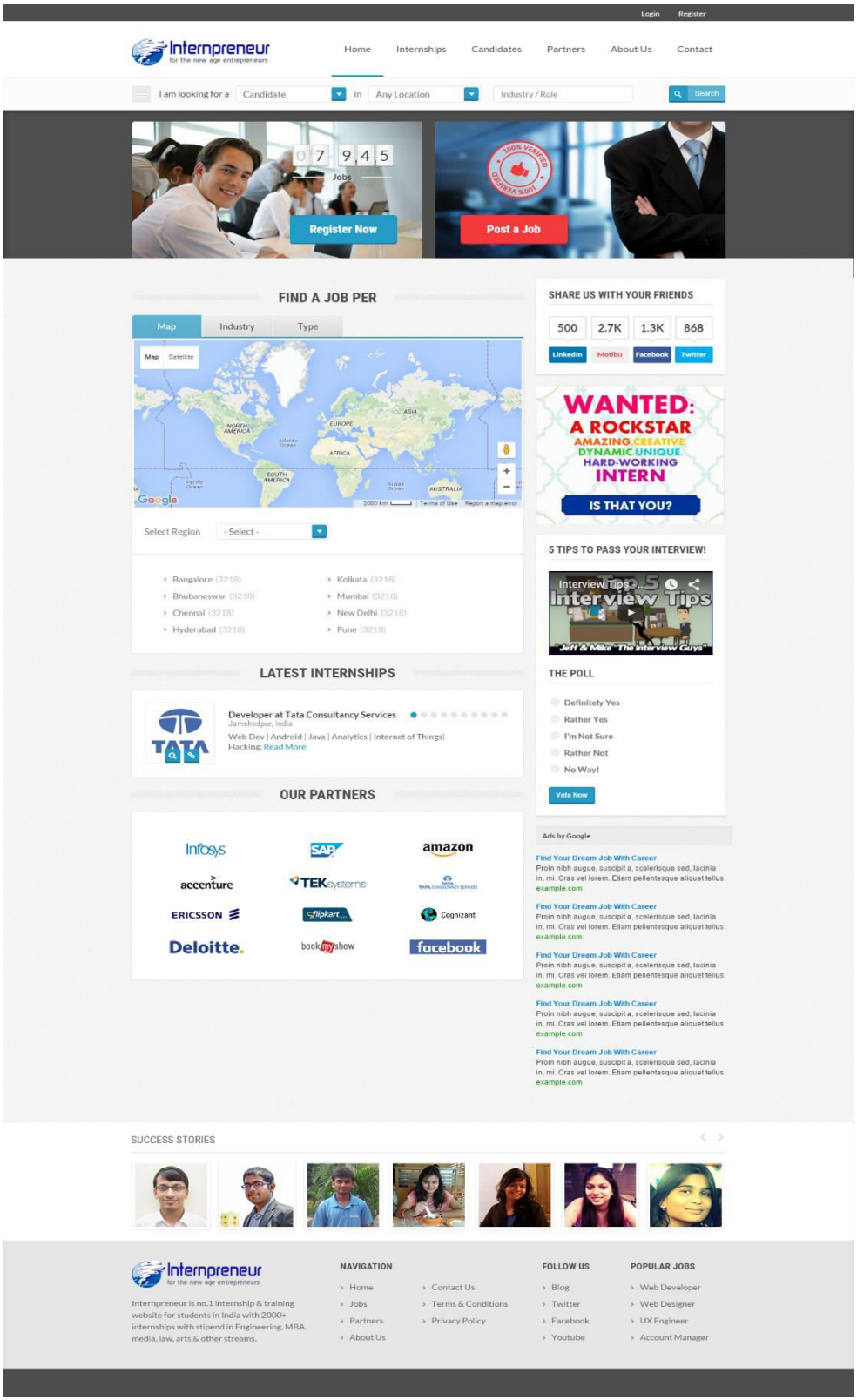

Figure 6.1.2 : Homepage (localhost/Internpreneur/index.php)

6.1.3 Registration Page

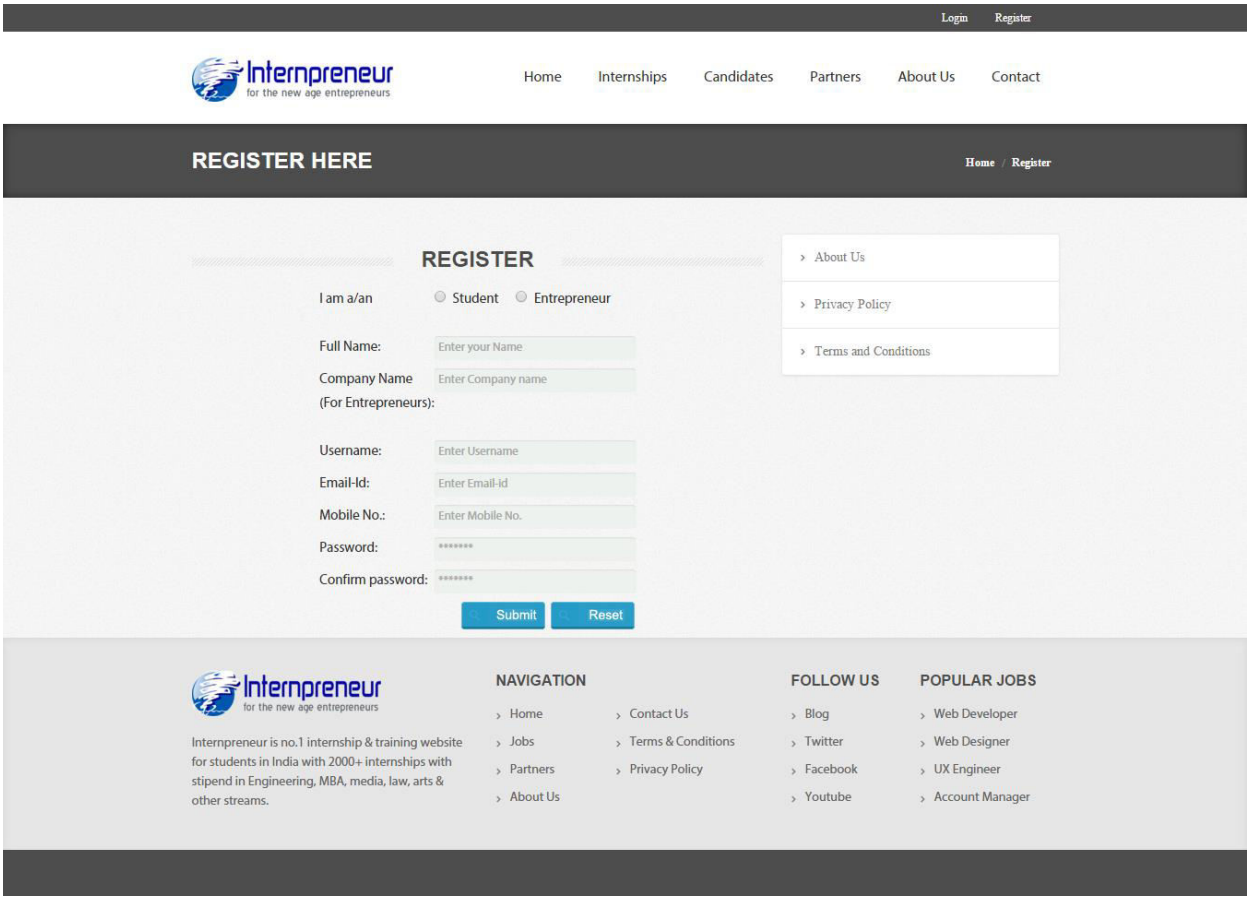

Figure 6.1.3 : Registration Page (localhost/Internpreneur/register.php)

