User Manual

of
AppsoFluna SimpleApps

written by Charaka Gunatillake
for BCS Professional Graduate Diploma Project

Introduction

Here it's described the functions of core features by creating a sample application for educational institutes. The document has 4 sections.

1. **Sample Case**
2. **Creating application (for regular users)**
3. **Running it live (for regular users)**
4. **Generating source code (for developers)**

1.Sample Case

Problem

Here's the problem that will be solved by the sample application. A **school** needs an system to manage their **courses**, **students** and **registrations**. One **student** may register for several **courses**.

The system needs to keep the **course names** for **courses**, and **first name**, **last name** and **address** of each **student**. The **students** must be registered for each **course** they follow. Therefore the **student**, **course** and the **date of registration** must be recorded for each **registration**.

Only **administrators** must be able to **create**, **edit** or **delete courses**. **Administrators** can also **create**, **edit**, **delete** other records such as **students** and **registrations**. In short, they can **create**, **edit**, **delete any** record. **Officers** can **create**, **edit** and **delete** students and **registration** details. However **officers** can only view **courses**. They are **NOT** allowed to **create**, **edit** or **delete courses**.

Break down

The application will have 3 *items* called **Course**, **Student** and **Registration** and 2 *roles* called **Administrator** and **Officer**.

Items:

- **Student** *item* will have 3 *fields* called **First name**, **Last name** and **Address**.
- **Course** *item* will have one *field* called **Course name**.
- **Registration** *item* will have 3 *fields* called **Student**, **Course** and **Date of registration**.

Roles:

- Users with **Administrator** *role* can **view/create/edit/delete** anything without any restriction.
- Users with **Officer** *role* can also **view/create/edit/delete Student** and **Registration** records.
- Users with **Officer** *role* can only **view** courses (**create/edit/delete NOT** allowed).

2.Creating application

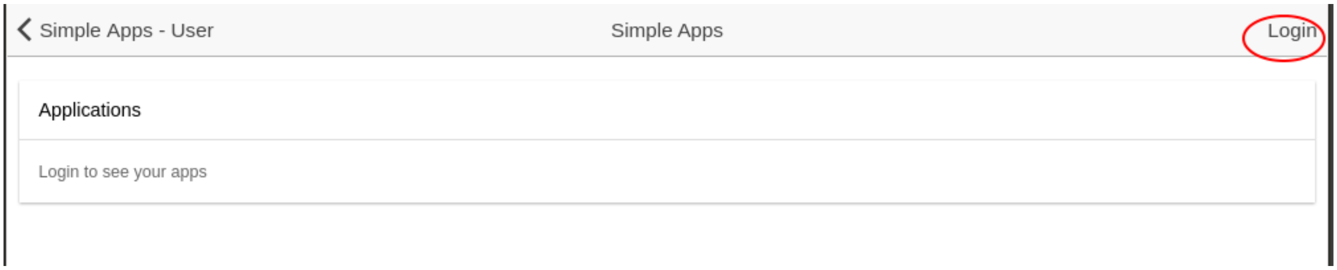
Getting Started

Before starting the **SimpleApps** must be running. If you are running the application on localhost, make sure it's loaded and visit it. The prototype version (hosted in Openshift) can be found here: **http://simpleapps-appsofluna.rhcloud.com/**.

For further details, visit **http://www.appsofluna.com/**.

How to login

1.Click **Login**



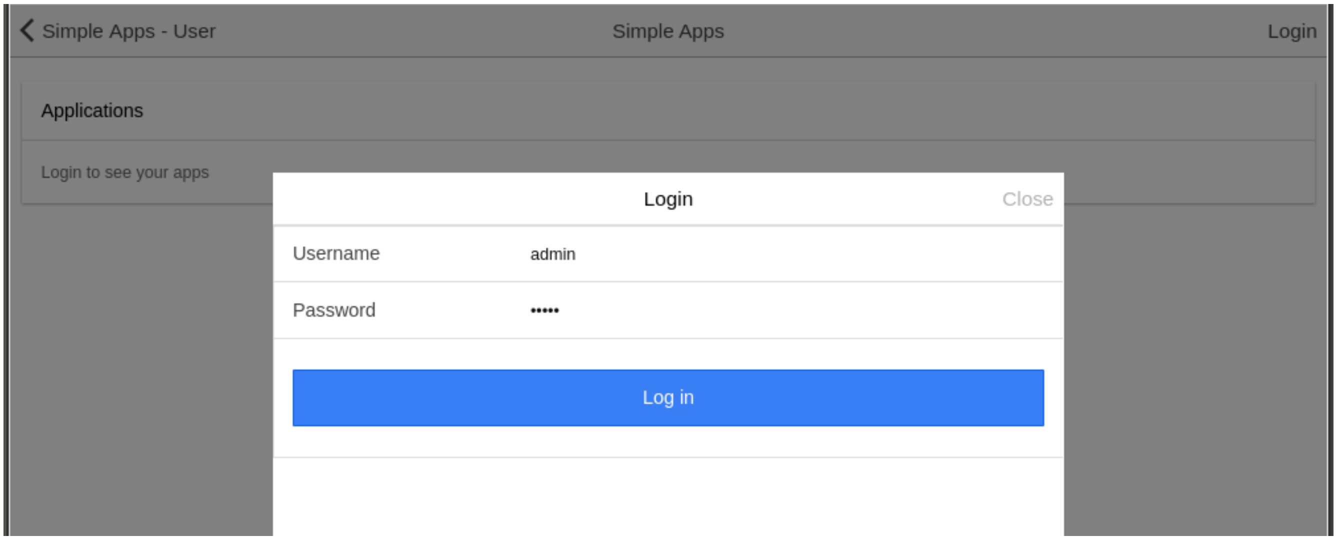
Applications

Login to see your apps

FIGURE 1 – INTIAL SCREEN

2.Enter username and password

Default username: **admin**
Default password: **guess**



Username admin

Password •••••

FIGURE 2 – LOGIN SCREEN

3.Click **Log in**.

How to create an app

1.Click **plus sign(+)** in **Applications** section

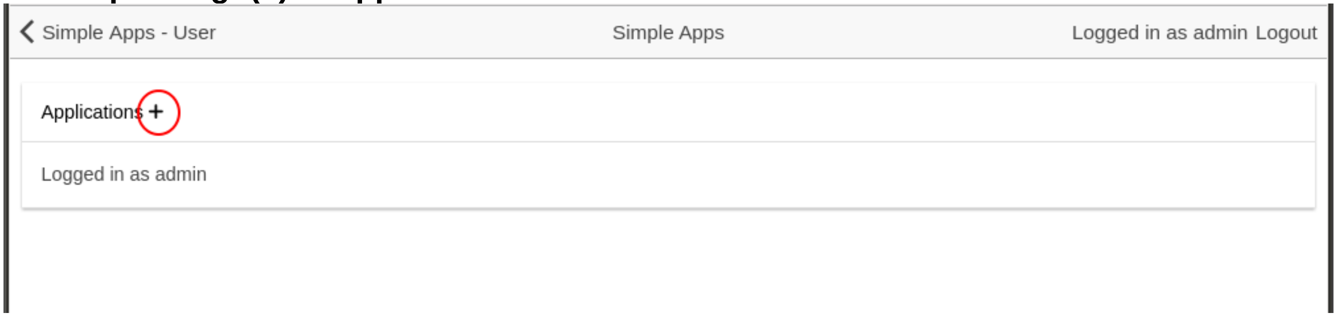
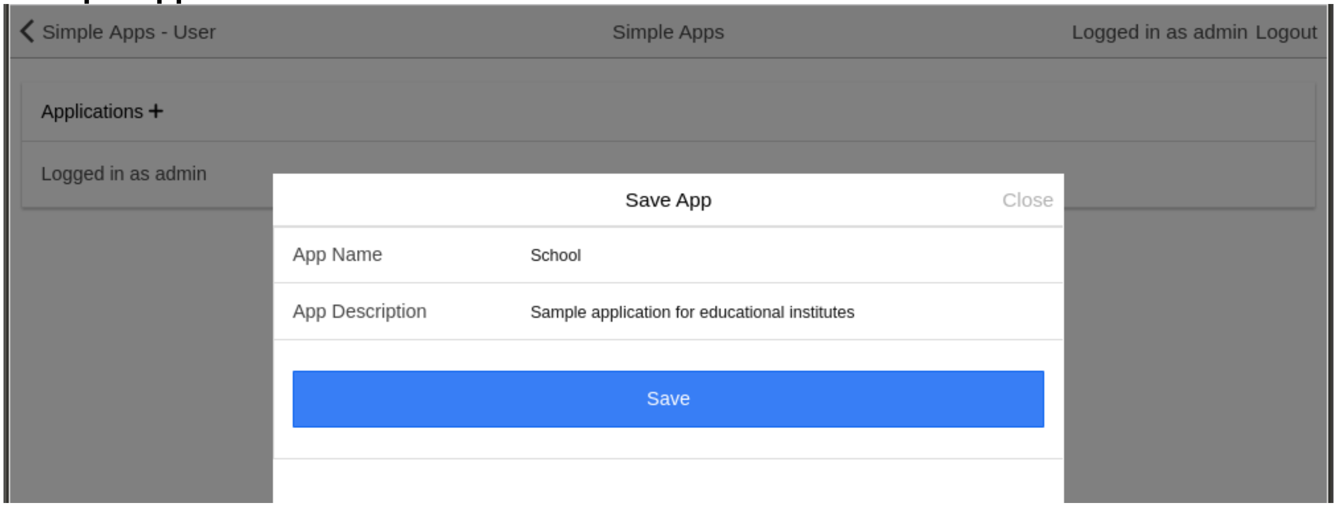


FIGURE 3 – APPLICATION PLUS SIGN

2.Enter app **name** and **description** and Click **Save**

Here for the sample application we use the name “**School”** and description as “**Sample application for educational institutes”**



