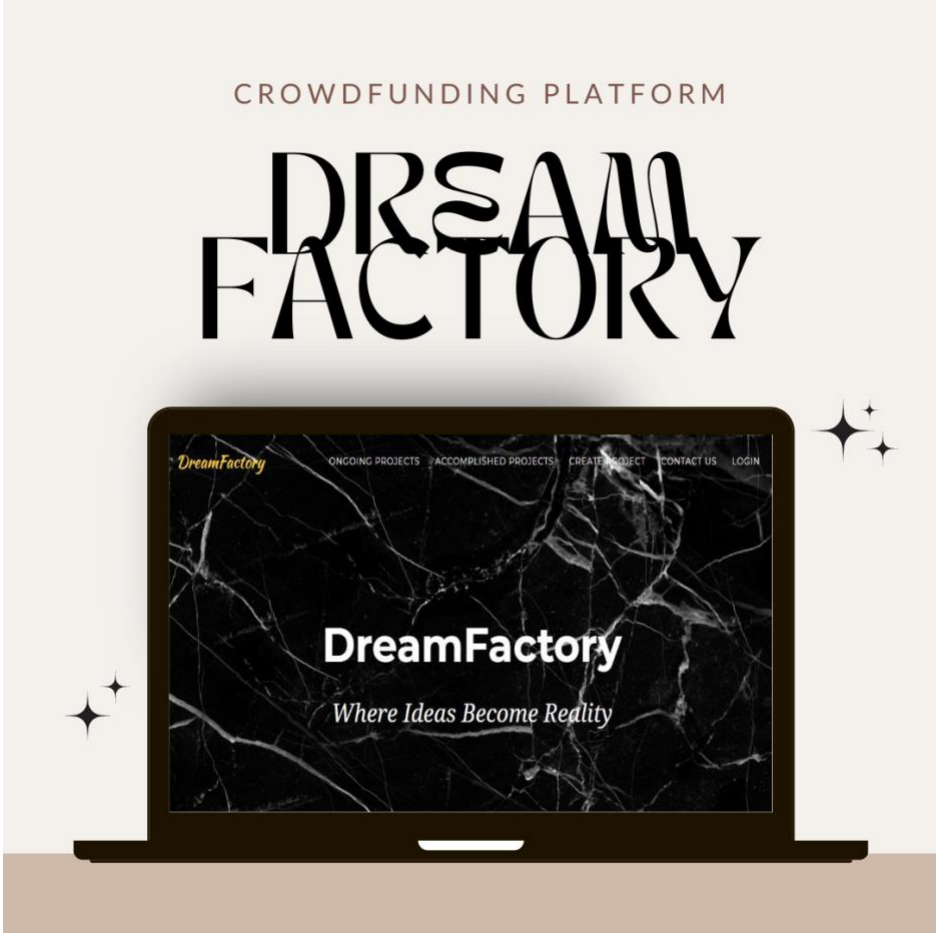


Crowdfunding platform

Report

Team Members:
Alvian Prasetya
Nurul Afiqah Binte Rashid
Adeeb Ashraf
Madeline Tooh Weiping

Table of Contents

Application Topic

1.1 Topic B: Crowdfunding

This application provides aims to provide entrepreneurs a platform to procure capital via crowdfunding. (e.g.,https://www.globalgiving.org, https://fundanything.com).

- Entrepreneurs can advertise their projects (title, description, start date, duration, keywords or categories, amount of funding sought, etc.).
- Users can browse the projects and choose projects to fund.
- Users can play roles of both entrepreneurs and investors.
- The application tracks the current amount of funds raised, updates the status of funded projects, and advertises successfully funded projects on a webpage and shows unfunded projects
- Each user has an account that can create and fund projects.
- Administrators can create/modify/delete entries.

1.2 Solution: DreamFactory

The group came up with DreamFactory , a crowdfunding interface platform, for entrepreneurs to advertise the projects and also, for investors and other entrepreneurs to browse and fund these projects. This platform will be on a webpage that resides on a database to store all the user account details, entries, project details such as project name, funding amount, etc. The logic of the transactions, funding and browsing of project will be coded in Hypertext Preprocessor (PHP) and Structured Query Language (SQL) queries, while the users' interfaces will be the webpage.
Refer to Table 1 in Annex B for the breakdown of tiers in DreamFactory.

Specifications of Software Tools/ Frameworks

2.1 Software Tools

Here are the software tools used for the project:

- PhpMyAdmin for MySQL framework,
- Apache Web Server to run and test web pages,
- Responsive Web framework
    - using Bootstrap framework
    - consists of Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), Hypertext Preprocessor (PHP) scripts and Javascript files.

2.2 Frameworks

The framework comprises of the MySQL framework and responsive web framework.

2.2.1 MySQL Framework

The MySQL framework can be visualised and accessed using PhpMyAdmin. The database is implemented using an imported MySQL Data Definition Language (DDL) script. Refer to

Annex A for the DDL script.

The group has decided to use PhpMyAdmin because it has a user-friendly interface that supports MySQL functions like browse, drop, create, copy and alter databases, tables, views, fields and indexes, execute MySQL queries, manage stored procedures and functions.
Refer to Figure 1, in Annex B, for the snapshot of the tables residing in the database for DreamFactory.

2.2.2 Responsive Web Framework

The responsive web framework involves an integration in Bootstrap, CSS, HTML, JavaScript and PHP. Bootstrap provides all the necessary and essential templates to have well-rounded web page design.

Bootstrap has templates of HTML, CSS and JavaScript which we used to generate an aesthetically pleasing website. We used CSS to add colour and enhancements to the design and appearance of the webpages. JavaScript adds behavior to the web pages in order for users to load a new web page by responding to user's actions and requests. We used PHP to interact with our database and process queries and sensitive information (login, etc) on the server side to ensure security. Refer to web page screenshots in Annex D.

Entity Relationship (ER)

3.1 Entities

The entities that are present in DreamFactory are 'users', 'projects', 'categories' and 'keywords'.

- Users (**user_id**, email, password, full_name, privilege, session_id, session_exp)
- Projects (**project_id**, **user_id**, **title**, image_url, description, end_datetime, **category**, funding_goal)
    - user_Id: references foreign category in Users
    - category: references foreign category in Categories
- Categories (**category**)
- Keywords (**keyword**)

Refer to Annex C for the ER diagram.

3.2 Relationships

There are two relationships between 'users' and 'projects' entities. One of the relationship is 'own' which is a one-to-many relationship, 'projects'(1,1) and 'users'(0,n). There is a constraint that each project created has to be owned by one user but one user can own zero to many projects. The other relationship is 'fundings', which has attributes, 'amount' and 'timestamp', and has a many-to-many relationship, 'projects'(0,n) and 'users'(0,n). There is a constraint that

each project can be funded by zero to many users and each user can fund zero to many projects.