6.1.4 Login Page

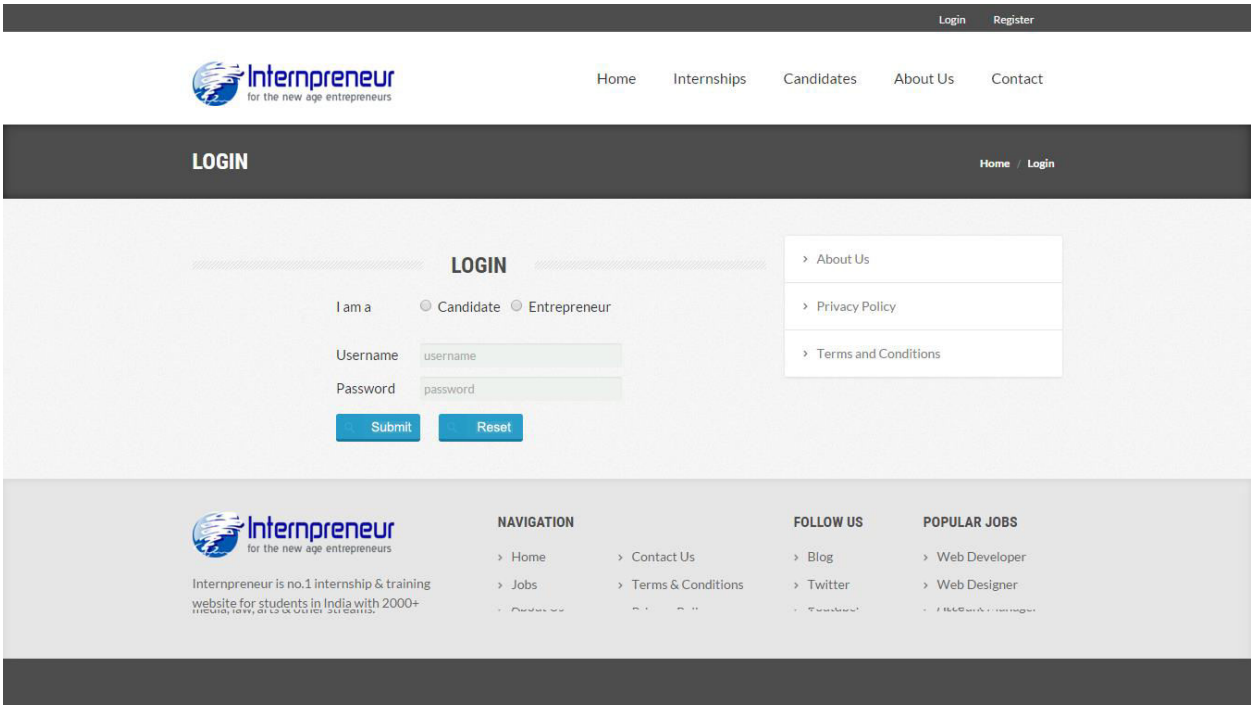

Figure 6.1.4 : Login Page (localhost/Internpreneur/login.php)

6.1.5 Student's Dashboard

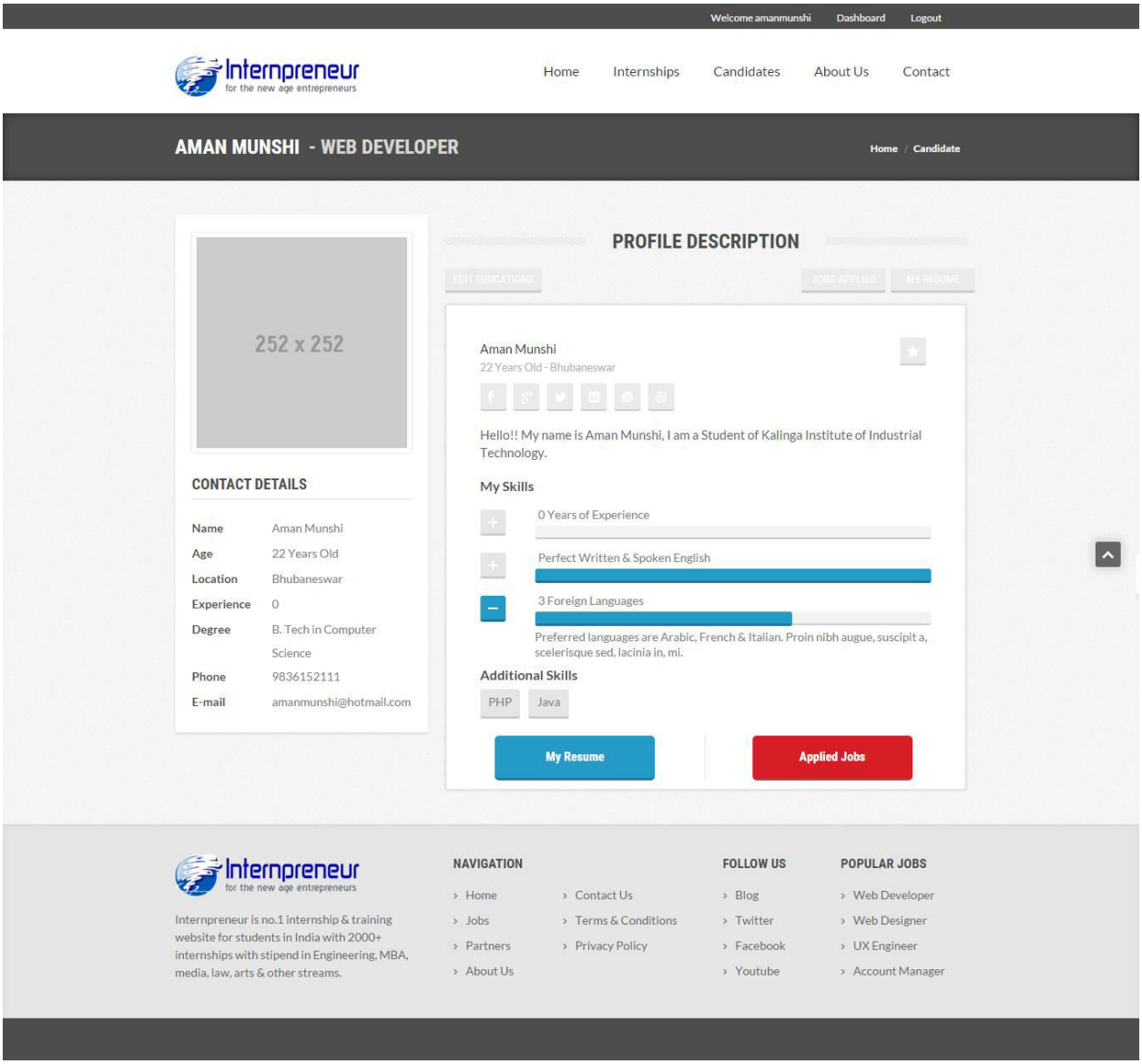

Figure 6.1.5 : Student's Dashboard Page (localhost/Internpreneur/cand_dashboard.php)