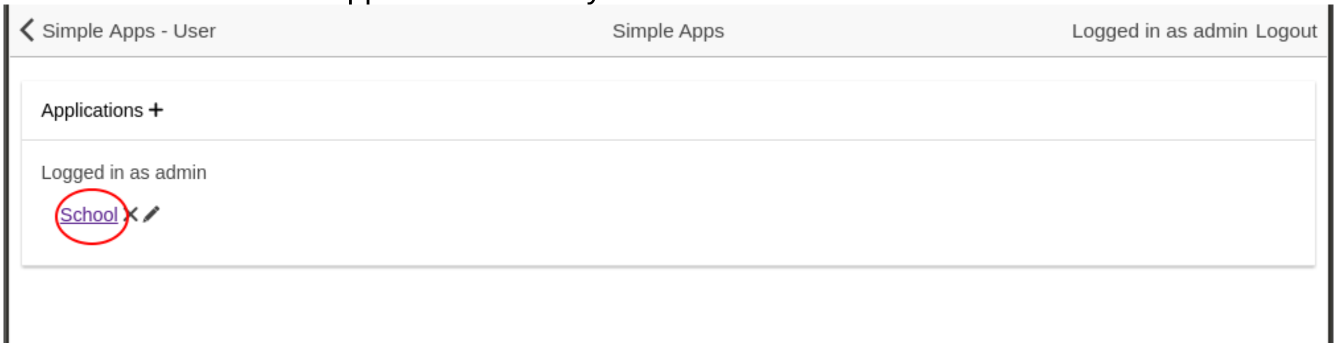
App Name School

App Description Sample application for educational institutes

FIGURE 4 – SAVE APP SCREEN

How to add items

1.Click on the relevant application where you need to add an item



Simple Apps - User

Simple Apps

Logged in as admin Logout

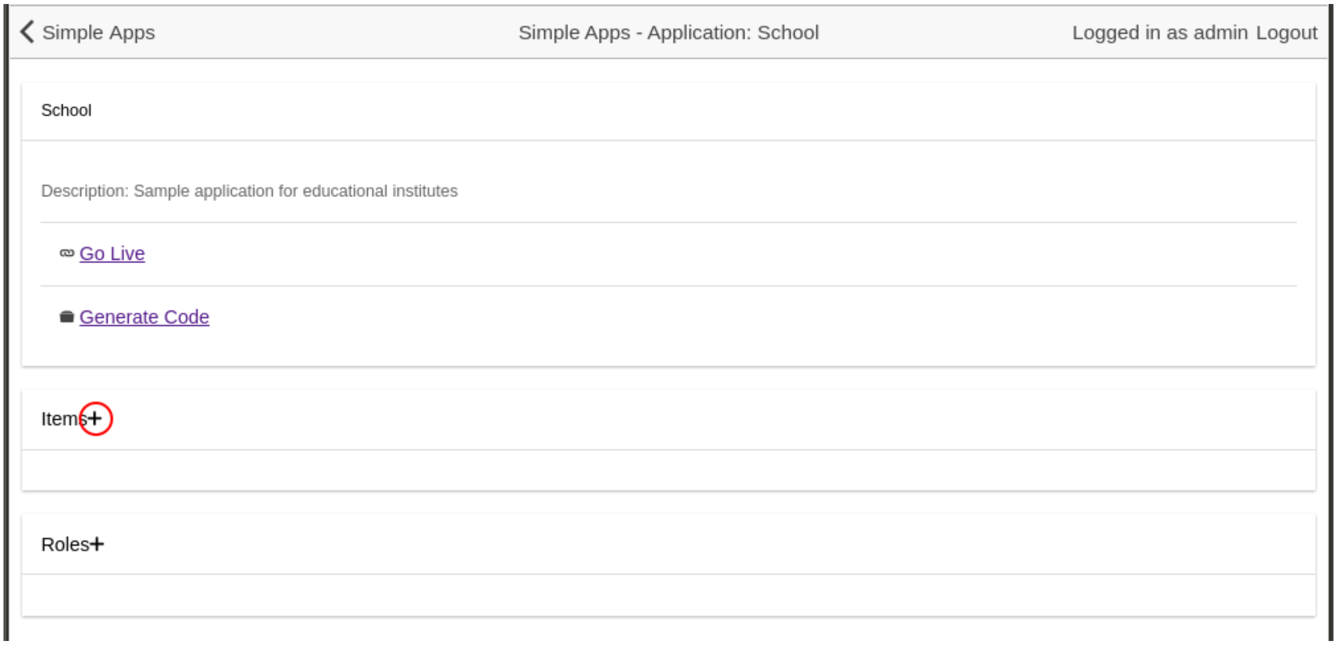
Applications +

Logged in as admin

School ✕ ✎

FIGURE 5 – CLICK APPLICATION

2.Click **Plus(+) sign** under **Items** section



Simple Apps

Simple Apps - Application: School

Logged in as admin Logout

School

Description: Sample application for educational institutes

Go Live

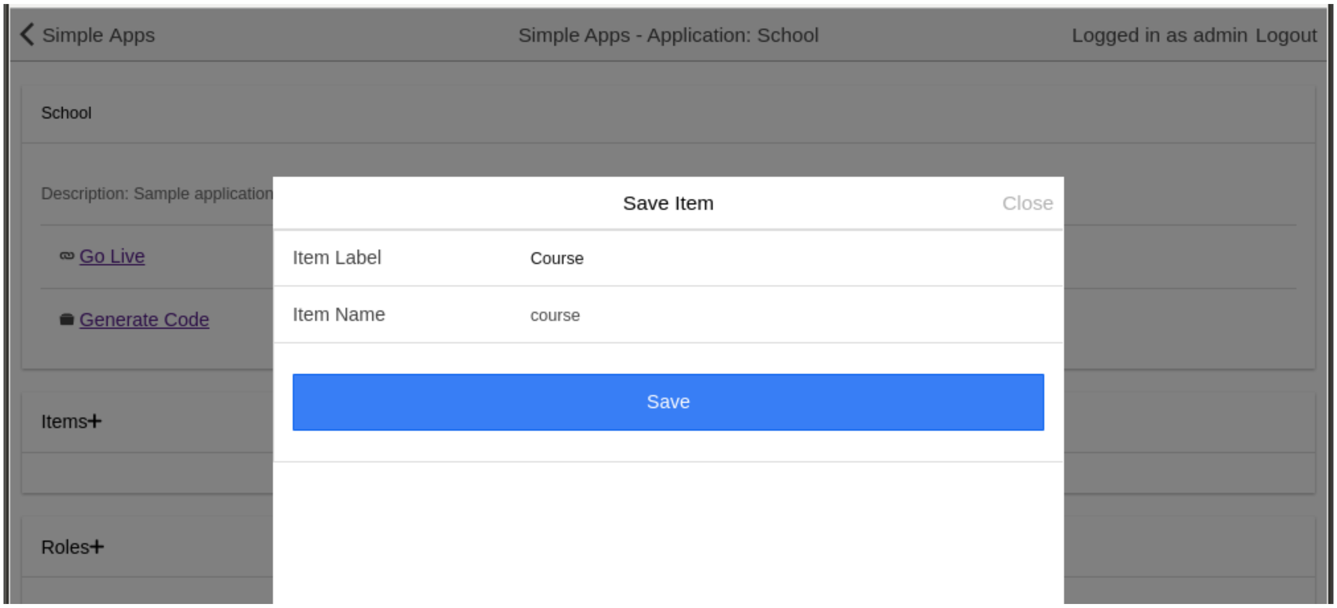
Generate Code

Items+

Roles+

FIGURE 6 – ITEMS PLUS SIGN

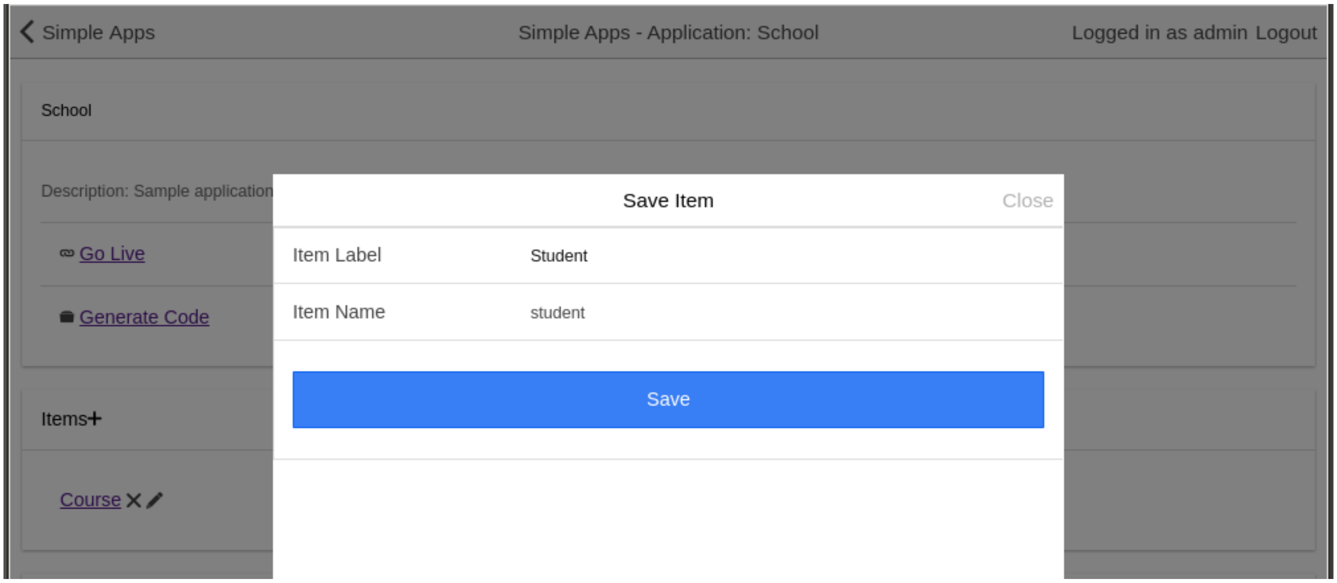
3.Enter a **label** for the item and Click **Save**
For the sample application we need to create 3 items called Course, Student and Registration as below.


Simple Apps
Simple Apps - Application: School
Logged in as admin Logout
School
Description: Sample application
Go Live
Generate Code
Items+
Roles+
Save Item
Close
Save

Item Label: Course

Item Name: course

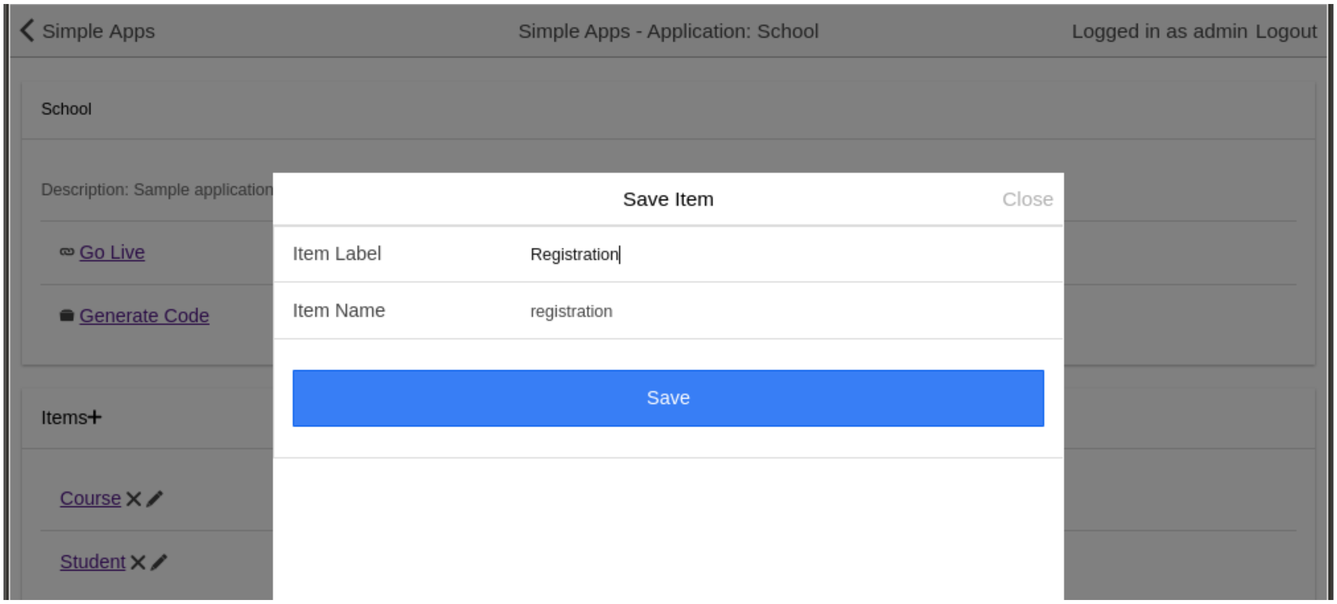
FIGURE 7 – SAVE ITEM SCREEN (FOR COURSE)


Simple Apps
Simple Apps - Application: School
Logged in as admin Logout
School
Description: Sample application
Go Live
Generate Code
Items+
Course
Save Item
Close
Save

Item Label: Student

Item Name: student

FIGURE 8 – SAVE ITEM SCREEN (FOR STUDENT)


Simple Apps
Simple Apps - Application: School
Logged in as admin Logout
School
Description: Sample application
Go Live
Generate Code
Items+
Course
Student
Save Item
Close
Save

Item Label: Registration

Item Name: registration

FIGURE 9 – SAVE ITEM SCREEN (FOR REGISTRATION)

How to add fields to an item

1.Click on the relevant item where you need to add a field
Here in sample application, we first go to **Course** item.

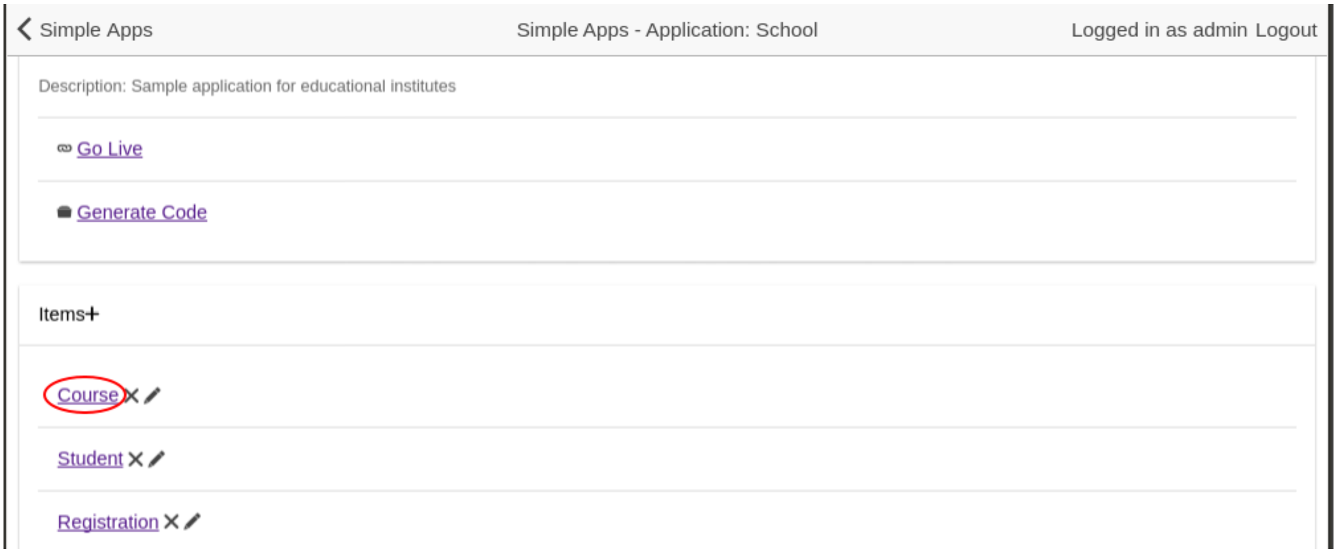

Simple Apps
Simple Apps - Application: School
Logged in as admin Logout
Description: Sample application for educational institutes
Go Live
Generate Code
Items+
Course
Student
Registration

FIGURE 10 – CLICK ITEM (FOR COURSE)