There is also a relationship between ‘projects’ and ‘keywords’ entities called ‘contain’ and it is a many-to-many relationship, ‘projects’(1,n) and ‘keywords’(1,n). There is a constraint that projects have to contain at least one keyword and keywords have to be related to at least one project.

Lastly, there is a relationship between ‘categories’ and ‘projects’ entities called ‘has’ and it is a one-to-many relationship, ‘projects’ (1,1) and ‘categories’(0,n). There is a constraint that each project has one category and one category can be related to zero to many projects. Refer to Annex C for the ER diagram.

Relational Schema

4.1 Relational Schema of DreamFactory Database

The database is named as ‘dreamfactory’. Once the entities and relationships are in place, the database is set up to conduct operations on entities such as ‘projects’, ‘keywords’, ‘categories’ and ‘users’. Refer to Annex A for the DDL script.

There are six tables in total which are ‘fundings’, ‘projects’, ‘keywords’, ‘project_keywords’, ‘categories’ and ‘users’. Refer to Section 3.1 and 3.2 for the attributes of the entities. The following constraints are in place:

- Table ‘fundings’ has attributes that are all ‘NOT NULL’, amount is ‘UNSIGNED NOT NULL’ because it ensures that funding amounts inputted by users are not negative values and timestamp is ‘NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP’. The primary key is ‘(user_id, project_id)‘ whereby ‘user_id’ is referenced from Table ‘users’ and ‘project_id’ is referenced from Table ‘projects’.
- Table ‘projects’ has attributes that are all ‘NOT NULL’ except for ‘image_url’ because it is optional. ‘project_id’ is set as ‘AUTO_INCREMENT’. ‘title’ is set as unique key.
- Table ‘project_keywords’ has attributes that are set as ‘NOT NULL’. The primary key is ‘(project_id, keyword)’ whereby ‘project_id’ is referenced from Table ‘projects’ and ‘keyword’ is referenced from Table ‘keywords’.
- Table ‘users’ has attributes that are all ‘NOT NULL’ except for ‘session_id’ and ‘session_exp’. ‘user_id’ is set as ‘AUTO_INCREMENT’. ‘privilege’ is set as ‘DEFAULT ‘0’’ because only those users labelled as ‘1’ are administrators of DreamFactory. ‘email’ is set as unique key.
- Table ‘categories’ has attributes that are set as ‘NOT NULL’.

Application Functionalities

5.1 Creation of Data

5.1.1 Creation of User

All users have fixed accounts identified by their emails. They are also tagged server side with indexes. A special index number is used for Administrator accounts. Users can create their accounts via the registration page. Refer to Figure 6 in Annex D for screenshot of Registration page.

5.1.2 Creation of Projects

All users have the ability to create a project for funding. This can be done via the create project page. Each project is identified by a unique project ID that is assigned by the server. To create project, user has to be logged in first. Refer to Figure 8 in Annex D for the screenshot of Create Project page.

5.1.3 Creation of Category

Only Administrators are allowed to create a category. This category is universal and users can only choose a category that is already on the list when creating project.

5.1.4 Creation of Keyword

All users can add keywords to their project. The keyword is added to a master list and also tagged to their project.

5.1.5 Funding

All users can fund any project for any non-negative amount. Attempting to input a negative amount will result in an error and no amendment is made to the database. Refer to Figure 9 in Annex D for the Funding page.

5.2 Deletion of Data

5.2.1 Deletion of Project

Only administrators or the owners of the project can delete the project.

5.3 Update of Data

5.3.1 Update Funding

The total amount funded will be updated automatically based on how much funds users have inputted into the project. Each amount funded, it is associated with a unique user id and project id.

5.4 Browsing Data or Search Functions

5.4.1 Search Project

The search function can be found in the Browse Project page where users can search projects by typing the project title or clicking on the highlighted keywords or categories under the projects tables to get projects with the same keywords or categories. Refer to Figure 4 in Annex D for the search function under Browse Projects page.

5.4.2 Browsing Projects

Browsing of projects can be filtered into two tables, funded projects and ongoing projects. Refer to Section 6.1.1 and 6.1.2 for detailed explanations on how the tables of projects are derived.

5.5 Retrieval of Interesting Statistics on Data

5.5.1 Percentage of Project Funding Completion

This statistic can be found under projects in the Homepage and it is useful to reflect how close or how far the project is from being fully funded. Refer to Figure 3 in Annex D for examples displayed in the homepage. The percentage of project funding completion is calculated as follows:

$$\frac{total\ funding\ amount\ from\ users}{funding\ goal\ of\ project} \times 100\%$$

5.5.2 Total Number of Users

This statistic can be found in the Homepage and it reflects how many users DreamFactory currently has. Refer to Figure 11 in Annex D the statistic displayed in the homepage.

5.5.3 Total Number of Projects

This statistic can be found in the Homepage and it reflects how many projects have been created by users in DreamFactory. Refer to Figure 11 in Annex D the statistic displayed in the homepage.

5.5.4 Average Funding of Projects

This statistic can be found in the Homepage and it is useful to reflect how much the average funding of projects is. Refer to Figure 11 in Annex D the statistic displayed in the homepage. The average funding of projects is calculated as follows:

$$\$ \left(\frac{total\ funding\ amount\ from\ all\ projects}{total\ number\ of\ projects}\right) per\ project$$

5.6 Non-trivial Integrity Constraint

5.6.1 Funding Amount

The amount under ‘fundings’ table is set as ‘UNSIGNED NOT NULL’ in the DDL. It is set as so to ensure that the input for amount is always non-negative. If user types in a negative funding amount to the project an error message will be shown and no change will be made to the database. This is also done to ensure that no user can reduce total funded amount.

5.6.2 Foreign Keys under ‘projects’ Table

There are constraints set to the ‘projects’ table. Below is a snippet of the DDL code.

```
ALTER TABLE `projects`
  ADD CONSTRAINT `fk_projects_categories_category` FOREIGN KEY
(`category`) REFERENCES `categories` (`category`) ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_projects_users_user_id` FOREIGN KEY (`user_id`)
REFERENCES `users` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE;
```

SQL Documentation

6.1 SQL DML Complex Queries

6.1.1 Funded Projects

Funded projects are selected projects whereby (1) the funding goal has been reached and the deadline is over, or (2) the funding goal has been reached but the deadline is not over yet.
Refer to Figure 12 for the query of retrieving tables of funded projects in Annex E.

6.1.2 Ongoing Projects

Ongoing projects are selected projects whereby the funding goal is not reached and the deadline is now over yet.
Refer to Figure 13 for the query of retrieving tables of ongoing projects in Annex E.