6.1.6 Entrepreneur's Dashboard

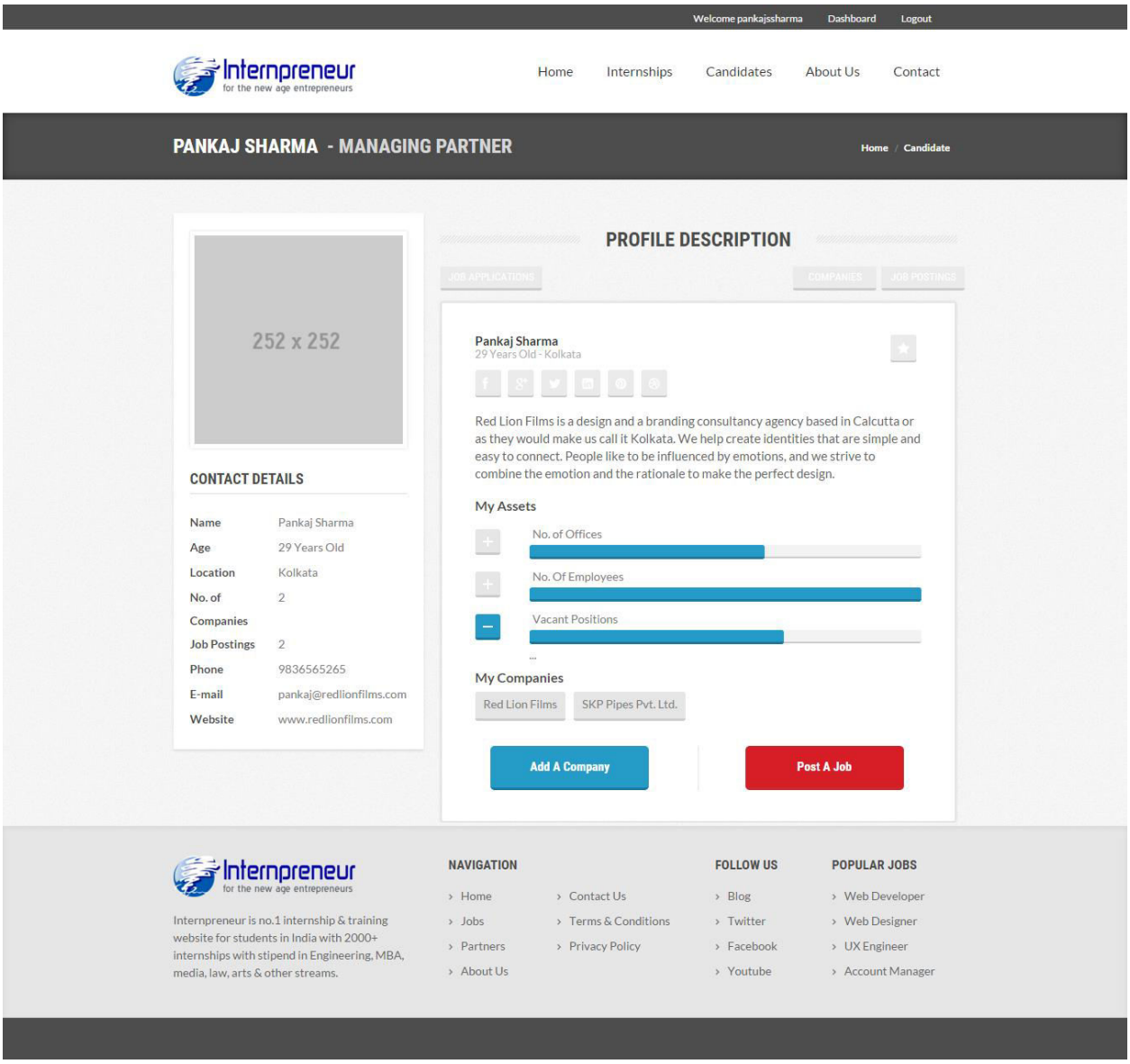

Figure 6.1.6 : Entrepreneur's Dashboard Page (localhost/Internpreneur/entrp_dashboard.php)

6.1.7 Add A Company

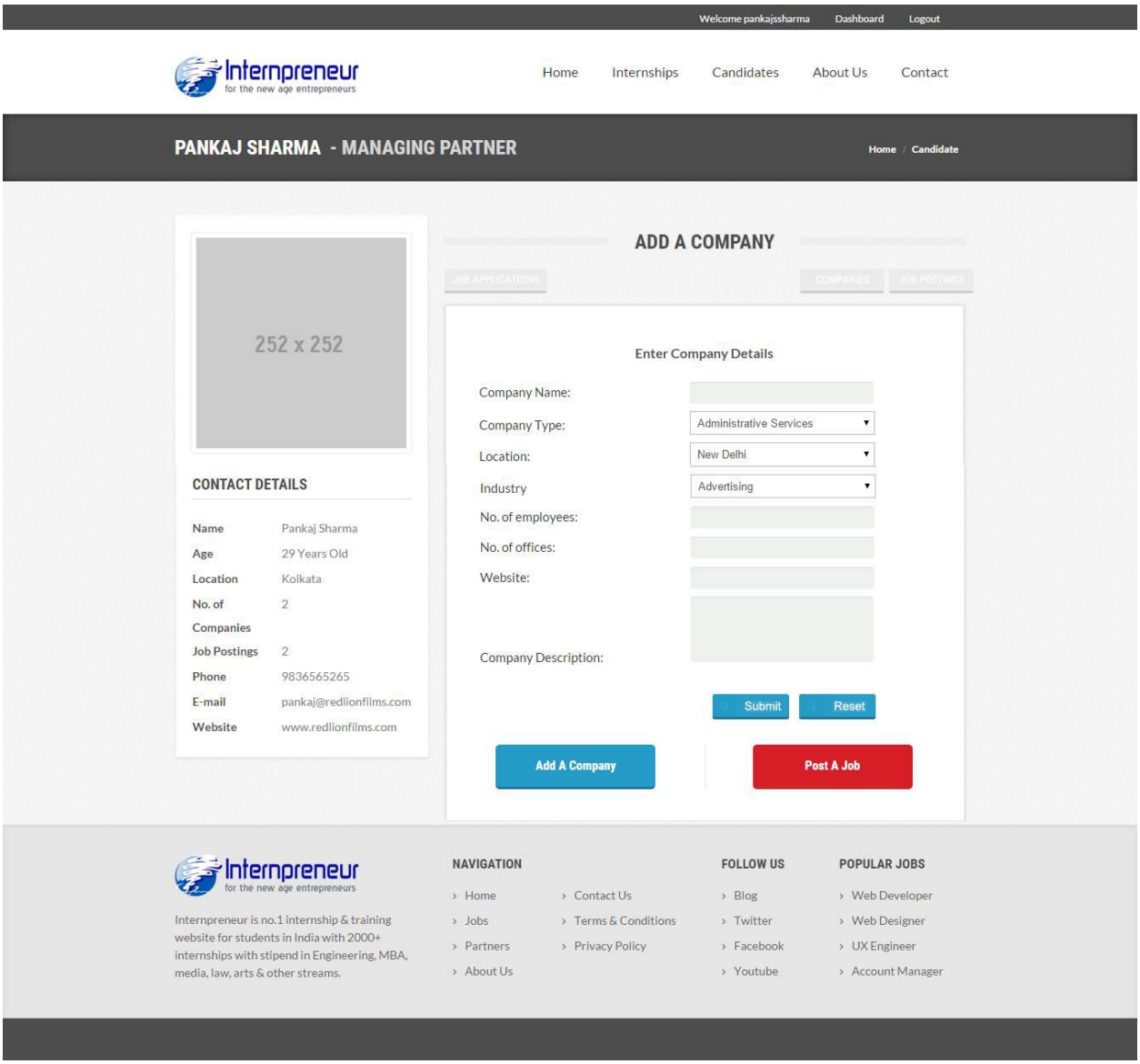

Figure 6.1.7 : Add A Company Page (localhost/Internpreneur/addacompany.php)