2.Click **Plus(+)** sign under **Fields** section

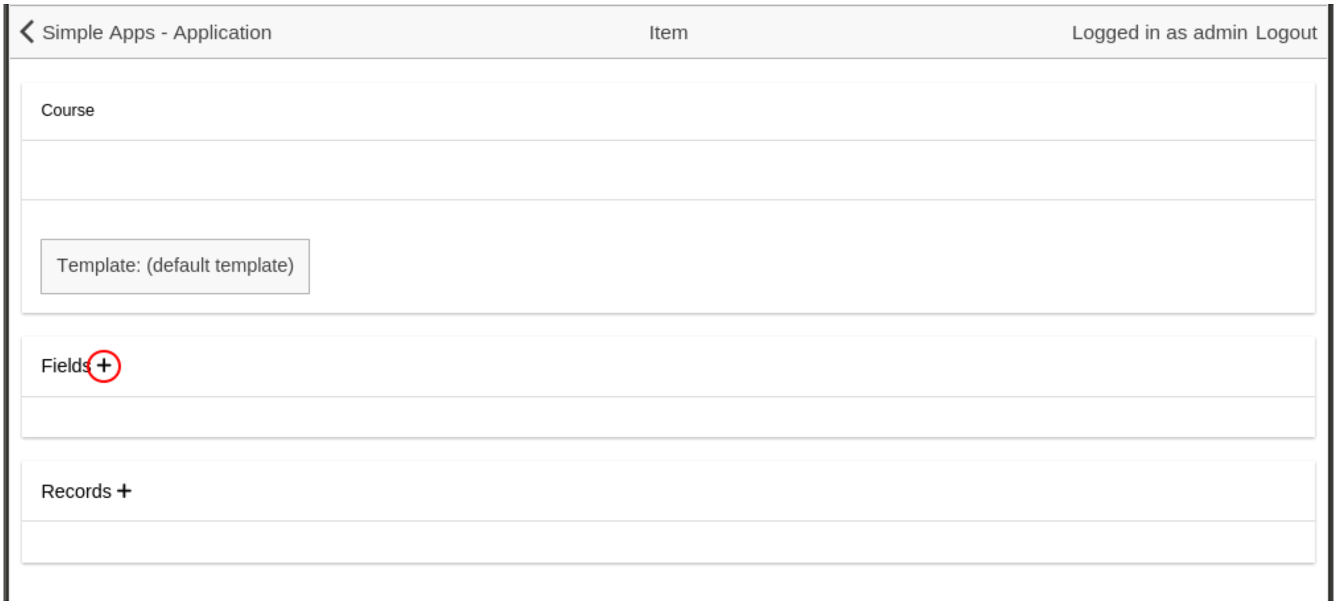
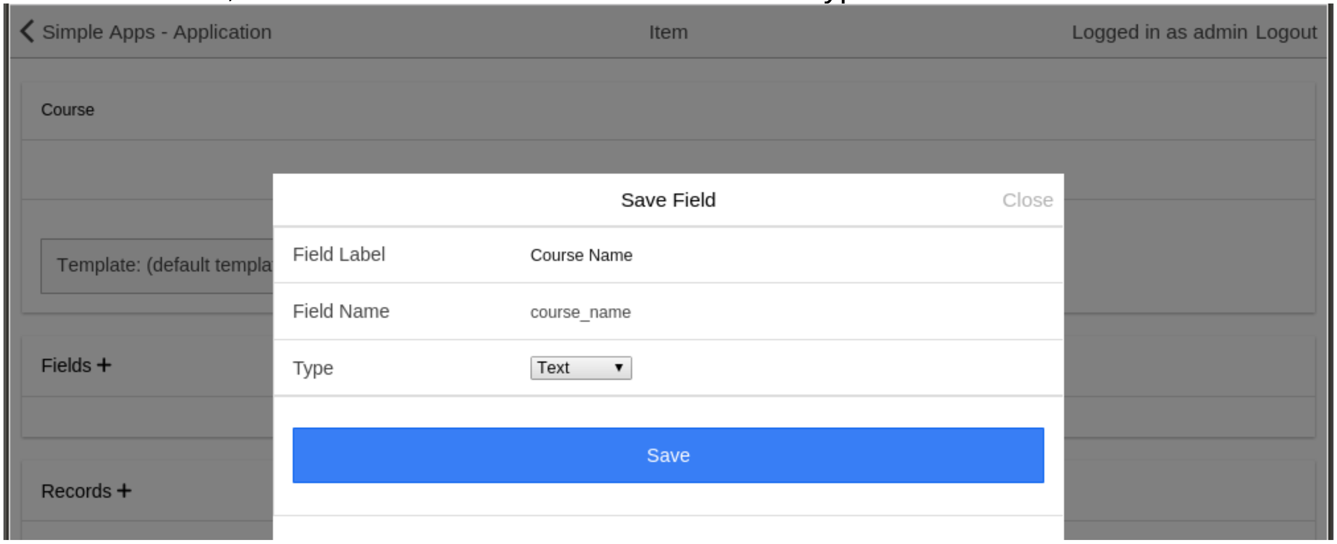

Simple Apps - Application
Item
Logged in as admin Logout
Course
Template: (default template)
Fields +
Records +

FIGURE 11 – FIELDS PLUS SIGN

3.Enter a **label**, select field **type**, enter other options if necessary and Click **Save**.
For course item, we first add **Course name** field with the type **text**.


Simple Apps - Application
Item
Logged in as admin Logout
Course
Template: (default templa
Fields +
Records +
Save Field
Close
Save

Field Label Course Name

Field Name course_name

Type Text ▾

FIGURE 12 – SAVE FIELD SCREEN (FOR COURSE NAME)

Note that for certain field types such as ***range***, ***selection*** and ***item***, you will need to add extra fields. The field type item will be covered below.

You can go back to application screen by clicking back button and select a different item.

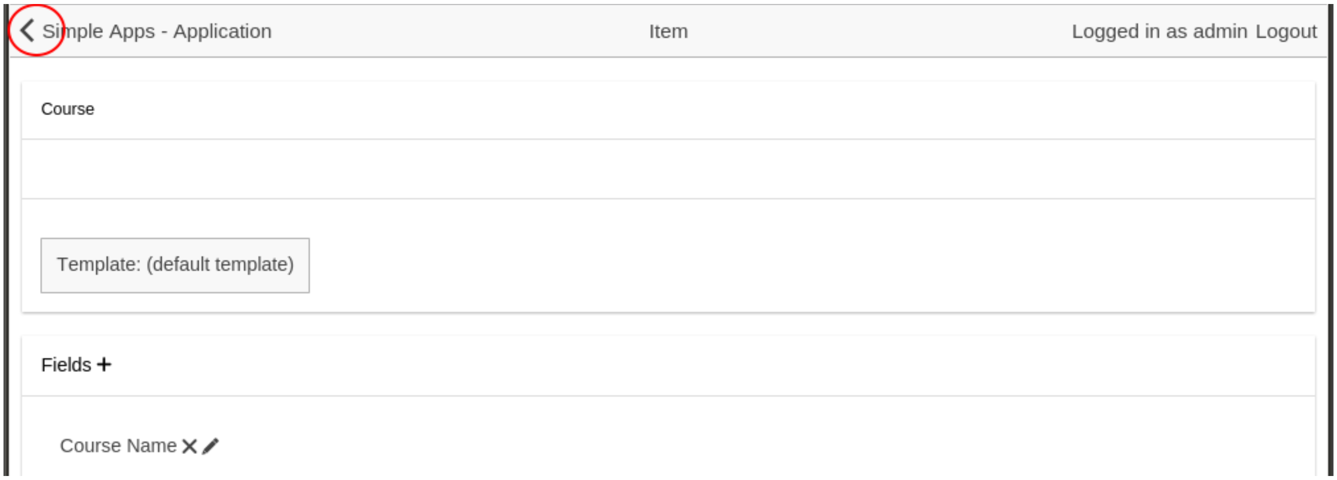
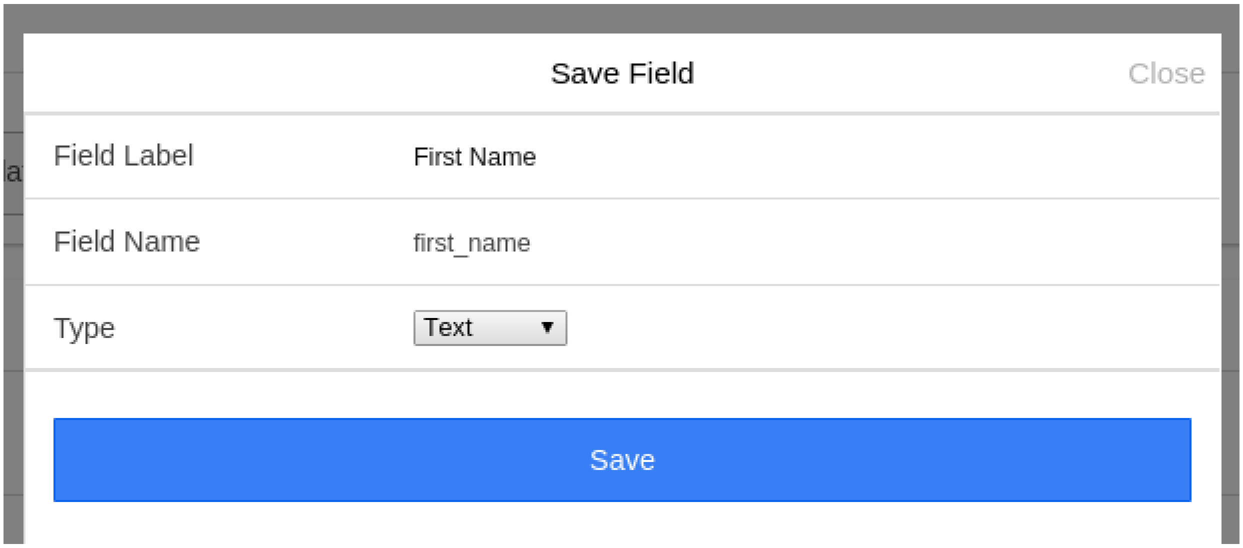

Simple Apps - Application
Item
Logged in as admin Logout
Course
Template: (default template)
Fields +
Course Name

FIGURE 13 – BACK BUTTON

There following the first 3 steps mentioned above, you can add **First name**, **Last name** and **Address** fields under **Student** item.

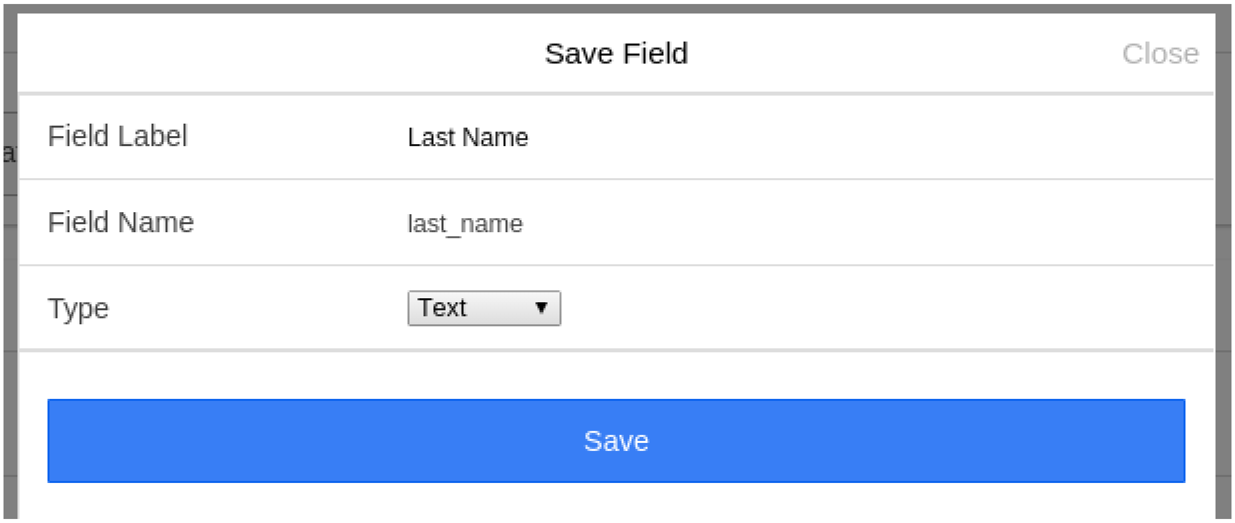


Field Label First Name

Field Name first_name

Type Text ▾

FIGURE 14 – SAVE FIELD SCREEN (FOR FIRST NAME)

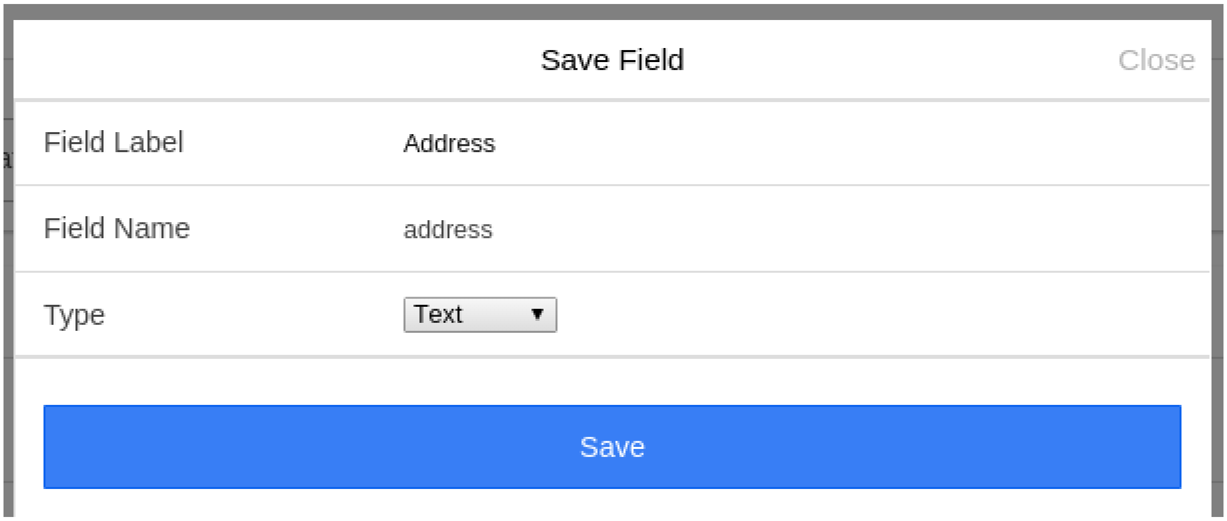


Field Label: Last Name

Field Name: last_name

Type: Text ▼

FIGURE 15 – SAVE FIELD SCREEN (FOR LAST NAME)