Future Improvements

7.1 SQL Queries

7.1.1 Transactions

Modify SQL queries into transactions to maintain isolation. This will prevent incorrect updates to the database when concurrent users are making requests. For instance, adding two different projects but with the same titles which then violates the constraint of project titles being unique. The transaction would be able to control which project will be added first and the other will be thrown an error that project title already exists.

7.2 Back end

7.2.1 Security

Implement SSL to encrypt the interaction between users and the server.

7.3 Front end

7.3.1 Interface

Implement a better interface to make it easier to fund projects.

Annex A

Data Definition Language Script

Below is the DDL script inclusive of some data already created as examples.

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `dreamfactory`
--

-- --------------------------------------------------------

--
-- Table structure for table `categories`
--

DROP TABLE IF EXISTS `categories`;
CREATE TABLE IF NOT EXISTS `categories` (
  `category` varchar(32) NOT NULL,
  PRIMARY KEY (`category`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `categories`
--

INSERT INTO `categories` (`category`) VALUES
('Business'),
('Ecommerce'),
('Education'),
('Entertainment'),
('Financials'),
('Food'),
('Gaming'),
('Media'),
('Social Network'),
('Technology'),
('Telecommunication'),
('Travel'),
('Utilities');

-- --------------------------------------------------------

--
-- Table structure for table `fundings`
--

DROP TABLE IF EXISTS `fundings`;
CREATE TABLE IF NOT EXISTS `fundings` (
  `user_id` int(11) NOT NULL,
  `project_id` int(11) NOT NULL,
  `amount` int(11) UNSIGNED NOT NULL,
  `timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`user_id`,`project_id`),
  KEY `user_id` (`user_id`),
  KEY `project_id` (`project_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
```

```
-- Dumping data for table `fundings`
--

INSERT INTO `fundings` (`user_id`, `project_id`, `amount`, `timestamp`) VALUES
(1, 1, 2000, '2018-04-09 11:57:28'),
(1, 12, 34000, '2018-04-10 07:09:26'),
(5, 1, 12000, '2018-04-09 13:01:53'),
(6, 3, 25000, '2018-04-09 12:48:08'),
(7, 2, 9000, '2018-04-09 13:02:41'),
(7, 4, 4000, '2018-04-09 12:46:42'),
(8, 4, 34000, '2018-04-09 13:03:51'),
(9, 7, 7000, '2018-04-09 12:45:15'),
(9, 10, 5000, '2018-04-09 13:05:41'),
(10, 2, 2000, '2018-04-09 13:02:31'),
(11, 6, 7500, '2018-04-09 12:47:34'),
(14, 10, 230, '2018-04-09 13:04:15'),
(18, 10, 10000, '2018-04-09 13:04:55'),
(20, 2, 4500, '2018-04-09 13:04:28');

-- --------------------------------------------------------

--
-- Table structure for table `keywords`
--

DROP TABLE IF EXISTS `keywords`;
CREATE TABLE IF NOT EXISTS `keywords` (
  `keyword` varchar(32) NOT NULL,
  PRIMARY KEY (`keyword`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `keywords`
--

INSERT INTO `keywords` (`keyword`) VALUES
('app'),
('coding'),
('computer'),
('dating'),
('delivery'),
('expert'),
('final'),
('fire'),
('food'),
('funding'),
('game'),
('getjar'),
('knowledge'),
('loop'),
('money'),
('music'),
('project'),
('score'),
('sky'),
('song'),
('travel');

-- --------------------------------------------------------

--
-- Table structure for table `projects`
--

DROP TABLE IF EXISTS `projects`;
CREATE TABLE IF NOT EXISTS `projects` (
  `project_id` int(11) NOT NULL AUTO_INCREMENT,
  `user_id` int(11) NOT NULL,
  `title` varchar(256) NOT NULL,
  `image_url` varchar(256) DEFAULT NULL,
  `description` varchar(1024) NOT NULL,
  `end_datetime` datetime NOT NULL,
  `category` varchar(32) NOT NULL,
```

```
  `funding_goal` int(11) NOT NULL,
  PRIMARY KEY (`project_id`),
  UNIQUE KEY `title` (`title`),
  KEY `category` (`category`),
  KEY `user_id` (`user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `projects`
--

INSERT INTO `projects` (`project_id`, `user_id`, `title`, `image_url`, `description`,
`end_datetime`, `category`, `funding_goal`) VALUES
(1, 3, 'Moolah',
'https://www.shareicon.net/data/128x128/2016/06/30/788748_notes_512x512.png',
'Financing company', '2018-02-28 00:00:00', 'Financials', 12000),
(2, 5, 'Skyrocket',
'https://i.pinimg.com/564x/6d/a8/13/6da813bbec754b95ba5bb72b1a61eca9.jpg', 'Educating
business', '2018-03-10 00:00:00', 'Education', 10000),
(3, 4, 'Gimmefood',
'https://www.shareicon.net/download/2016/09/02/824429_fork_512x512.png', 'Food
Delivery Company', '2018-12-31 00:00:00', 'Food', 340000),
(4, 6, 'Helloworld',
'http://icons.iconarchive.com/icons/dtafalonso/modern-xp/256/ModernXP-73-Globe-icon.png'
, 'Coding class start up', '2018-02-28 00:00:00', 'Education', 30000),
(5, 7, 'Mixcloud',
'http://icons.iconarchive.com/icons/dtafalonso/yosemite-flat/512/Music-icon.png',
'Music record start up', '2018-04-03 00:00:00', 'Entertainment', 70000),
(6, 8, 'Dating Life',
'https://ih1.redbubble.net/image.30156406.3433/sticker,375x360-bg,ffffff.png', 'Dating
enhancement application the start up is focusing on', '2018-12-28 00:00:00', 'Social
Network', 50000),
(7, 9, 'Foodies Deliver',
'https://www.shareicon.net/download/2016/11/25/856584_food_512x512.png', 'Start up
doing food delivery services', '2018-12-19 00:00:00', 'Food', 60000),
(8, 1, 'Crossknowledge', 'https://png.icons8.com/ios/1600/knowledge-sharing.png',
'Education start up', '2018-06-28 00:00:00', 'Education', 37000),
(9, 10, 'Funding Circle',
'https://labs.robinhood.org/wp-content/uploads/2015/07/FUNDING.png?x94397', 'VC
company for smaller entreupreneurs', '2018-12-17 00:00:00', 'Financials', 57000),
(10, 11, 'Songkick',
'https://png.pngtree.com/element_origin_min_pic/16/08/30/2057c5817bcf859.jpg', 'Music
industry entertainment', '2018-01-03 00:00:00', 'Entertainment', 12000),
(11, 16, 'Songtify',
'https://upload.wikimedia.org/wikipedia/en/thumb/a/af/Song_logo.svg/1200px-Song_logo.svg
.png', 'Music streaming services company', '2018-12-27 00:00:00', 'Media', 80000),
(12, 19, 'Experteer',
'https://www.shareicon.net/download/2016/07/03/790220_school_512x512.png', 'Education
that offers any course.', '2019-04-26 00:00:00', 'Education', 124000),
(13, 5, 'GetJar', 'https://png.icons8.com/color/1600/cash-in-hand.png', 'Saving
investments for you', '2019-04-20 00:00:00', 'Financials', 65000),
(14, 8, 'Game3D',
'https://png.pngtree.com/element_pic/16/11/22/56551424a96d8b34d760f5c4fc338e07.jpg',
'Gaming VR company', '2020-04-12 00:00:00', 'Gaming', 140000),
(15, 6, 'Scoreloop',
'https://cdn4.iconfinder.com/data/icons/flat-icon-set/2133/flat_icons-graficheria.it-13.
png', 'Score in our games', '2018-09-08 00:00:00', 'Gaming', 48000),
(16, 7, 'Appsfire',
'http://www.clker.com/cliparts/1/0/9/e/1487324295866254597fire-vector-icon-png-27.hi.png
', 'Social media', '2018-10-25 00:00:00', 'Social Network', 9300),
(17, 21, 'Skyscanner',
'https://cdn.iconscout.com/public/images/icon/free/png-512/aeroplane-airplane-plane-air-
transportation-vehicle-pessanger-people-emoj-symbol-3306ff886517b0e9-512x512.png',
'Travelling company', '2018-08-18 00:00:00', 'Travel', 9000);