6.1.8 Post A Job

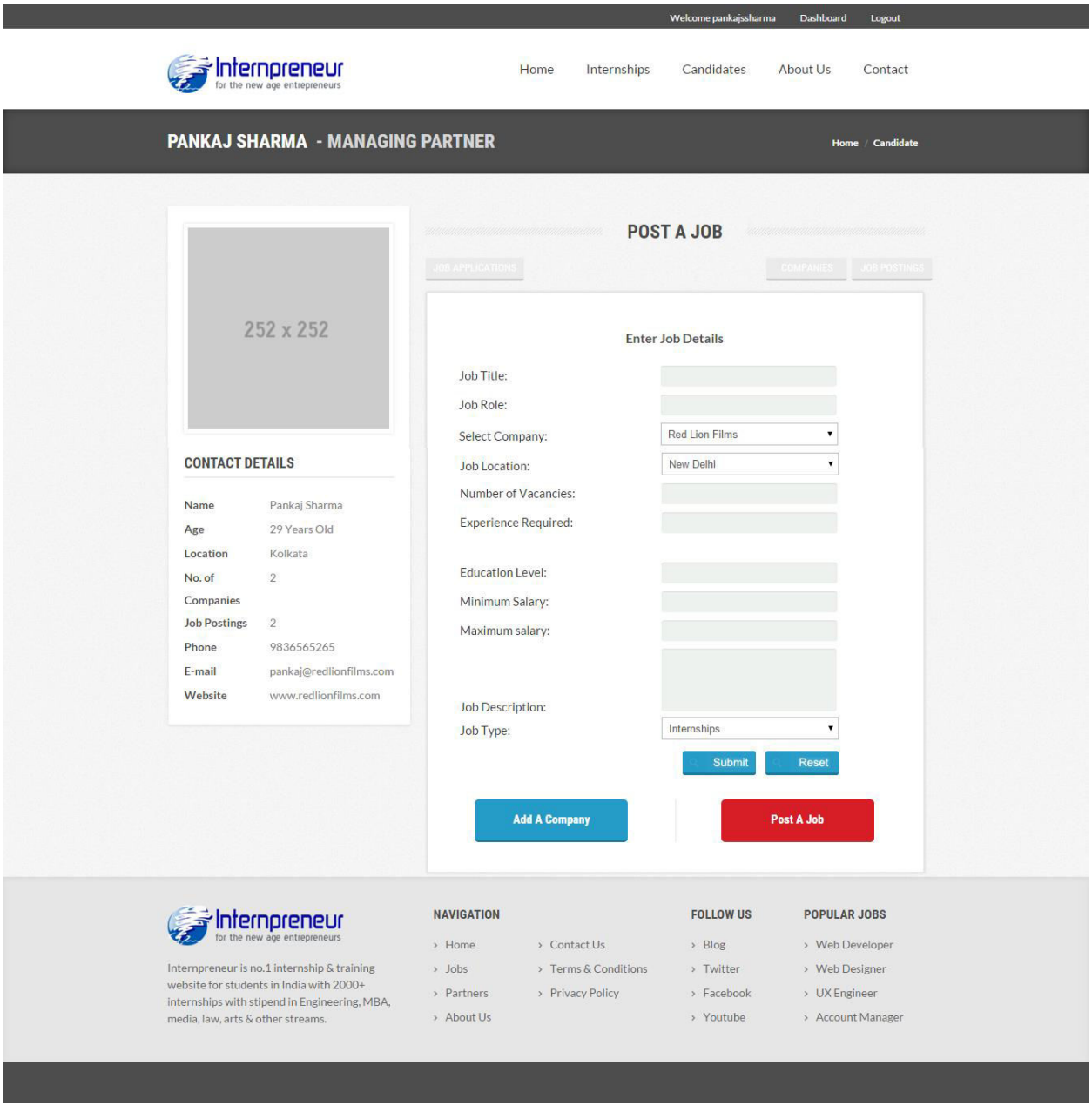

Figure 6.1.8 : Post A Job Page (localhost/Internpreneur/postajob.php)

6.1.9 Internships Page

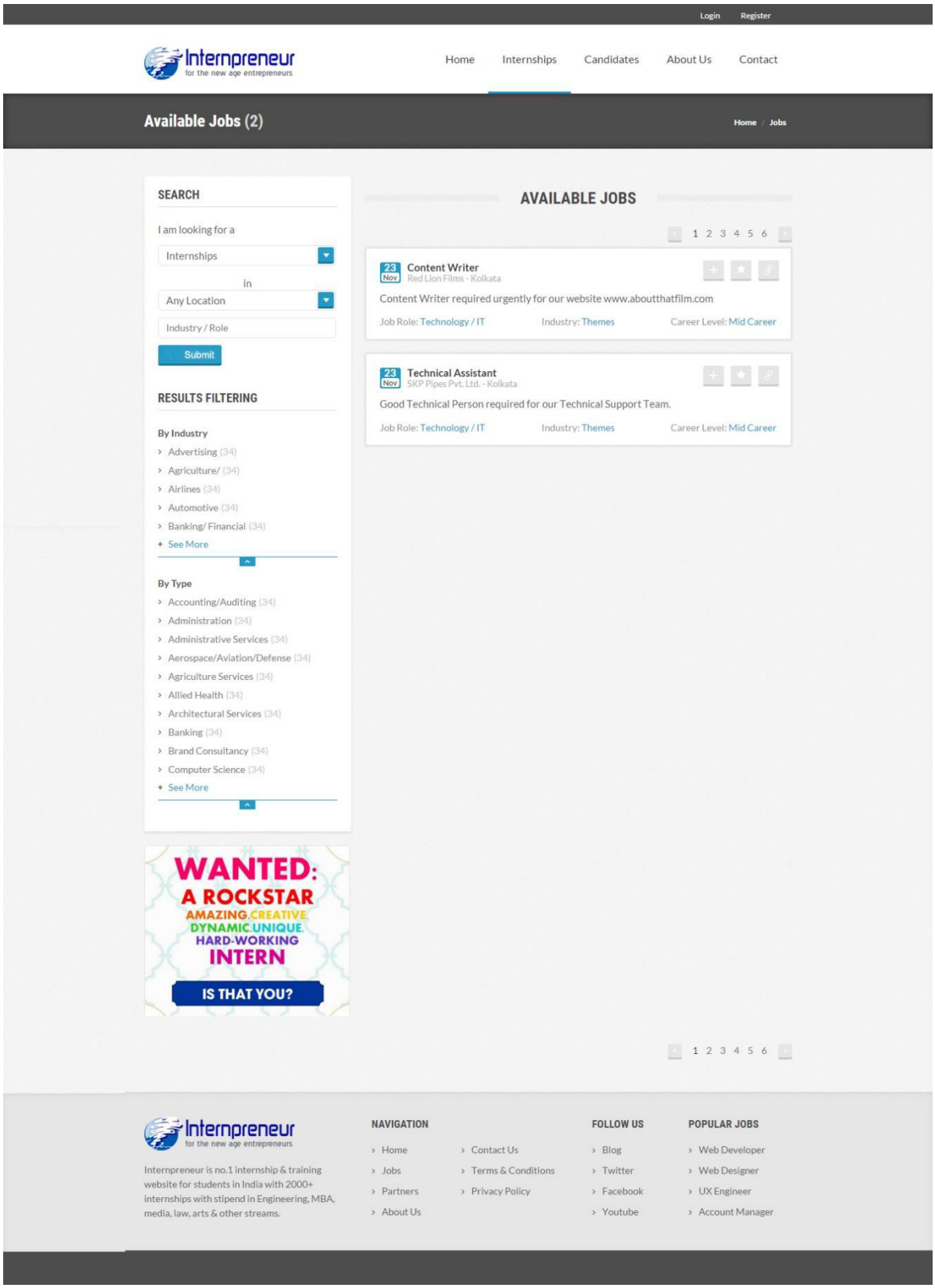

Figure 6.1.9 : Internships (localhost/Internpreneur/jobs.php)