Field Label: Address

Field Name: address

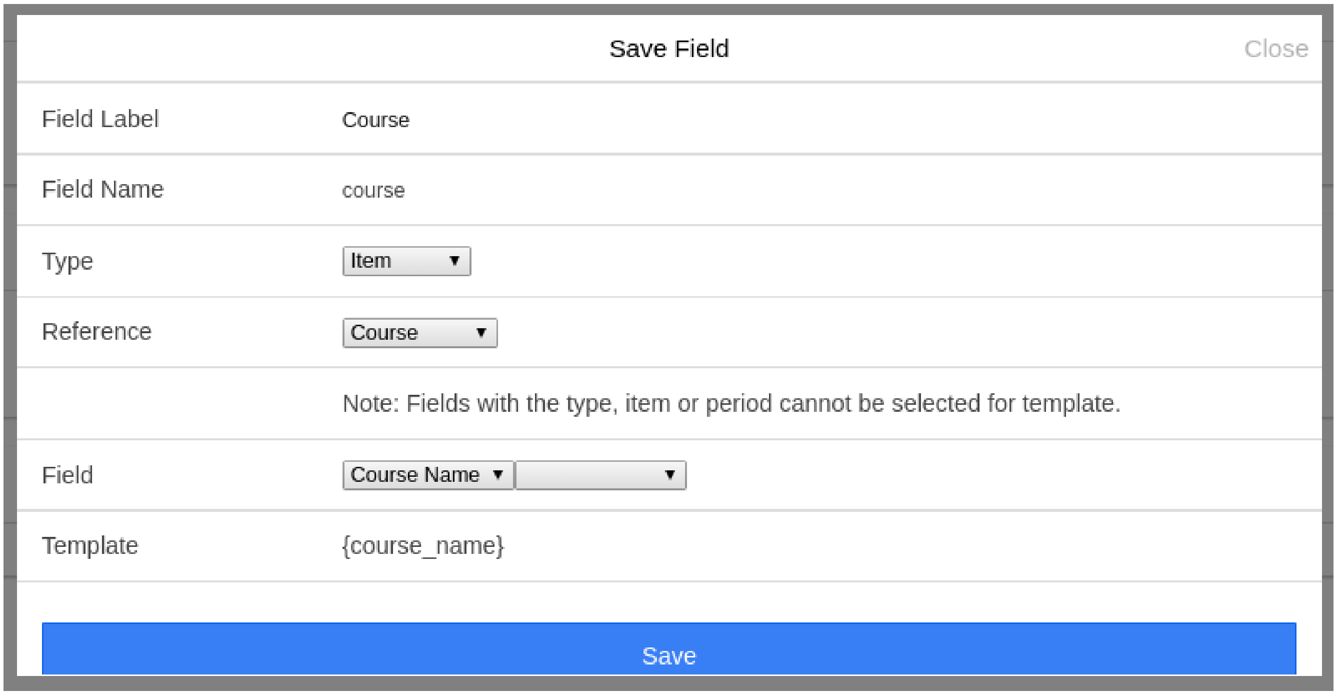
Type: Text ▼

FIGURE 16 – SAVE FIELD SCREEN (FOR ADDRESS)

Then under **Registration** item, you will need to add **Course**, **Student** and **Date of registration** fields.

Note that the **Course** and **Student** fields are references to other items. Therefore the field type must be **item**. Once the **type** is selected, it will show other option called **reference**.

As you may notice, once you select the reference as **Course**, you will need to select the fields which will be used to represent the **Course** records. For the sample application, we will be using **Course name** which is its only field.



Field Label: Course

Field Name: course

Type: Item

Reference: Course

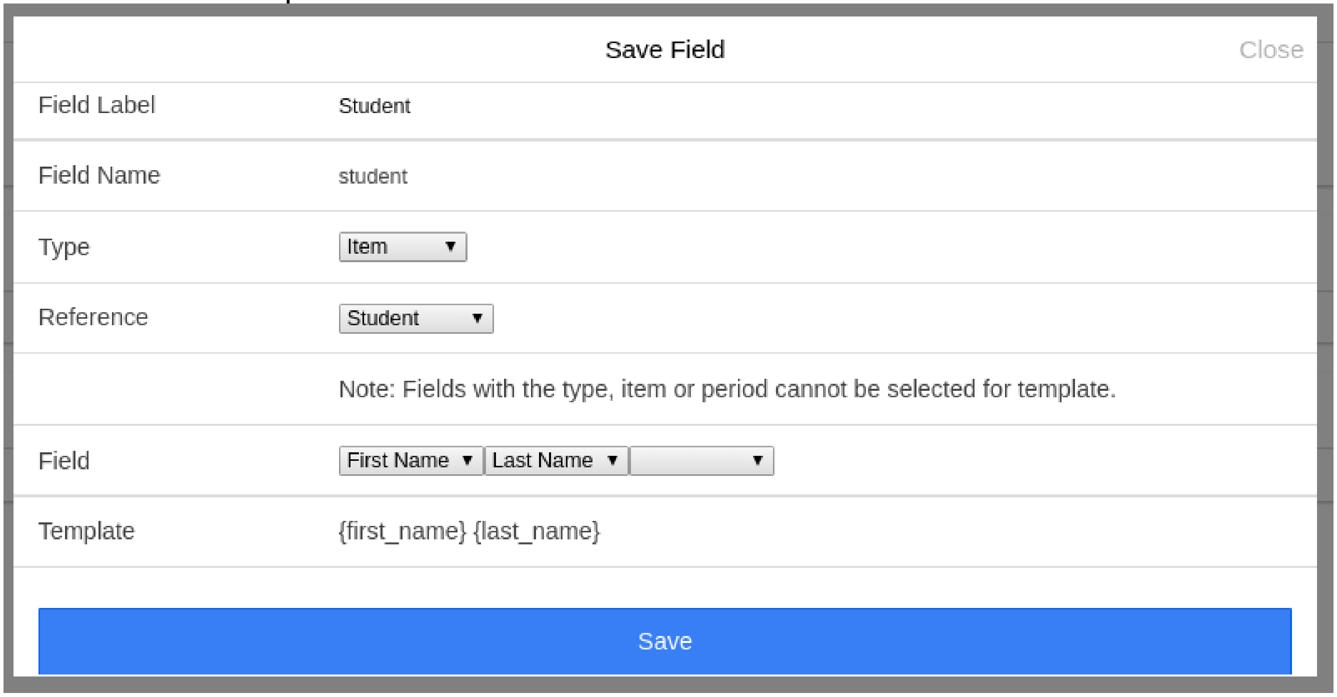
Note: Fields with the type, item or period cannot be selected for template.

Field: Course Name

Template: {course_name}

FIGURE 17 – SAVE FIELD SCREEN (FOR COURSE)

Similarly you should be able to add the **Student** field under **Registration** item. Make sure you select **item** as **type**, **Student** as the **Reference** and **First name** and **Last name** under as fields to represent a **Student** record.



Field Label: Student

Field Name: student

Type: Item

Reference: Student

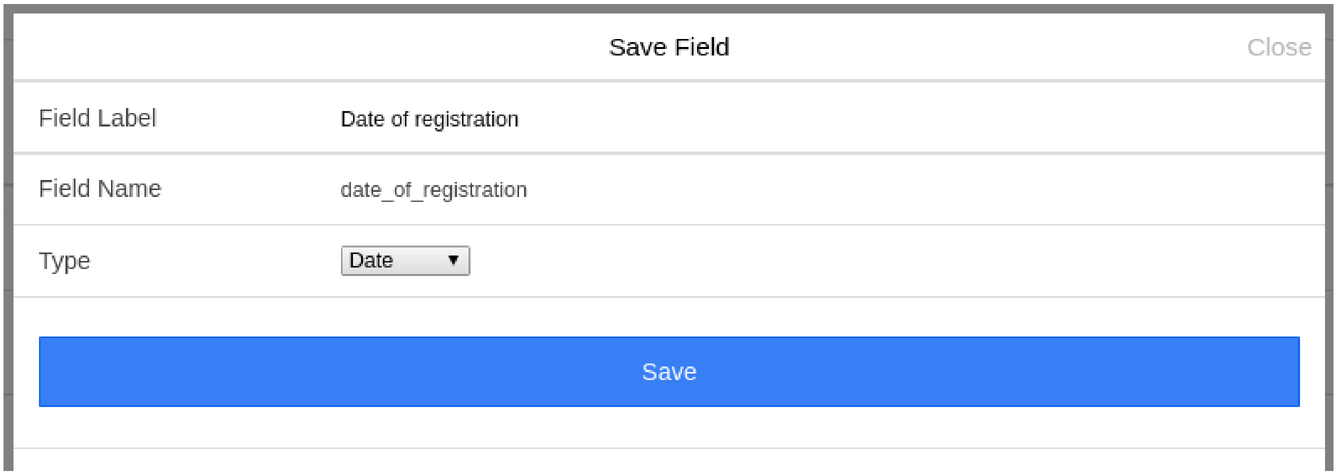
Note: Fields with the type, item or period cannot be selected for template.

Field: First Name, Last Name

Template: {first_name} {last_name}

FIGURE 18 – SAVE FIELD SCREEN (FOR STUDENT)

Finally you need to add **Date of registration** field, there make sure you select the field type as **date**.



Save Field

Field Label: Date of registration

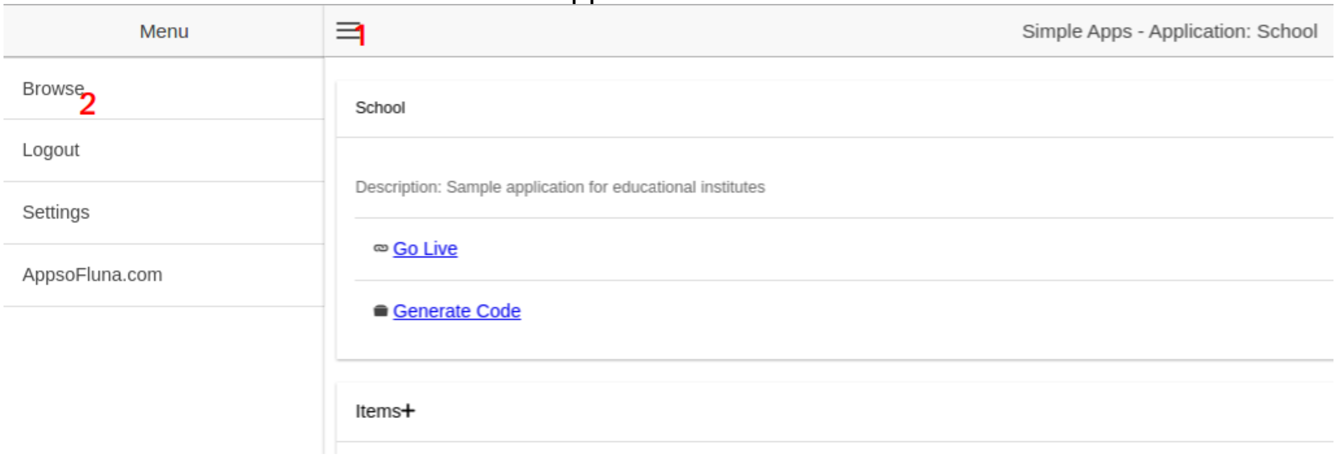
Field Name: date_of_registration

Type: Date

FIGURE 19 – SAVE FIELD SCREEN (FOR DATE OF REGISTRATION)

How to add a role to an application

1.Make sure you are in the app page where you need to add a role for. If not, you can move to main screen and select the relevant app.