-- --------------------------------------------------------

--
-- Table structure for table `project_keywords`
--

DROP TABLE IF EXISTS `project_keywords`;
CREATE TABLE IF NOT EXISTS `project_keywords` (
```

```
  `project_id` int(11) NOT NULL,
  `keyword` varchar(32) NOT NULL,
  PRIMARY KEY (`project_id`,`keyword`),
  KEY `project_id` (`project_id`),
  KEY `keyword` (`keyword`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `project_keywords`
--

INSERT INTO `project_keywords` (`project_id`, `keyword`) VALUES
(1, 'money'),
(2, 'money'),
(3, 'delivery'),
(3, 'food'),
(4, 'coding'),
(4, 'computer'),
(5, 'coding'),
(5, 'computer'),
(6, 'dating'),
(7, 'delivery'),
(7, 'food'),
(8, 'knowledge'),
(9, 'funding'),
(10, 'music'),
(10, 'song'),
(11, 'song'),
(12, 'expert'),
(13, 'getjar'),
(14, 'game'),
(15, 'loop'),
(15, 'score'),
(16, 'app'),
(16, 'fire'),
(17, 'sky'),
(17, 'travel');

-- --------------------------------------------------------

--
-- Table structure for table `users`
--

DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(255) NOT NULL,
  `password_hash` varchar(64) NOT NULL,
  `full_name` varchar(255) NOT NULL,
  `privilege` tinyint(1) NOT NULL DEFAULT '0',
  `session_id` varchar(32) DEFAULT NULL,
  `session_exp` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`user_id`),
  UNIQUE KEY `email` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`user_id`, `email`, `password_hash`, `full_name`, `privilege`,
`session_id`, `session_exp`) VALUES
(1, 'afi@cat.com', '35932720f27c684829c6b308235f47fb3394456793fce2a61deb418058b8cf7c',
'afi', 0, NULL, NULL),
(2, 'admin@admin.com', 'password', 'admin', 1, NULL, NULL),
(3, 'email0@email.com', 'password', 'USER0', 0, NULL, NULL),
(4, 'email@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'USER1', 0, NULL,
NULL),
(5, 'email1@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'USER2', 0, NULL,
NULL),
```

```
(6, 'email2@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'USER3', 0, NULL,
NULL),
(7, 'email3@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'USER4', 0, NULL,
NULL),
(8, 'email4@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'USER5', 0, NULL,
NULL),
(9, 'email5@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'USER6', 0, NULL,
NULL),
(10, 'email6@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'USER7', 0, NULL,
NULL),
(11, 'email7@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'USER8', 0, NULL,
NULL),
(12, 'user@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Amy', 0, NULL,
NULL),
(13, 'user1@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Betty', 0, NULL,
NULL),
(14, 'candice@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Candice', 0,
NULL, NULL),
(15, 'danny@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Danny', 0, NULL,
NULL),
(16, 'elizabeth@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Elizabeth', 0,
NULL, NULL),
(17, 'faith@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Faith', 0, NULL,
NULL),
(18, 'gary@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Gary', 0, NULL,
NULL),
(19, 'harry@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Harry', 0, NULL,
NULL),
(20, 'ivy@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Ivy', 0, NULL,
NULL),
(21, 'june@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'June', 0, NULL,
NULL),
(22, 'kane@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Kane', 0, NULL,
NULL),
(23, 'zachary@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'Zachary', 0,
NULL, NULL),
(24, 'user100@email.com',
'5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8', 'user100', 0,
NULL, NULL);

--
-- Constraints for dumped tables
--

--
-- Constraints for table `fundings`
--
ALTER TABLE `fundings`
  ADD CONSTRAINT `fk_fundings_projects_project_id` FOREIGN KEY (`project_id`)
  REFERENCES `projects` (`project_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_fundings_users_user_id` FOREIGN KEY (`user_id`) REFERENCES
  `users` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `projects`
```

```
--
ALTER TABLE `projects`
  ADD CONSTRAINT `fk_projects_categories_category` FOREIGN KEY (`category`) REFERENCES
  `categories` (`category`) ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_projects_users_user_id` FOREIGN KEY (`user_id`) REFERENCES
  `users` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `project_keywords`
--
ALTER TABLE `project_keywords`
  ADD CONSTRAINT `fk_project_keywords_keywords_keyword` FOREIGN KEY (`keyword`)
  REFERENCES `keywords` (`keyword`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `fk_project_keywords_projects_project_id` FOREIGN KEY (`project_id`)
  REFERENCES `projects` (`project_id`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