6.1.10 Candidates Listing Page

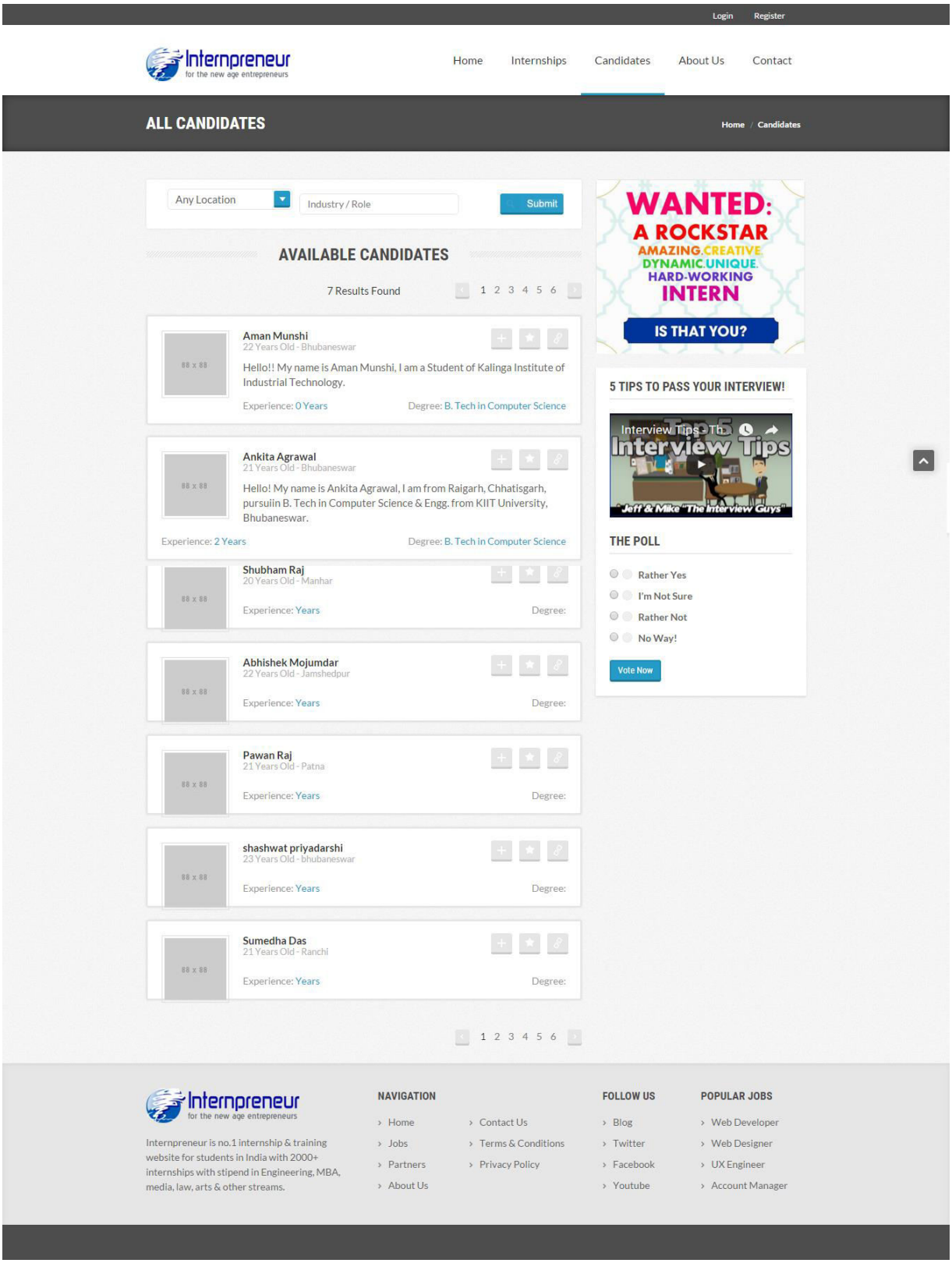

Figure 6.1.10 : Candidates (localhost/Internpreneur/candidates-listing.php)

6.1.11 ADMIN Dashboard Page

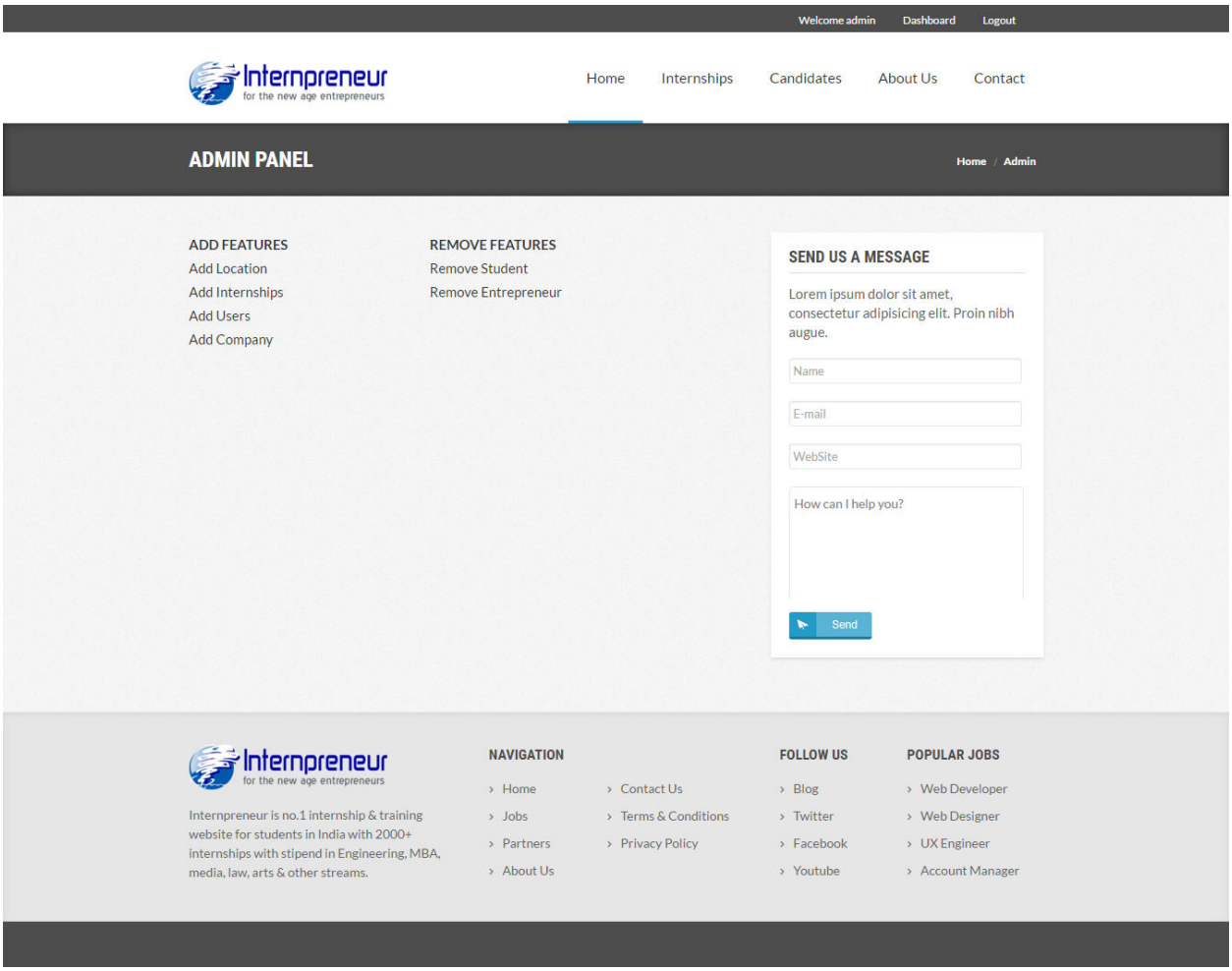

Figure 6.1.11 : Admin (localhost/Internpreneur/admin/admin.php)