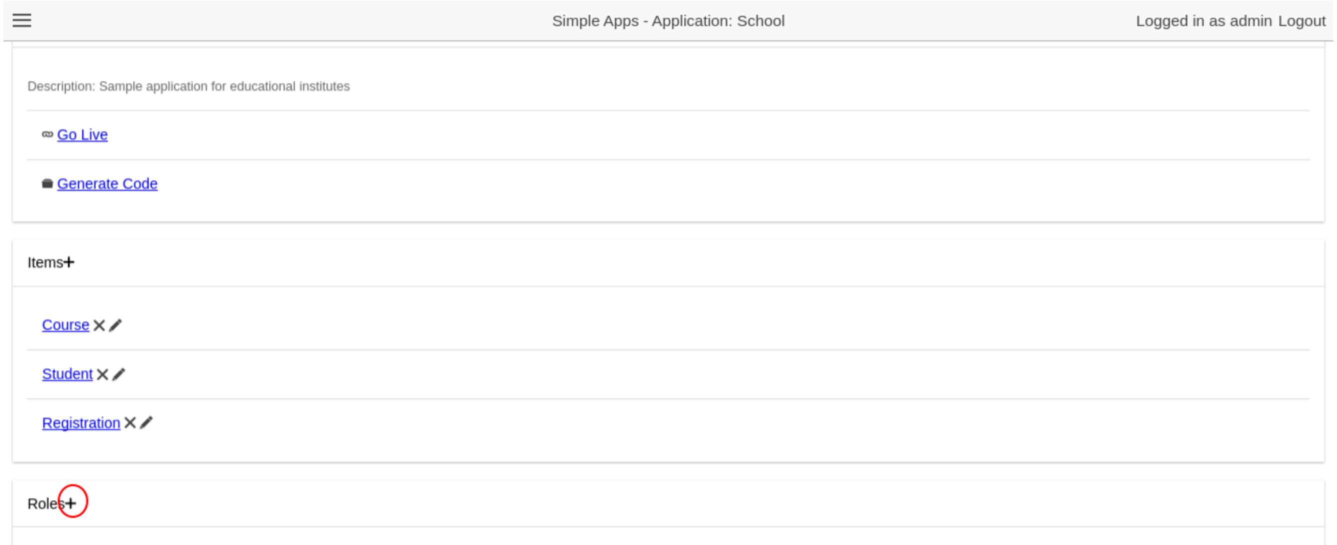


School

Description: Sample application for educational institutes

FIGURE 20 – MOVING TO HOME

2.Click **Plus(+) sign** under **Roles** section



Simple Apps - Application: School

Logged in as admin Logout

Description: Sample application for educational institutes

Go Live

Generate Code

Items

Course

Student

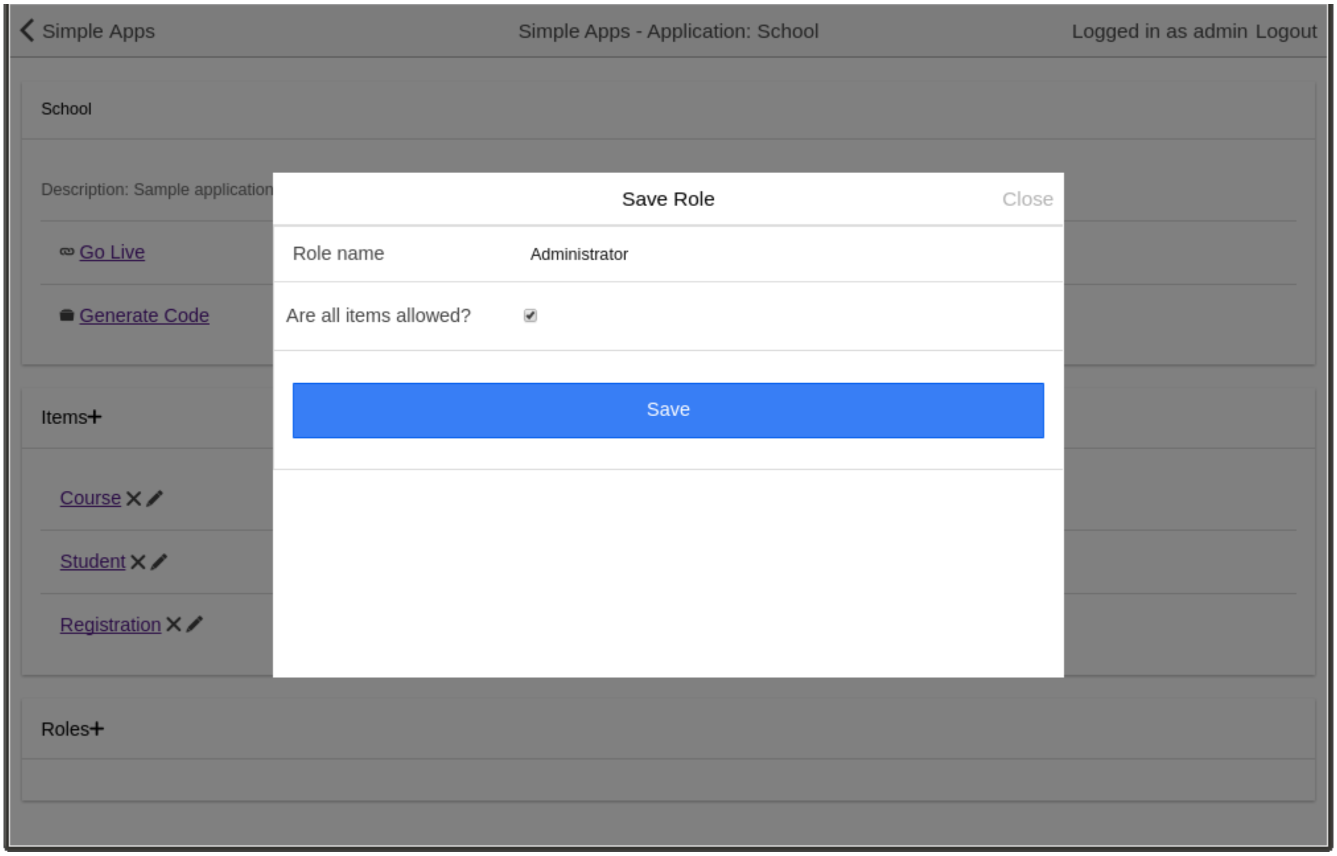
Registration

Roles

FIGURE 21 – ROLE PLUS SIGN

3.Enter a **name** for the role, and select whether ***all items are allowed*** for the **role**. If the option is not selected, permissions need to be set as described later.

For the sample application, we will first add **Administrator** role. Since the **Administrator** has all the permissions, the option *“**Are all items allowed?**”* must be checked.



Simple Apps

Simple Apps - Application: School

Logged in as admin Logout

School

Description: Sample application

Go Live

Generate Code

Items

Course

Student

Registration

Roles

Save Role

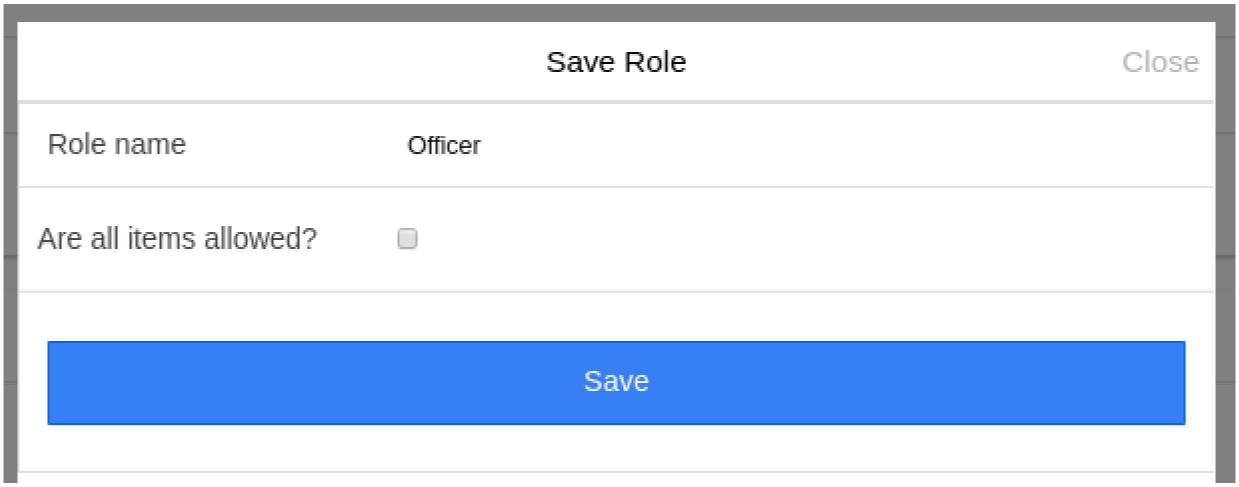
Close

Role name Administrator

Are all items allowed? ☑

FIGURE 22 – SAVE ROLE (ADMINISTRATOR)

Secondly we will add **Officer** role. There we will **NOT** select *“**Are all items allowed?**”* option.



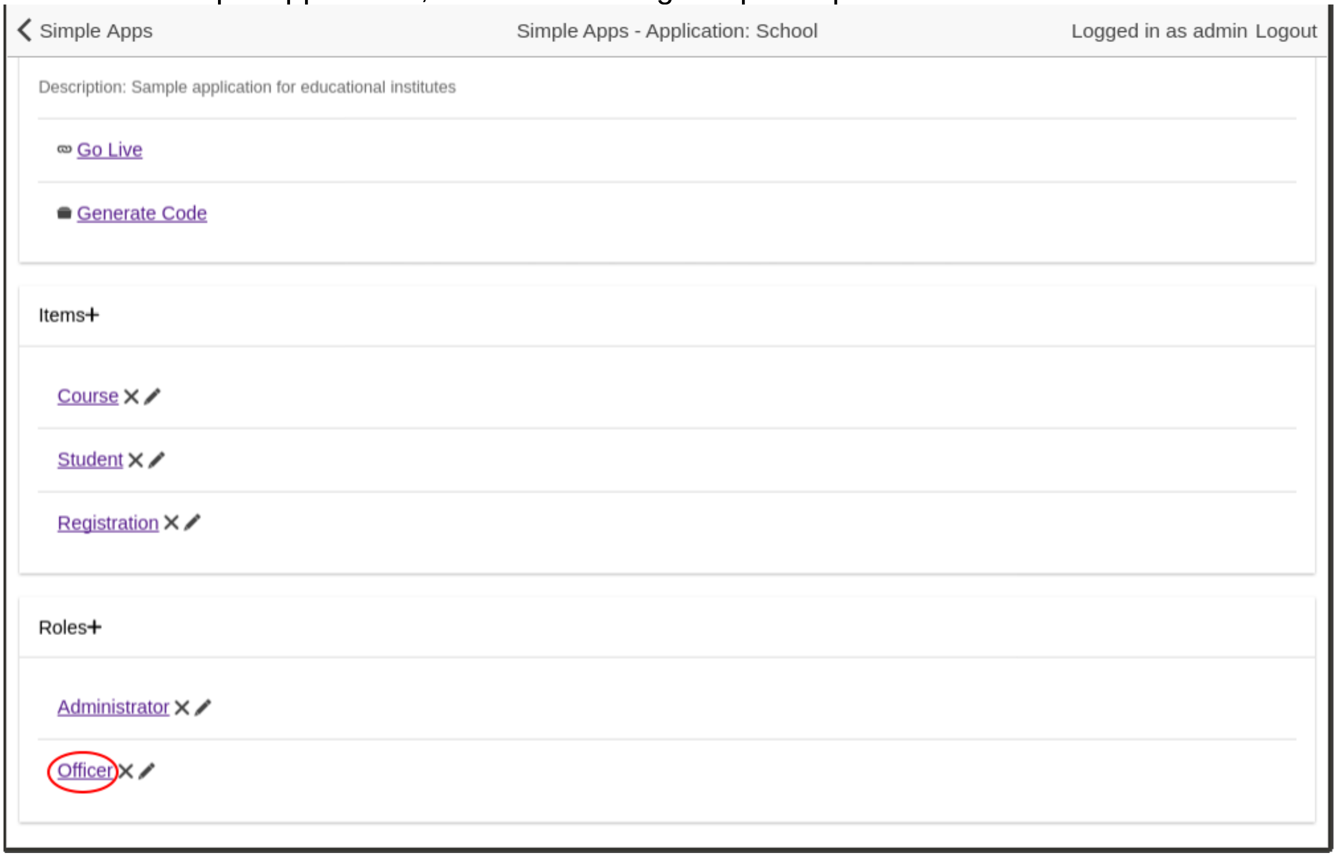
Save Role

Role name Officer

Are all items allowed? □

FIGURE 23 – SAVE ROLE (OFFICER)

4.Click on the role you added, if you need to add special permissions for different items. For sample application, we will need to give special permissions for the **Officer** role.

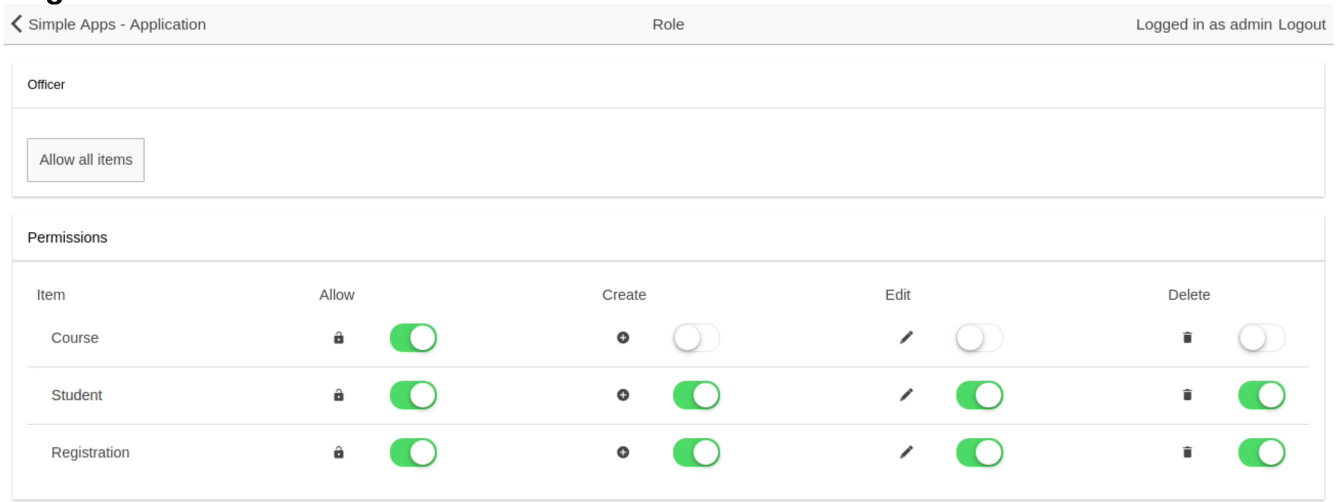


Description: Sample application for educational institutes

FIGURE 24 – CLICK ROLE (OFFICER)

5.Change permissions for each item. Note that the changes will be saved immediately as they are made without the need of saving. If you had selected *“**Are all item allowed?**”* option by accident when creating the **role**, you can change it here by clicking **Give Special Permissions** button.

For **Officer** role, all 3 records **Course**, **Student** and **Registration** items need to be allowed. However, **Create, Edit, Delete** permission must only be given for **Student** and **Registration** items.



Simple Apps - Application | Role | Logged in as admin Logout

Officer

Allow all items

Permissions

Item	Allow	Create	Edit	Delete
Course	●	○	○	○
Student	●	●	●	●
Registration	●	●	●	●

FIGURE 25 – CHANGE PERMISSIONS (OFFICER)

3.Running it live

How to assign a role to a user

1.Click **Menu** and go to **Settings**

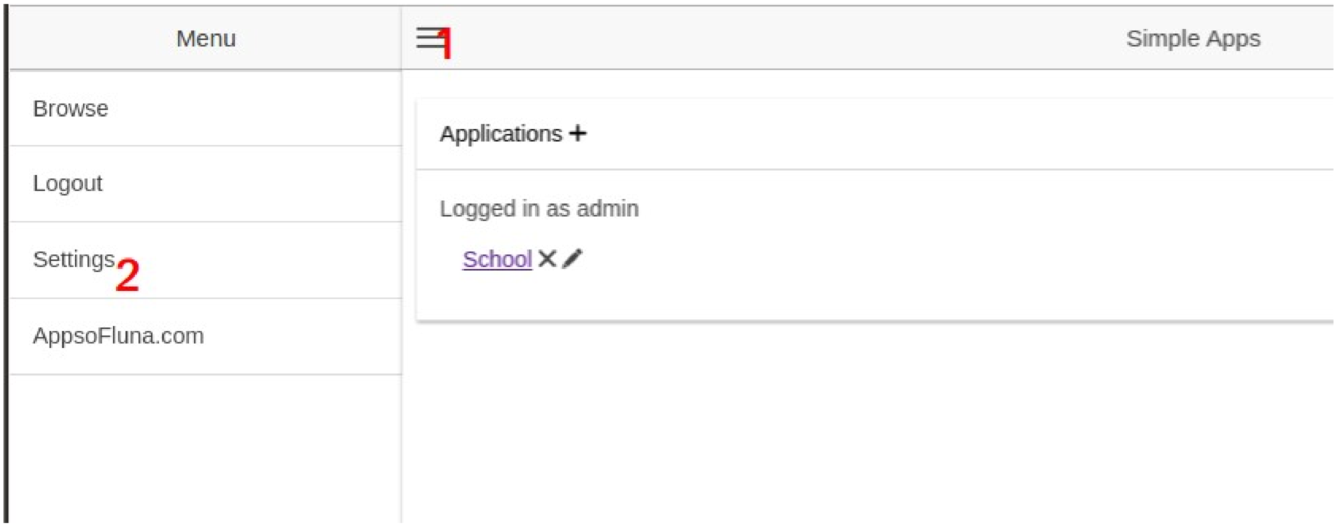


FIGURE 26 – CLICK SETTINGS

2.Click on a user you want to assign a role for.