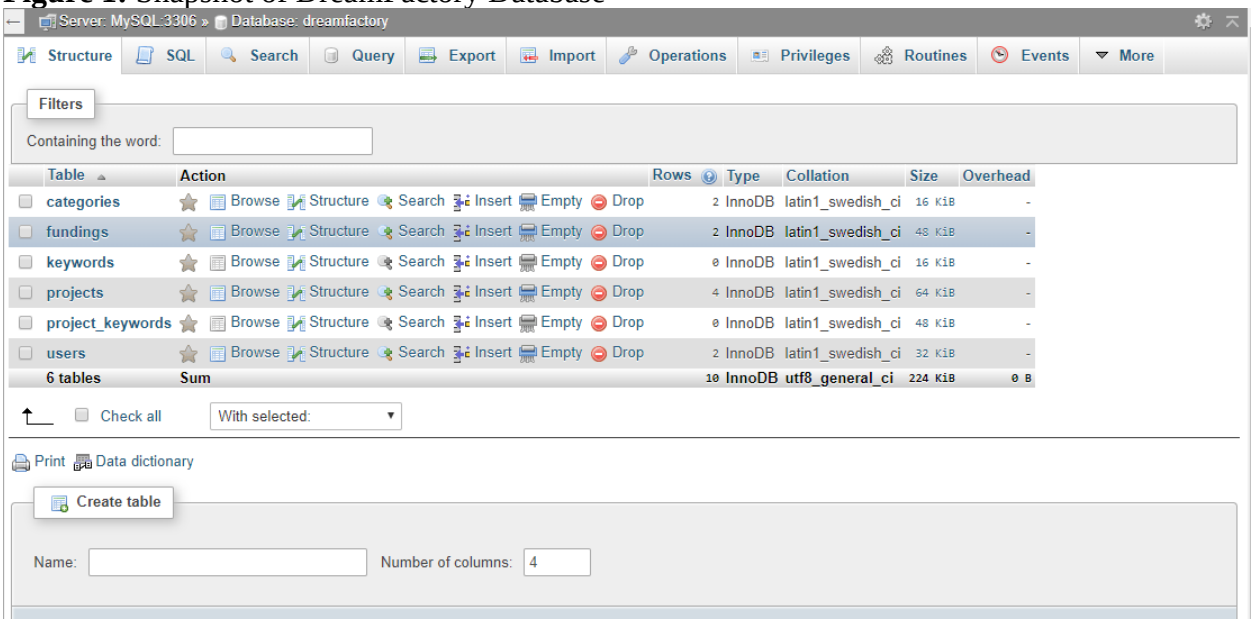
Annex B

Tables and Figures of the Main Report

Table 1: Breakdown of tiers in DreamFactory

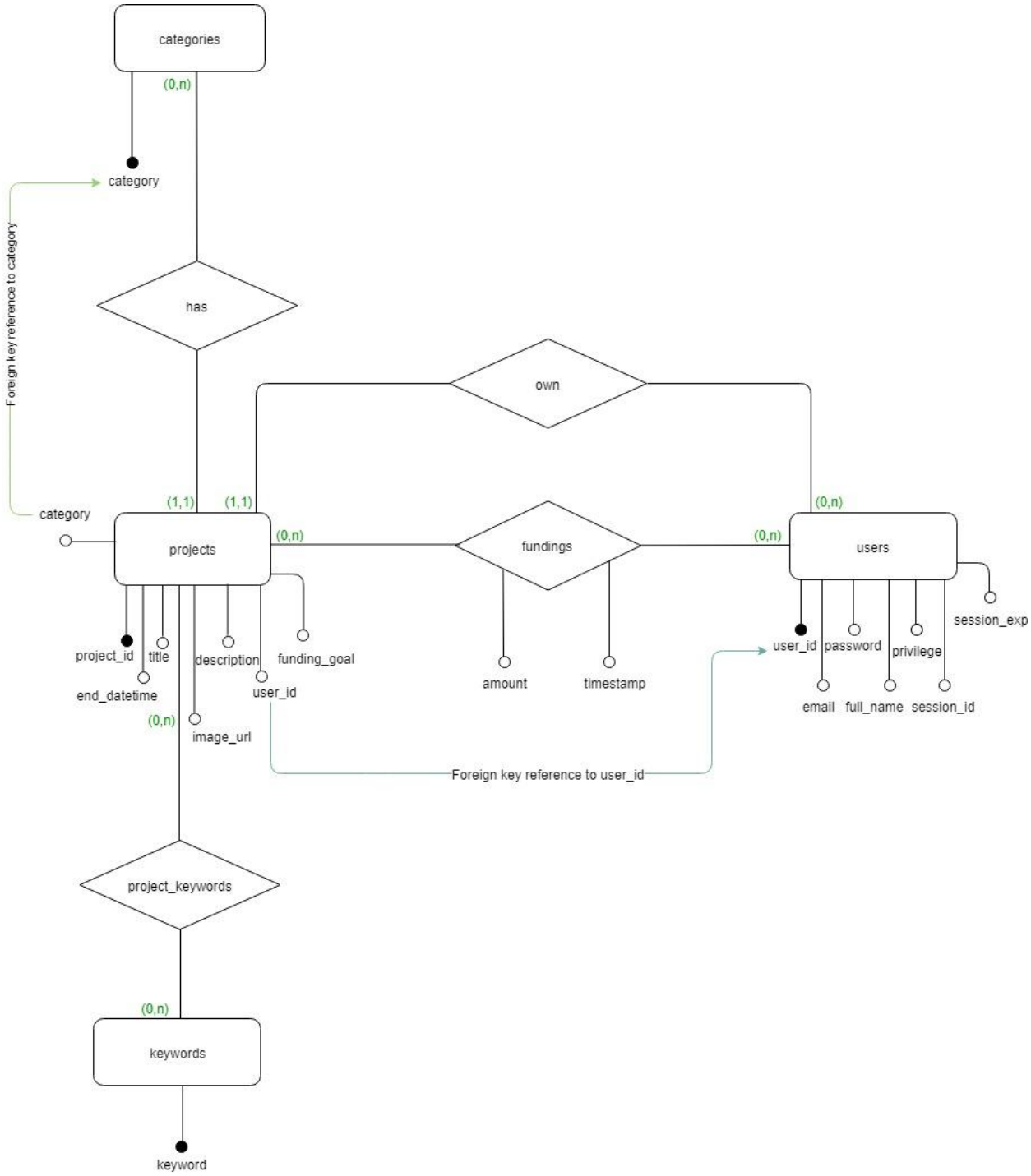
Presentation Tier	Web Page	HTML
Logic Tier	Software Components	Apache WebServer
		PHP, HTML5, Javascript
Database Tier	Database	MySQL
		PHPMyAdmin