6.1.12 Add Location Page

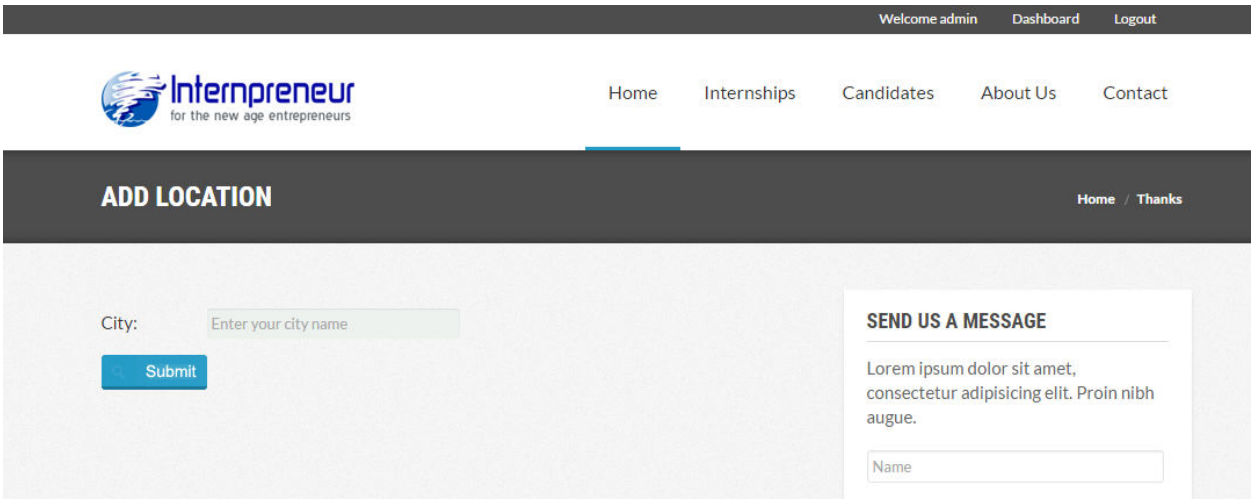

Figure 6.1.12 : Add Location (localhost/Internpreneur/admin/addlocation.php)

6.1.13 Add Users Page

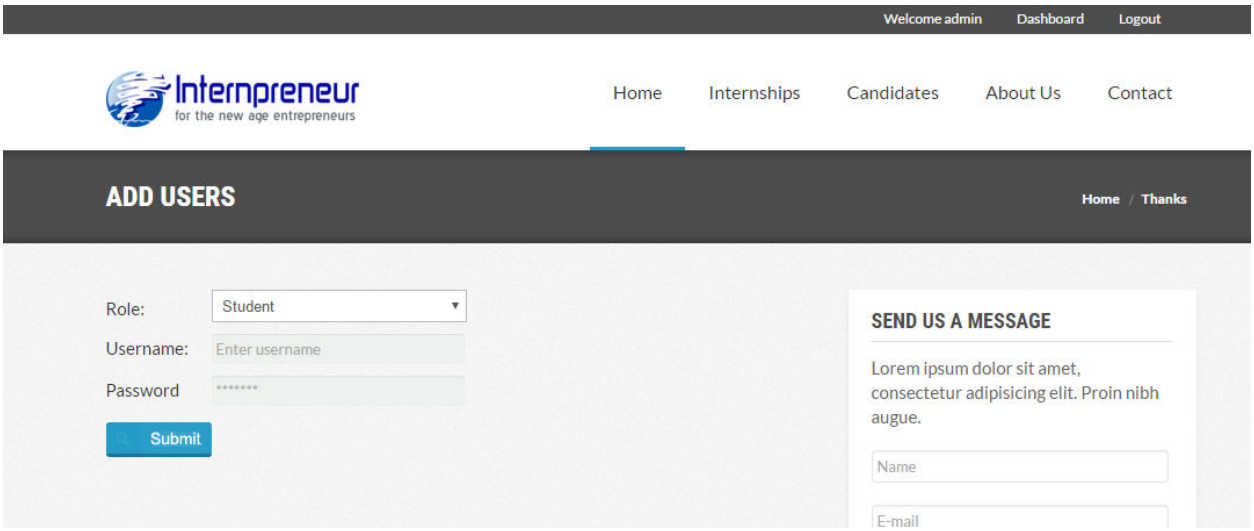

Figure 6.1.13 : Add Users (localhost/Internpreneur/admin/addusers.php)

6.1.14 Add Company Page

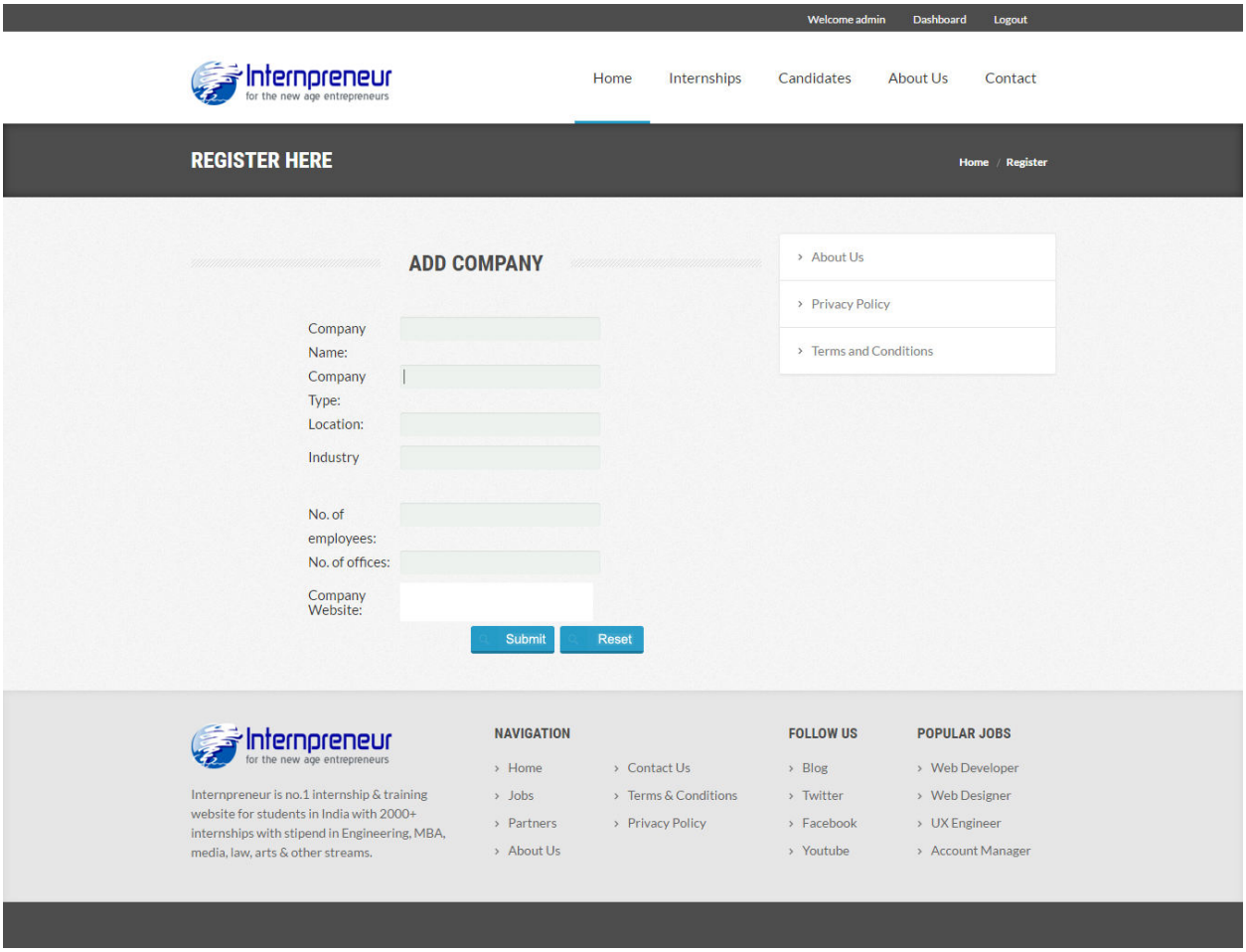

Figure 6.1.14 : Add Company (localhost/Internpreneur/admin/addcompany.php)