For the sample application, we will be using the **admin** user itself. If necessary, you may add more users and assign them other roles.

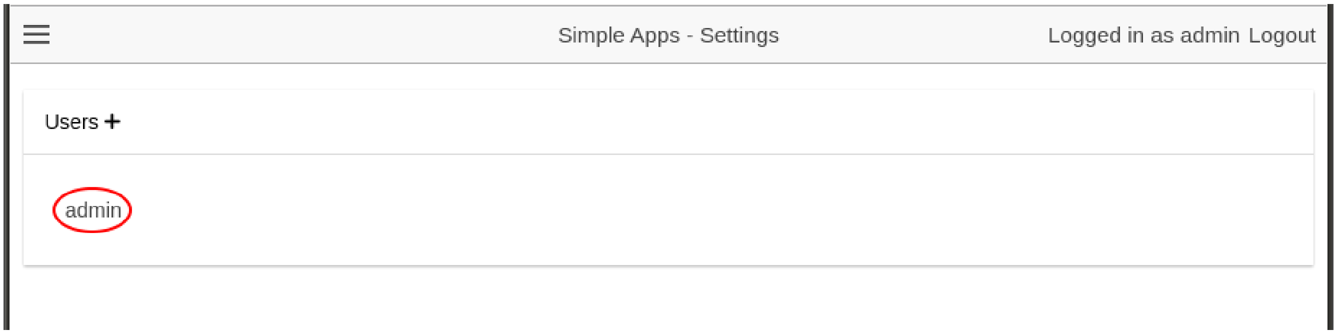
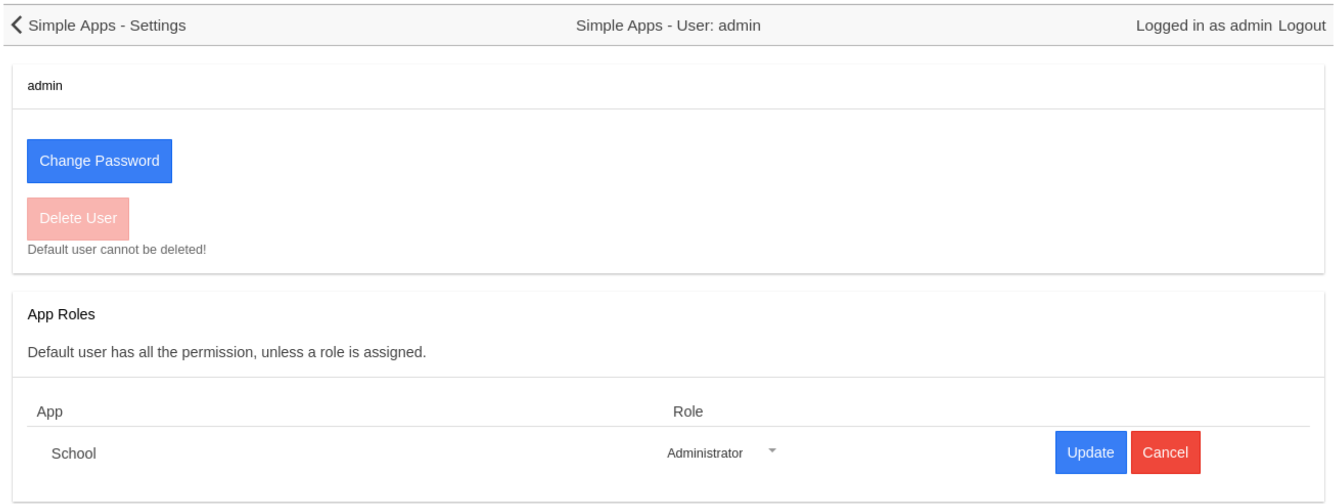


FIGURE 27 – CLICK USER

3.Select a **role** for the relevant app and click **Update**.

For the sample application, we will assign **Administrator** role for the **admin** user. You may add another user and assign **Officer** role, so that when you run it in live mode, the permissions will be applied.



Default user cannot be deleted!

App Roles

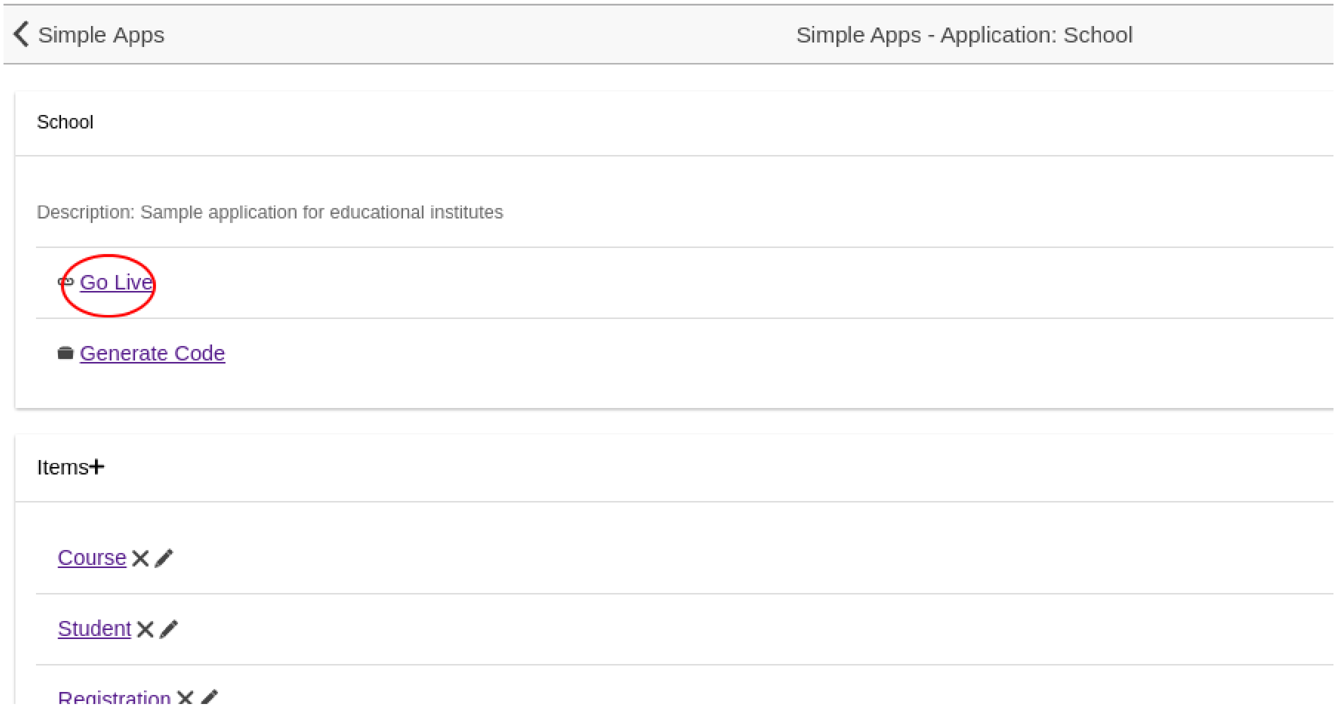
Default user has all the permission, unless a role is assigned.

App	Role
School	Administrator

FIGURE 28 – UPDATE APP USER ROLE

How to run in live mode

1.Go to the application you want to run and Click **Go Live**.



Simple Apps

Simple Apps - Application: School

School

Description: Sample application for educational institutes

Go Live

Generate Code

Items+

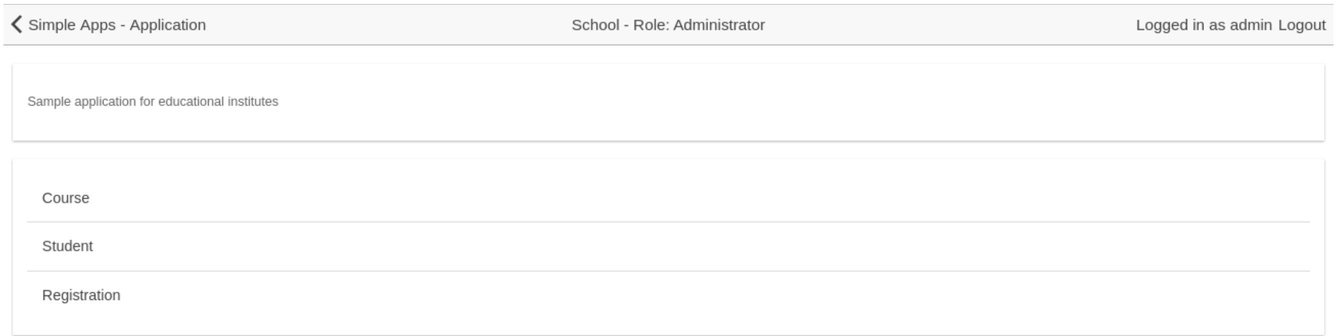
Course ✕

Student ✕

Registration ✕

FIGURE 29 – GO LIVE BUTTON

2.You may go to each item in live mode, if you are permitted.



Simple Apps - Application

School - Role: Administrator

Logged in as admin Logout

Sample application for educational institutes

Course

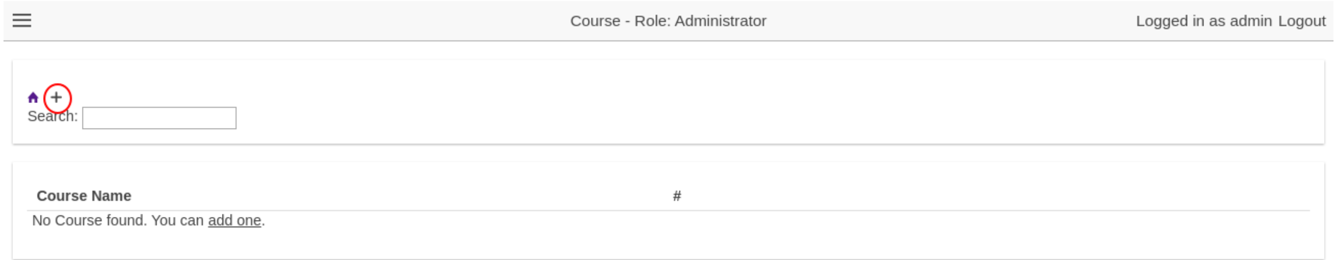
Student

Registration

FIGURE 30 – LIVE MODE HOME

How to add/edit/delete a record in live mode

1.Visit an item in LIVE mode and click **Plus(+)** sign. Note that the plus sign, won't be visible if you don't have the permission.



Course - Role: Administrator

Logged in as admin Logout

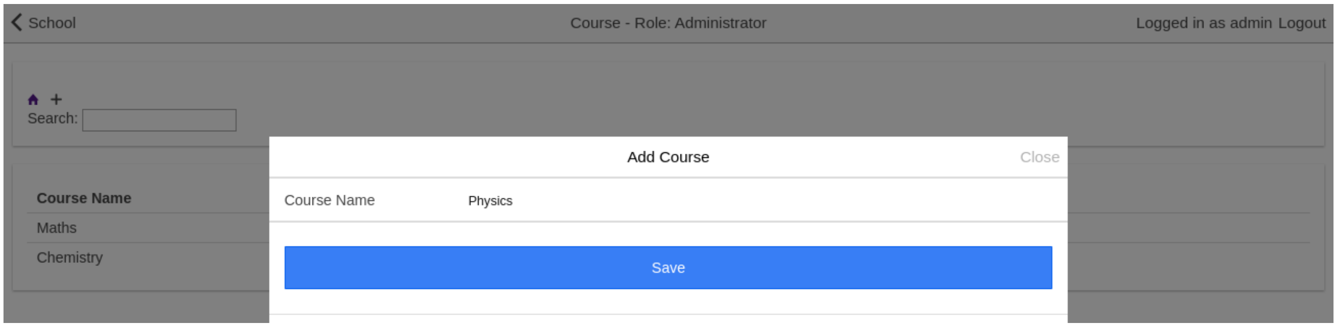
+

Search:

Course Name	#
No Course found. You can add one.	