Figure 1: Snapshot of DreamFactory Database



Annex C

Entity Relationship Diagram



Annex D

Webpage Screenshots

Figure 2: Homepage

Figure 3: Ongoing Projects displayed in Homepage

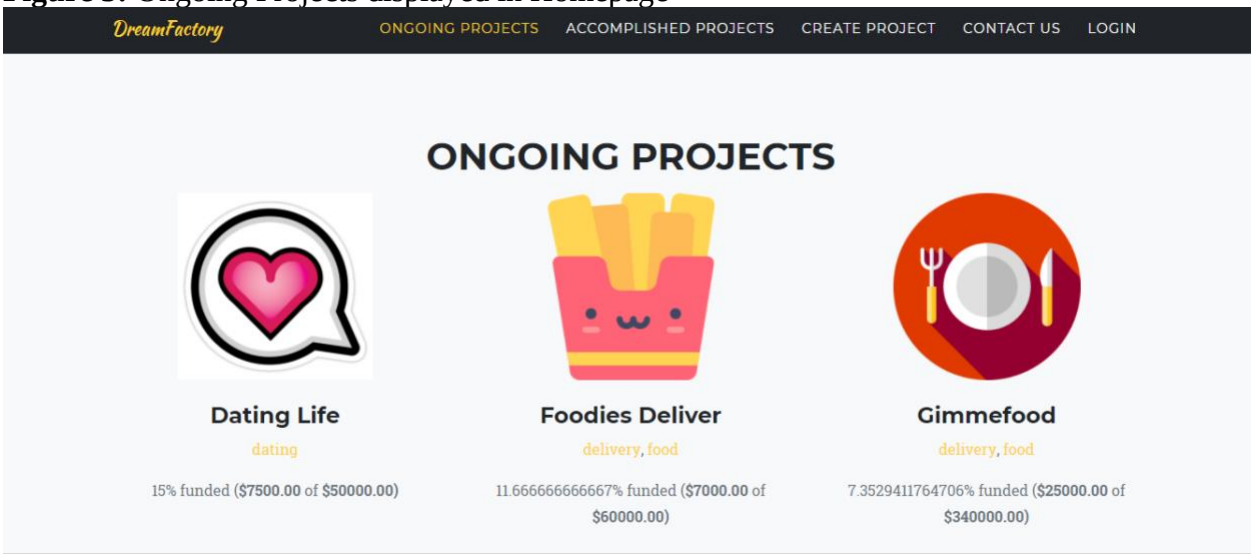


Figure 4: Search and Ongoing Projects in Browse Projects Page

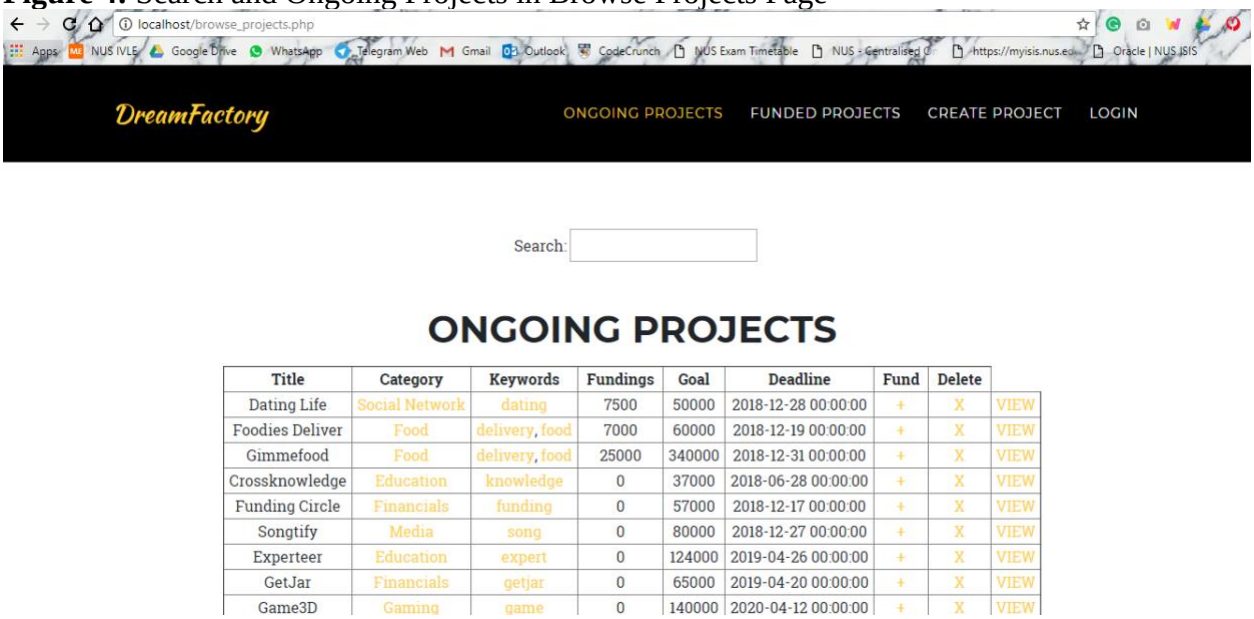


Figure 5: Funded Projects in Browse Projects Page

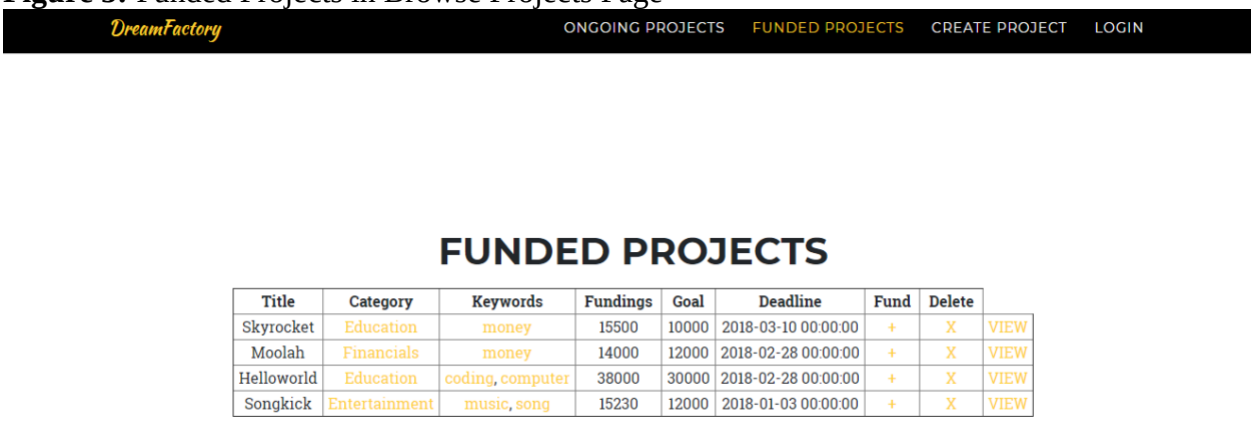


Figure 6: Login Page

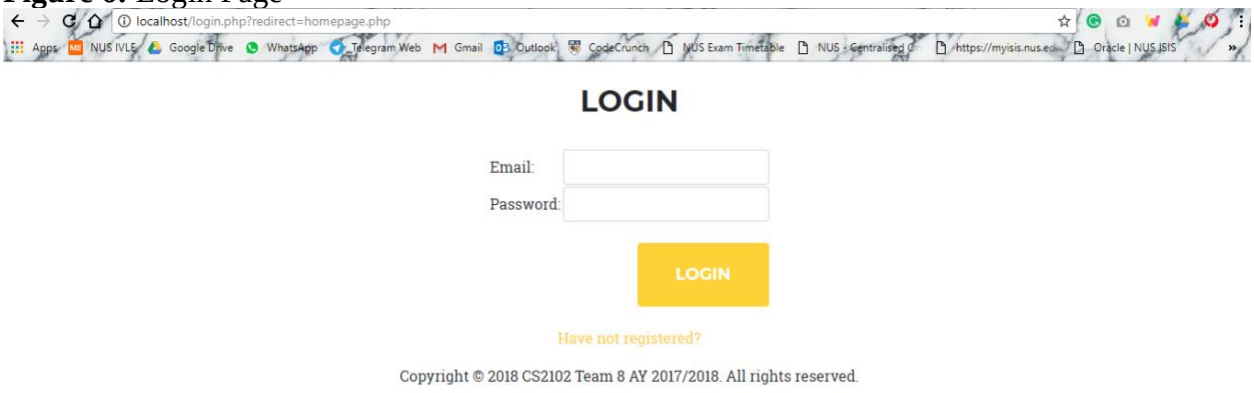


Figure 7: Register Page

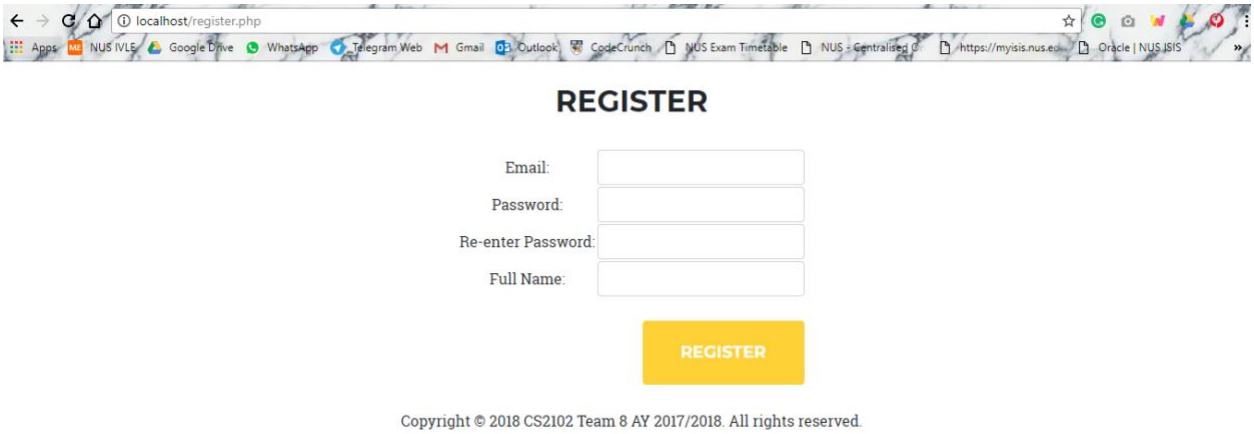


Figure 8: Create Project

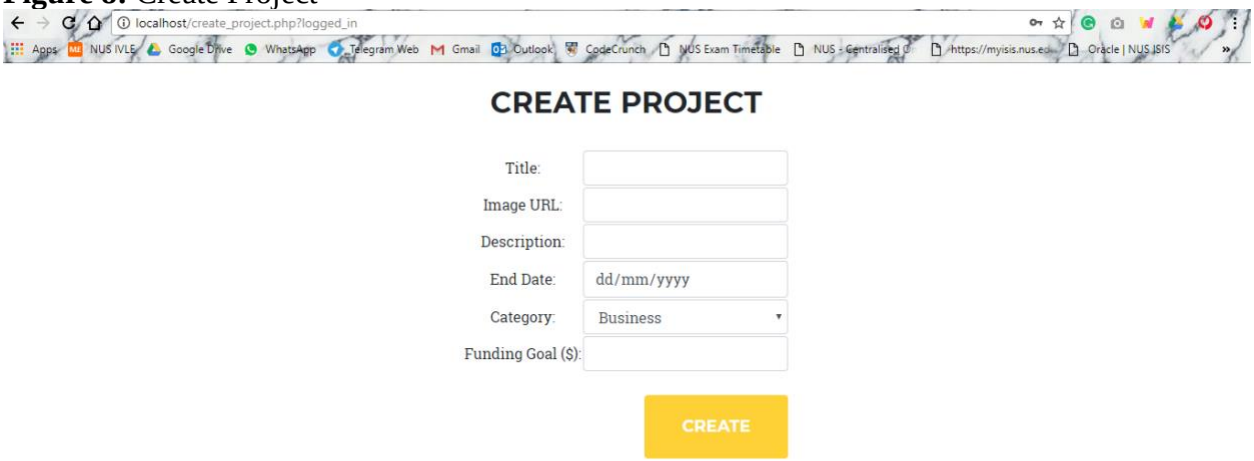


Figure 9: Funding Page of Selected Project

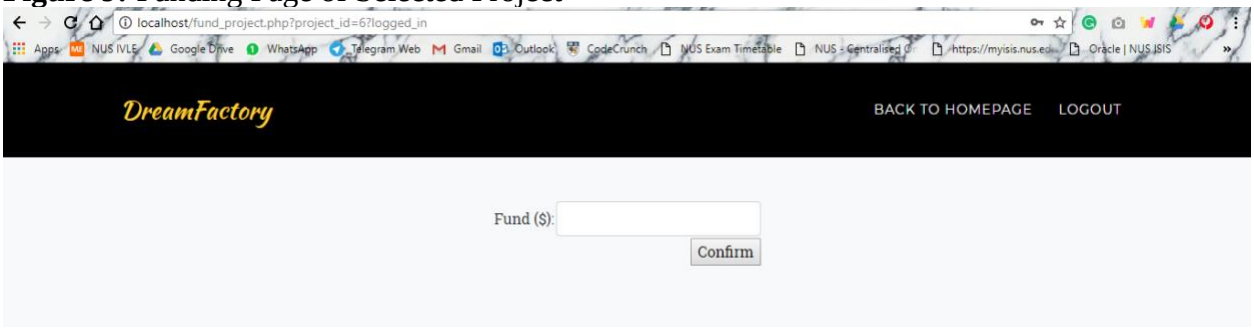


Figure 10 : View Project

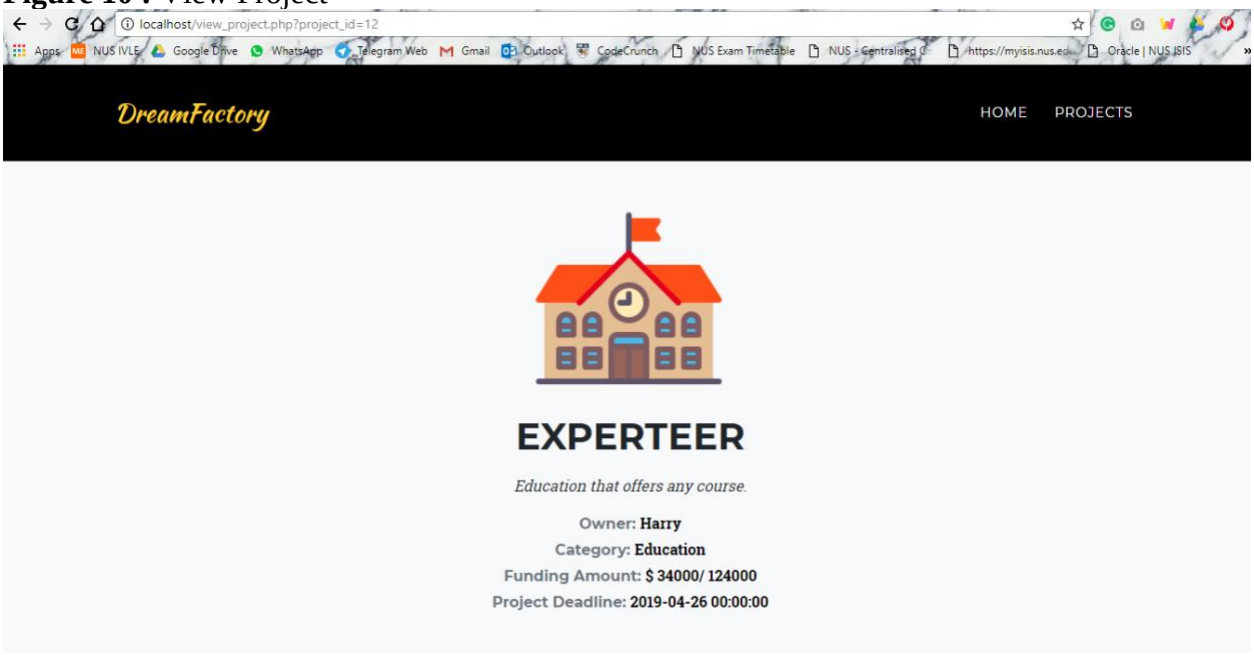
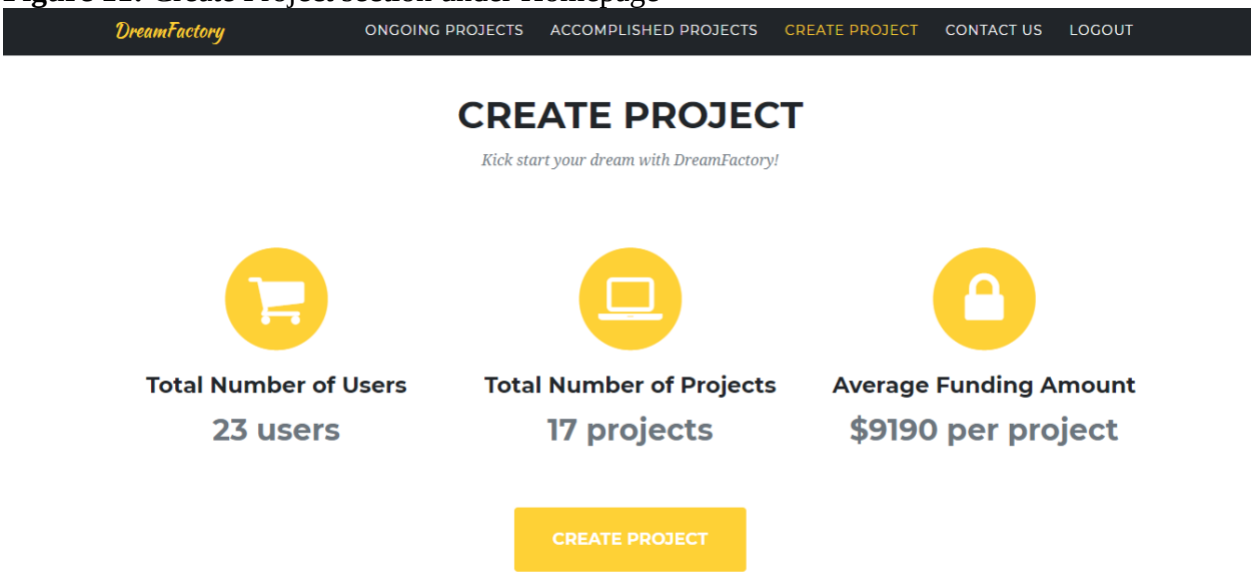