6.1.15 Add Internships Page

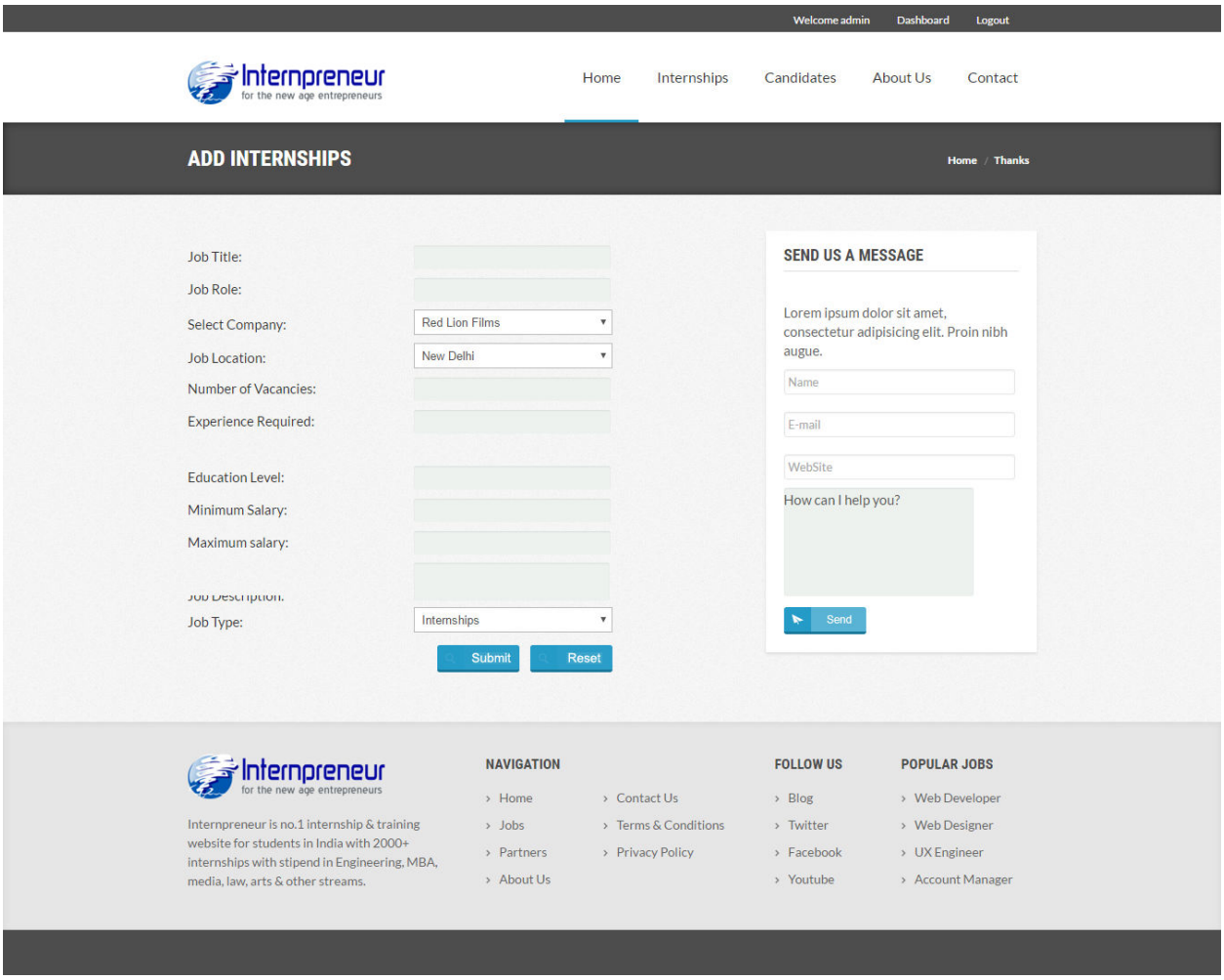

Figure 6.1.15 : Add Internships (localhost/Internpreneur/admin/addinternships.php)

6.1.16 About Us Page

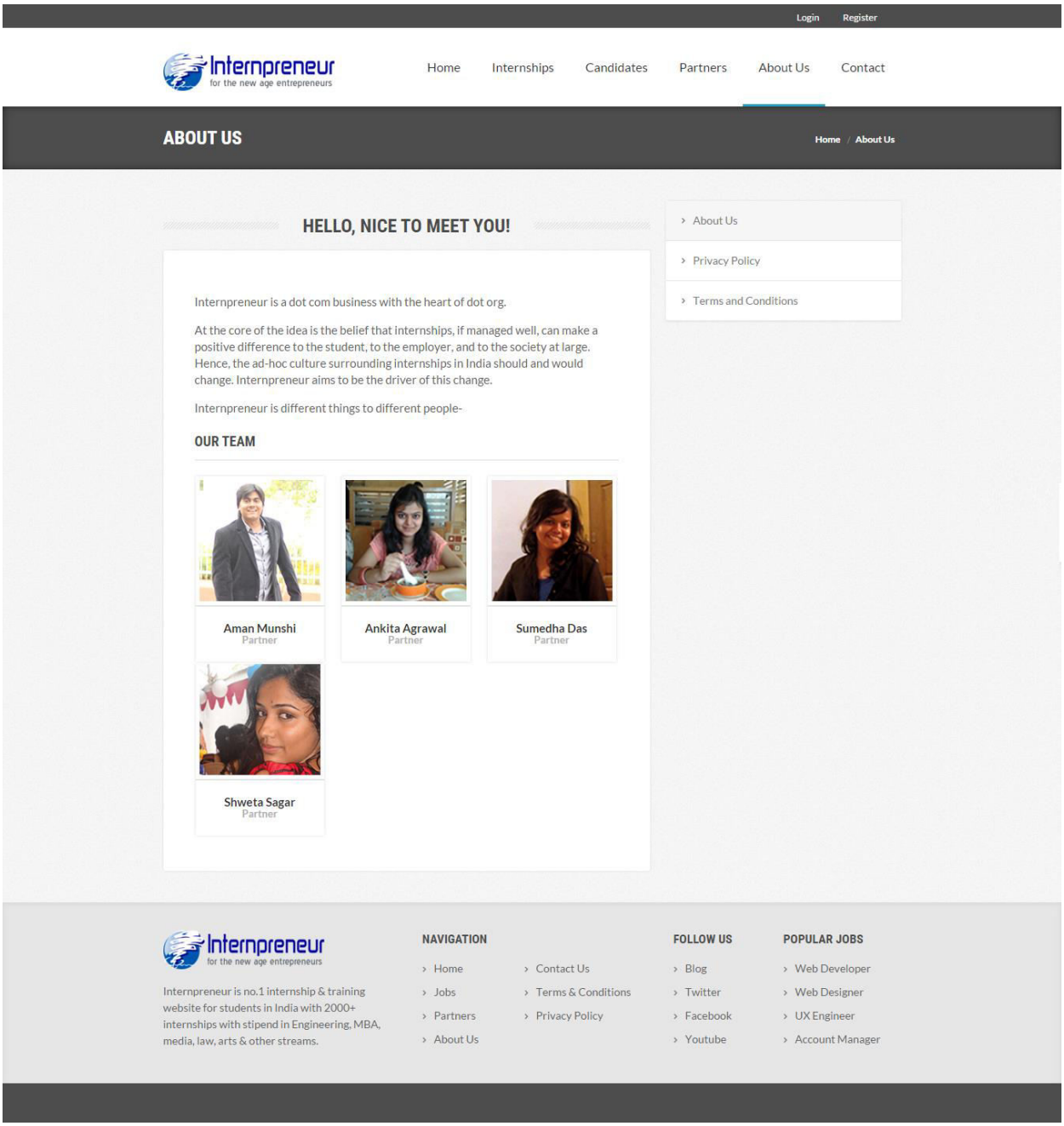

Figure 6.1.16: About Us (localhost/Internpreneur/about-us.php)