FIGURE 31 – COURSE LIVE MODE

2.Click enter details and Click **Save**



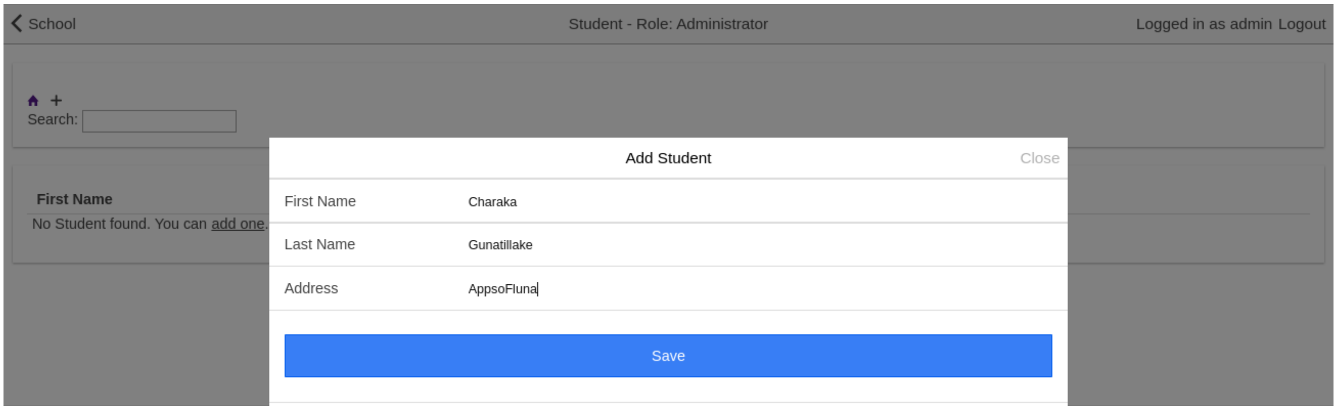
Course - Role: Administrator

Add Course

Course Name: Physics

Course Name
Maths
Chemistry

FIGURE 32 – ADD COURSE



Student - Role: Administrator

Add Student

First Name: Charaka

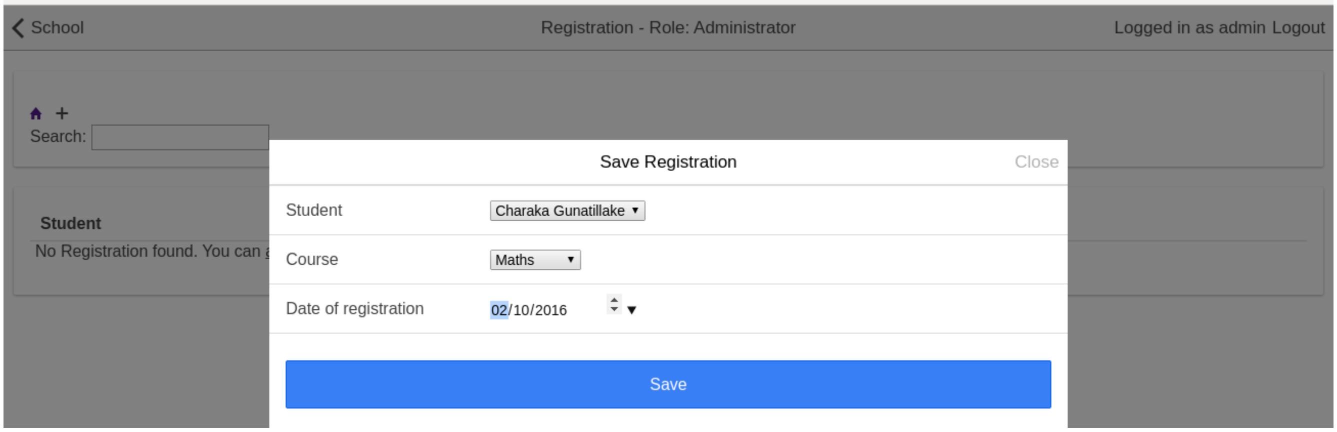
Last Name: Gunatillake

Address: AppsoFluna

First Name

No Student found. You can add one.

FIGURE 33 – ADD STUDENT



Registration - Role: Administrator

Save Registration

Student: Charaka Gunatillake

Course: Maths

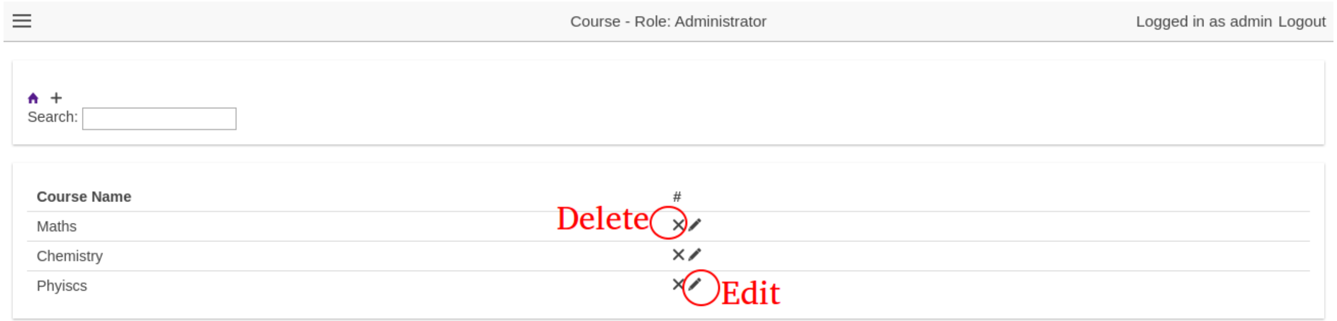
Date of registration: 02/10/2016

Student

No Registration found. You can

FIGURE 34 – ADD REGISTRATION

3.Once you visit back to the item, you may use **edit** and **delete** buttons if you have permissions.



Course Name	#
Maths	✕ ✎
Chemistry	✕ ✎
Phyiscs	✕ ✎

FIGURE 35 – EDIT/DELETE BUTTONS

Note that the records added in LIVE mode won't be included when you generate the source code as it's an standalone application separate from **SimpleApps**.

4.Generating source code (For Developers)

How to generate source code in PHP

1.Go to the application you want to run and Click **Generate Code**.

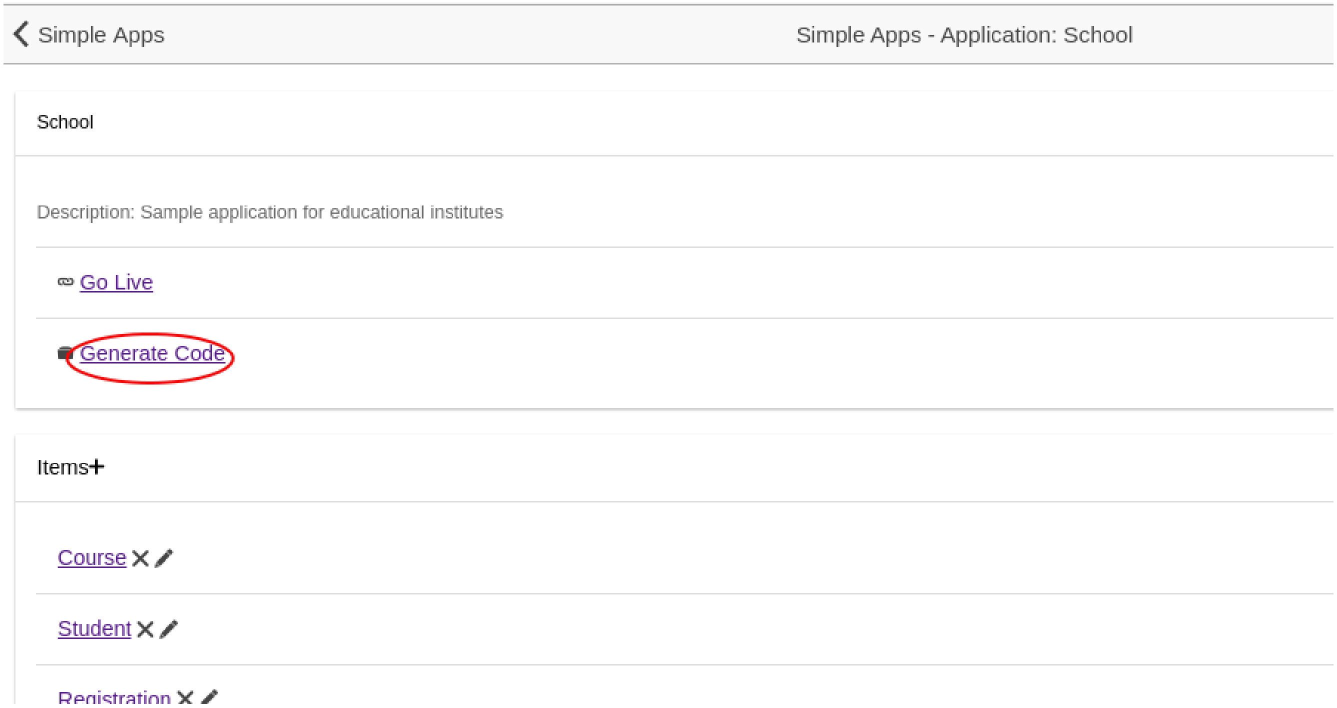
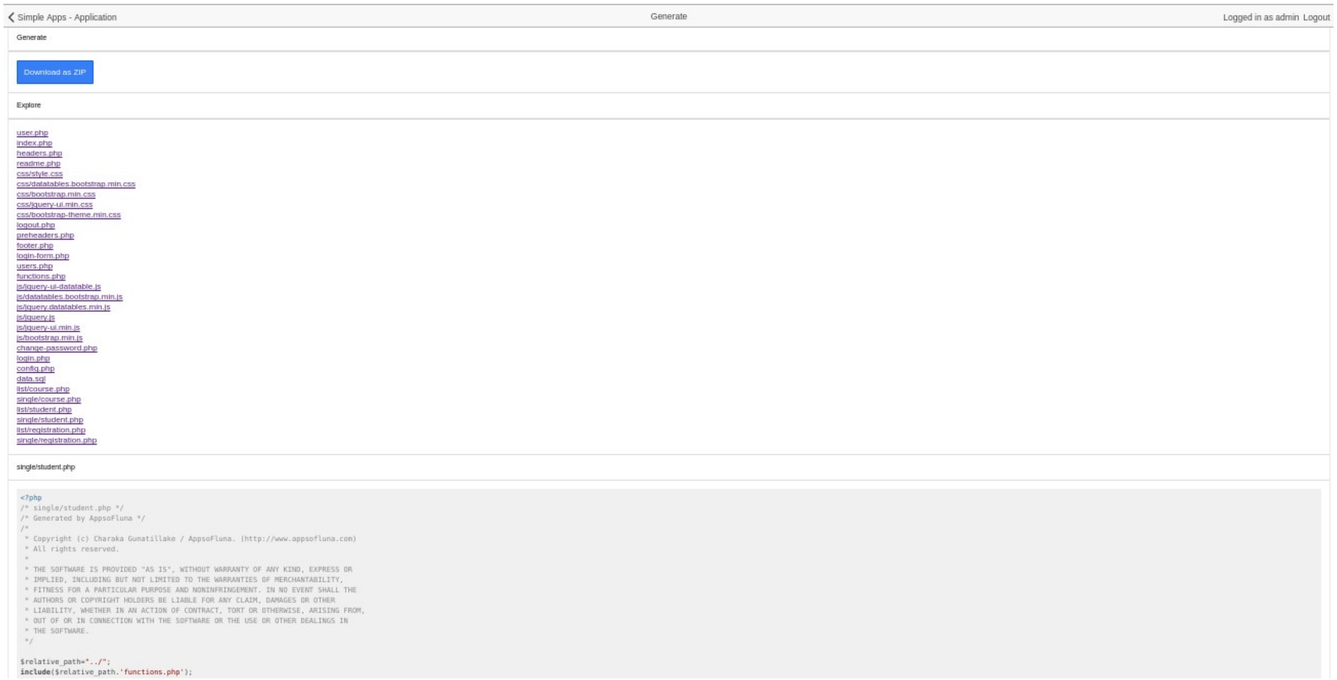

Simple Apps
Simple Apps - Application: School
School
Description: Sample application for educational institutes
Go Live
Generate Code
Items+
Course
Student
Registration

FIGURE 36 – GENERATE CODE BUTTON

You may view the source code of PHP files by clicking on them.



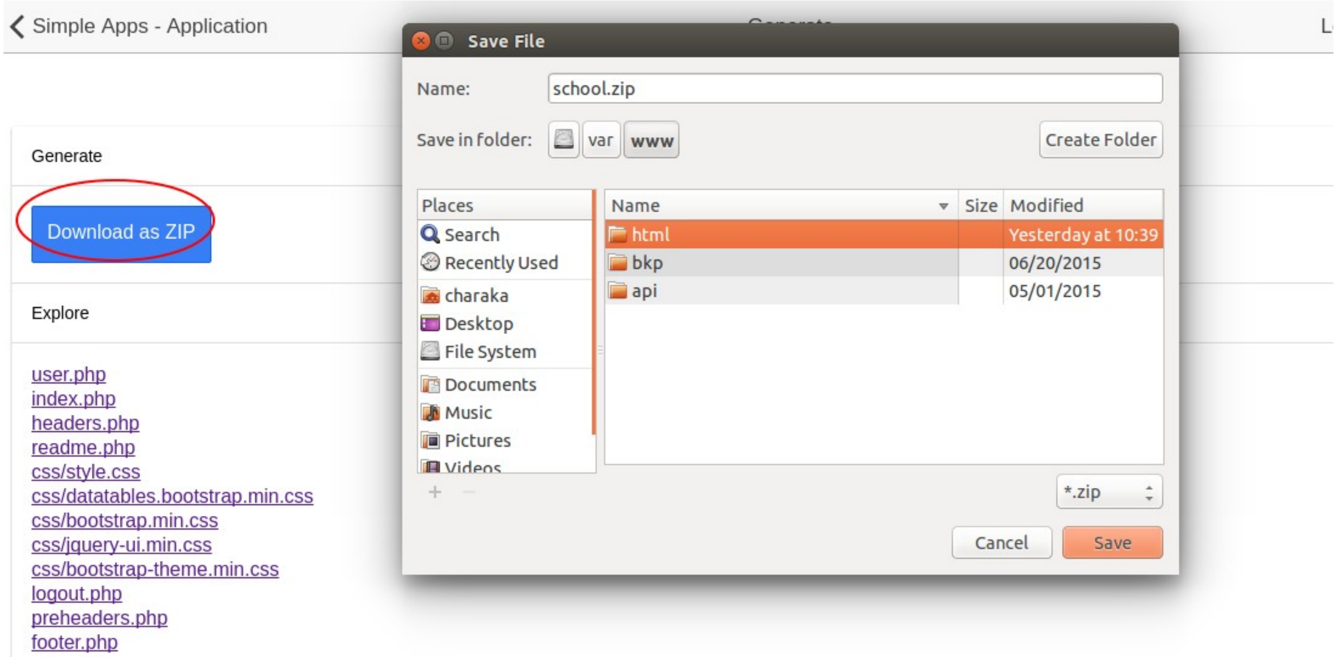
user.php
index.php
headers.php
readme.php
css/style.css
css/datatables.bootstrap.min.css
css/bootstrap.min.css
css/jquery-ui.min.css
css/bootstrap-theme.min.css
logout.php
preheaders.php
footer.php
login-form.php
users.php
functions.php
js/jquery-ui-datatable.js
js/datatables.bootstrap.min.js
js/jquery.datatables.min.js
js/jquery.js
js/jquery-ui.min.js
js/bootstrap.min.js
change-password.php
login.php
config.php
data.sql
list/course.php
single/course.php
list/student.php
single/student.php
list/registration.php
single/registration.php

single/student.php

```
<?php
/* single/student.php */
/* Generated by AppsoFluna */
/*
 * Copyright (c) Charaka Gunatillake / AppsoFluna. (http://www.appsofluna.com)
 * All rights reserved.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
 * THE SOFTWARE.
 */

$relative_path="../";
include($relative_path.'functions.php');
```

FIGURE 37 – VIEW CODE

2.Click **Download as ZIP** button to get the full source code along with images.