Figure 11: Create Project section under Homepage



Annex E

Data Manipulation Language Script

Figure 12: Funded Projects DML Snippet

```
SELECT
    p.project_id,
    title,
    category,
    kl.keyword_links AS keyword_links,
    COALESCE(f.fundings, 0) AS funding_amount,
    funding_goal,
    end_datetime
FROM
    projects p LEFT JOIN
    (SELECT
        project_id,
        GROUP_CONCAT(CONCAT(keyword) SEPARATOR ', ') AS keyword_links
    FROM
        project_keywords
    GROUP BY
        project_id) AS kl ON p.project_id = kl.project_id LEFT JOIN
    (SELECT
        project_id,
        SUM(amount) AS fundings
    FROM
        fundings
    GROUP BY
        project_id) AS f ON p.project_id = f.project_id
WHERE
    p.end_datetime <= CURDATE() AND
    COALESCE(f.fundings, 0) >= p.funding_goal
ORDER BY
    end_datetime DESC;
```

Figure 13: Ongoing Projects DML Snippet

```
SELECT
    p.project_id,
    title,
    category,
    kl.keyword_links AS keyword_links,
    COALESCE(f.fundings, 0) AS funding_amount,
    funding_goal,
    end_datetime
FROM
    projects p LEFT JOIN
    (SELECT
        project_id,
        GROUP_CONCAT(CONCAT('<a href=?keyword=', keyword, '>', keyword, '</a>')
        SEPARATOR ', ') AS keyword_links
    FROM
        project_keywords
    GROUP BY
        project_id) AS kl ON p.project_id = kl.project_id LEFT JOIN
    (SELECT
        project_id,
        SUM(amount) AS fundings
    FROM
        fundings
    GROUP BY
        project_id) AS f ON p.project_id = f.project_id
WHERE
    p.end_datetime > CURDATE()
ORDER BY
    COALESCE(f.fundings, 0) * 1.0 / funding_goal DESC;
```