6.1.17 Contact Us Page

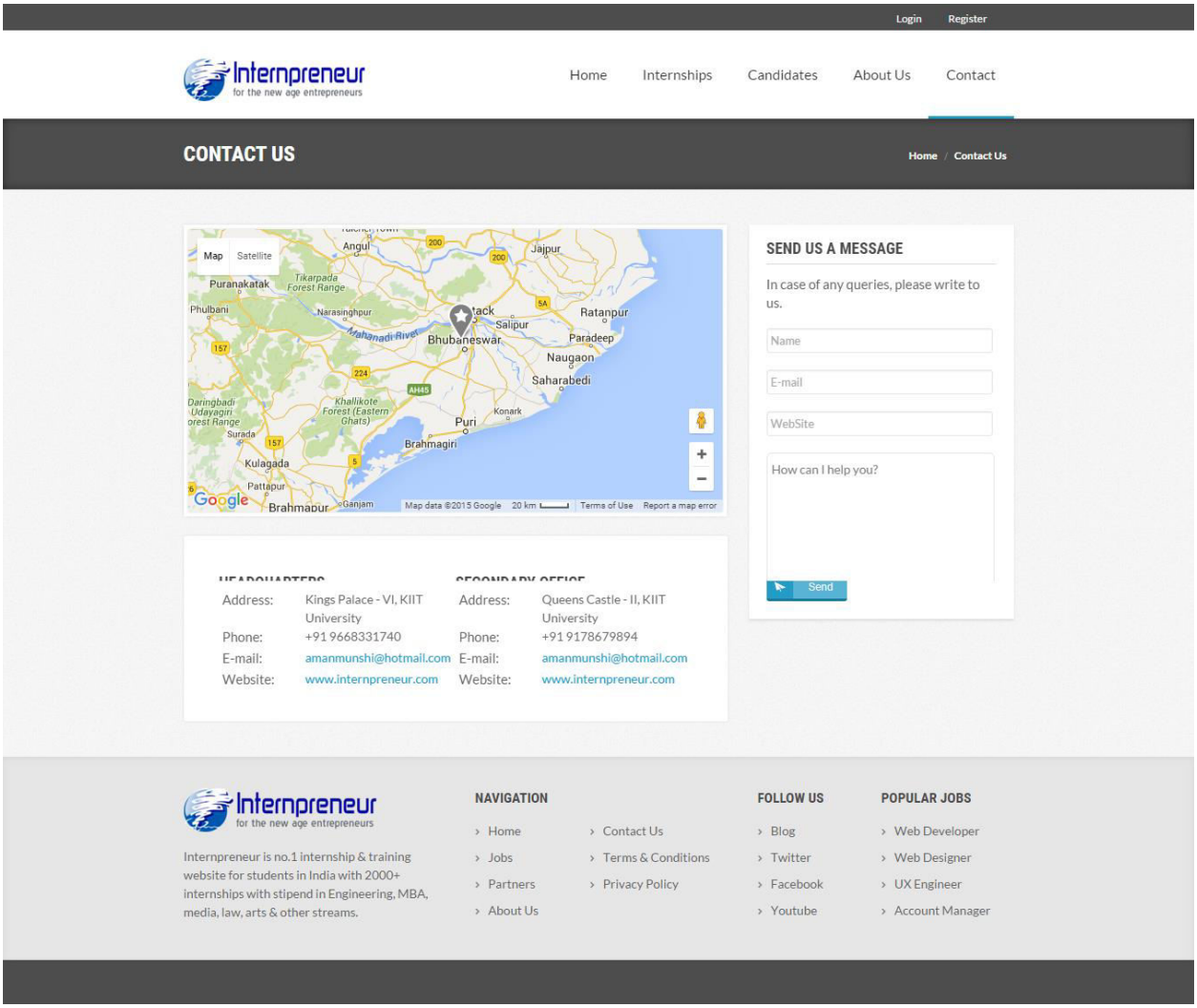

Figure 6.1.17 : Contact Us (localhost/Internpreneur/contacts.php)

CONCLUSION AND FUTURE SCOPE

For future development, we will first consider all previously assumed options which are not yet developed. This can make the website livelier in action, so that can be considered as popular online job portal website.

At this time, some future development has to be taken into account, such as:

- A fully-functional Admin Panel, which has not been implemented in this version.
- A payment system against a professional resume.
- Auto-Generated Mailing System informing both the Students and the Entrepreneurs.
- The errors occurred due to failure nature of the website, has to be reduce as far possible to make the site more user friendly in nature.
- This website would be launched in World Wide Web (www) to have the live version over the internet.

We will look for the whole prospective in the near future for a successful, most popular and user-friendly website which will be frequently updated with some extra new features as well as attractive altering ornamentations in the user interface.

REFERENCES

To develop this website certain helps were taken from different books of renowned authors. The references are, therefore, as follows:

1. Beginning PHP5, Apache, and MySQL Web Development
   - By **Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass**
2. PHP 6 and MySQL 5 for Dynamic Web Sites 2007
   - By **Larry Ullman**
3. Fundamentals of Software Engineering.
   - By **Rajib Mall**
4. www.php.net
5. www.w3schools.net

References

[1] pjreddie.com , YOLO and darknet resources.

[2] Stack over flow, python related queries.

[3] https://www.pyimagesearch.com, image processing related stuff.

[4] Way2sms.com - the messaging service API

[5] https://www.python.org/doc/essays/blurb/

[6] https://docs.opencv.org/master/d0/de3/tutorial_py_intro.html

[7] https://www.learnopencv.com/deep-learning-based-object-detection-using-yolov3-with-opencv-python-c/

[8] https://towardsdatascience.com/understanding-neural-networks-from-neuron-to-rnn-cnn-and-deep-learning-cd88e90e0a90

[9] https://www.guru99.com/introduction-to-mysql-workbench.html

[10] http://blog.way2sms.com/view.php?id=342&cid=6

[11] https://opensource.com/resources/what-docker

Appendix A

INDIVIDUAL CONTRIBUTION

NAME : RAHUL PANDEY

CONTRIBUTION IN THE PROJECT :

1. Contributed in the image processing part of the project.
2. Contributed in Yolo implementation in the system.
3. Traffic system code for counting the number of vehicles and regulating the traffic

NAME : ROHIT ROUSHAN

CONTRIBUTION IN THE PROJECT :

1. Implemented both the email and sms system that's sending message to the owner regarding fine for breaking traffic light.
2. Integrated the project with the database to search the owner information to send emails and messages.

Appendix B
Glossary

- **LMS :** Learning Management System
- **CMS :** Content Management System
- **LCMS :** Learning Content Management System
- **VLE :** Virtual Learning Environment
- **CBA :** Computer Based Assessment
- **CAA :** Computer Assisted Assessment
- **DSP :** Disital Signal Processing
- **MCQ :** Multiple Choice Question
- **TLE :** Teaching and Learning Environment
- **LMS :** Learning Management System
- **TMS :** Test Management System
- **TDS :** Test Delivery System
- **LDAP :** Lightweight Directory Access Protocol
- **LAMP :** Linux Apache MySQL PHP
- **WAMP :** Windows Apache MySQL PHP

In appendix include Source Code (Important code only, Optional) , Screen Shots etc.