Name: school.zip

Save in folder: var www

Name	Size	Modified
html		Yesterday at 10:39
bkp		06/20/2015
api		05/01/2015

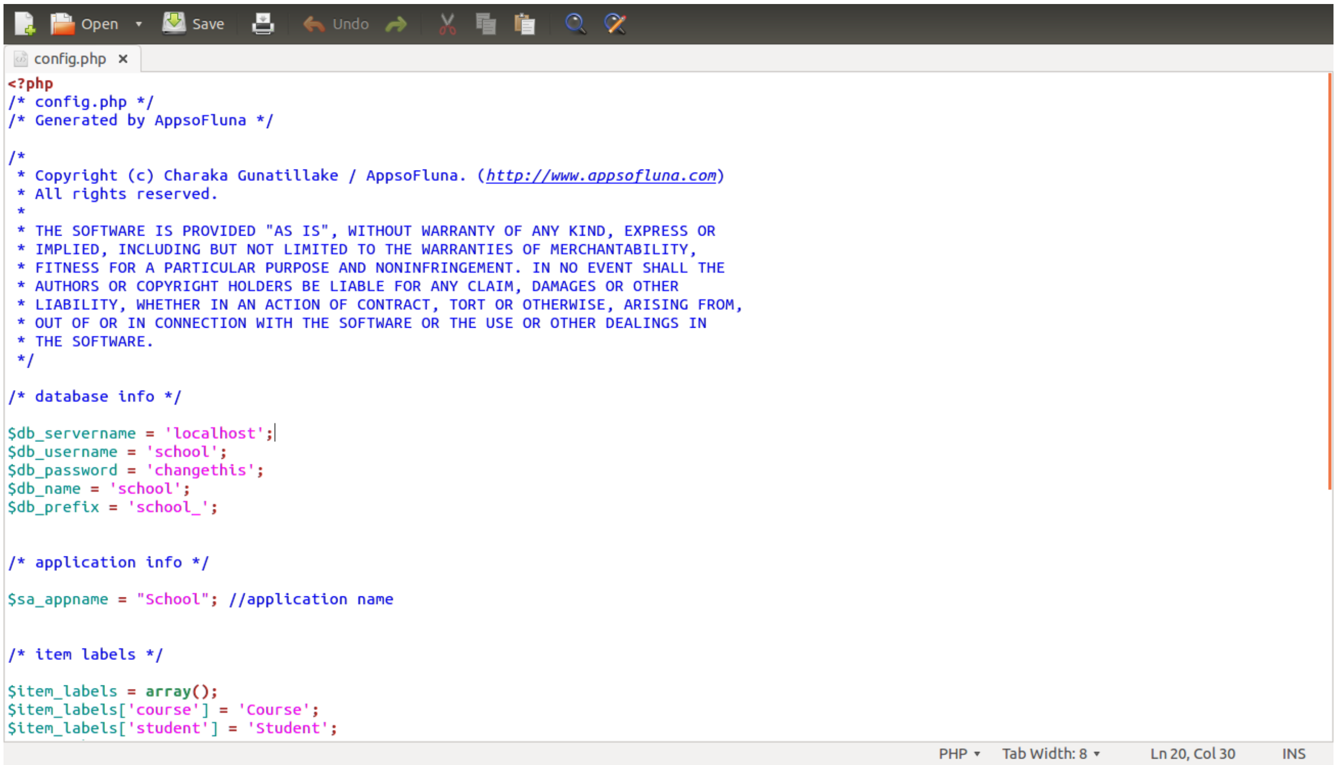
*.zip

user.php
index.php
headers.php
readme.php
css/style.css
css/datatables.bootstrap.min.css
css/bootstrap.min.css
css/jquery-ui.min.css
css/bootstrap-theme.min.css
logout.php
preheaders.php
footer.php

FIGURE 38 – DOWNLOAD BUTTON

3.Extract the zip file into your php server and make them accessible.

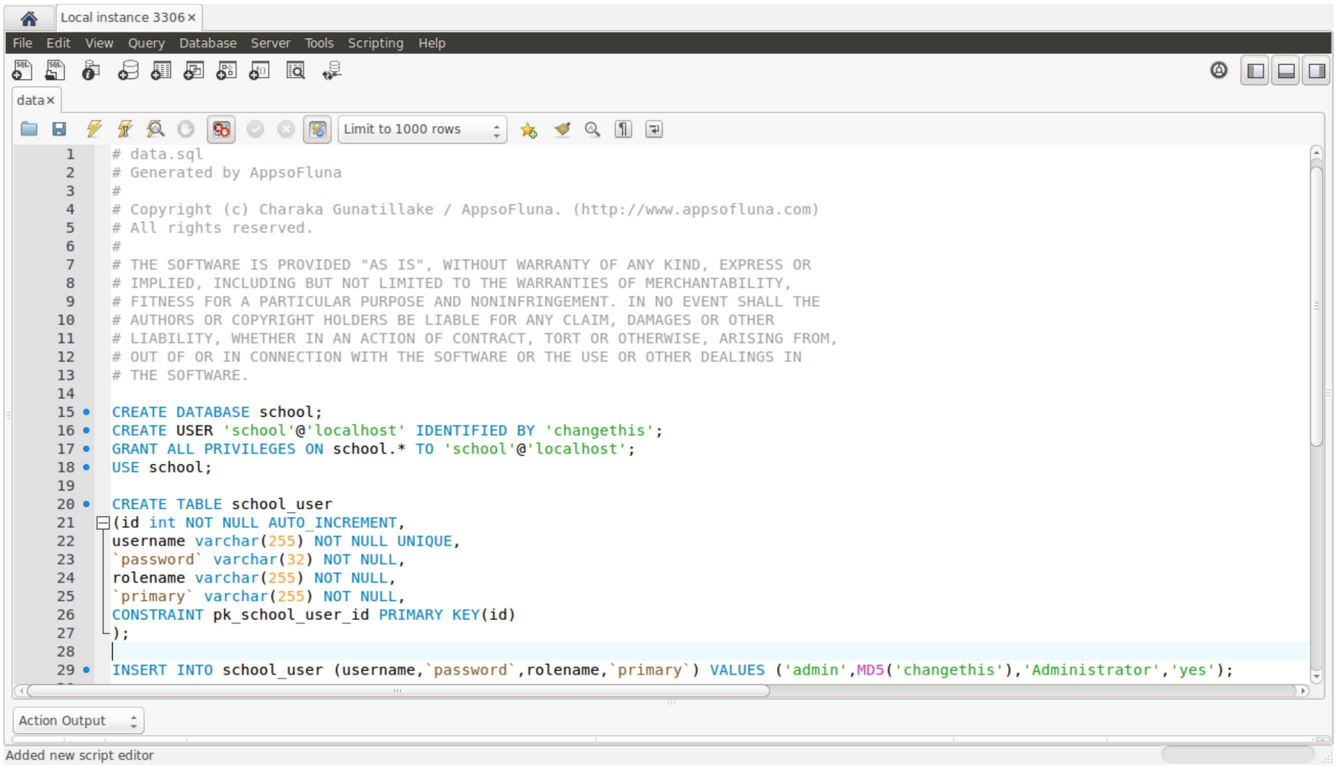
4.Edit **config.php** file to set the database details such as **hostname**, **username** and **password**.



```php
<?php
/* config.php */
/* Generated by AppsoFluna */

/*
 * Copyright (c) Charaka Gunatillake / AppsoFluna. (http://www.appsofluna.com)
 * All rights reserved.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
 * IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
 * FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
 * AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
 * LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
 * OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
 * THE SOFTWARE.
 */

/* database info */

$db_servername = 'localhost';
$db_username = 'school';
$db_password = 'changethis';
$db_name = 'school';
$db_prefix = 'school_';


/* application info */

$sa_appname = "School"; //application name


/* item labels */

$item_labels = array();
$item_labels['course'] = 'Course';
$item_labels['student'] = 'Student';
```

FIGURE 39 – CONFIG FILE

5.Edit data.sql file with database details (tested with MySQL) and apply them in a query browser. It will create tables and the admin user. If a role named **Administrator** has not been added, it will automatically be created and assigned to the user. For the sample application, we have already added one.



```
# data.sql
# Generated by AppsoFluna
#
# Copyright (c) Charaka Gunatillake / AppsoFluna. (http://www.appsofluna.com)
# All rights reserved.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
# FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
# AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
# LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
# OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
# THE SOFTWARE.

CREATE DATABASE school;
CREATE USER 'school'@'localhost' IDENTIFIED BY 'changethis';
GRANT ALL PRIVILEGES ON school.* TO 'school'@'localhost';
USE school;

CREATE TABLE school_user
(id int NOT NULL AUTO_INCREMENT,
username varchar(255) NOT NULL UNIQUE,
`password` varchar(32) NOT NULL,
rolename varchar(255) NOT NULL,
`primary` varchar(255) NOT NULL,
CONSTRAINT pk_school_user_id PRIMARY KEY(id)
);

INSERT INTO school_user (username,`password`,rolename,`primary`) VALUES ('admin',MD5('changethis'),'Administrator','yes');
```

FIGURE 40 – SQL FILE IN QUERY BROWSER

You may customize other files if necessary. For the sample application, we won't change anything.

How to use the PHP web application

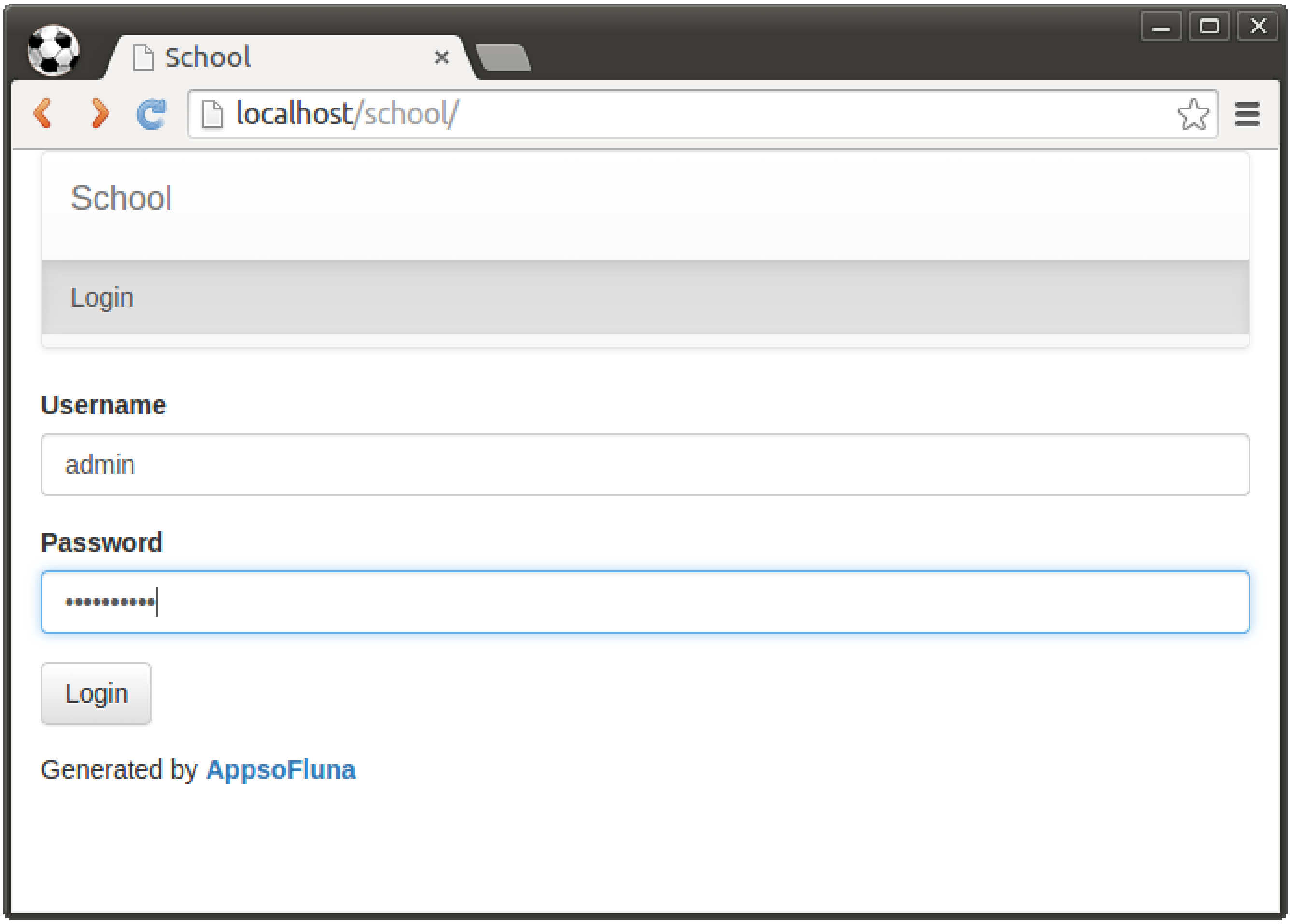
1.Make use your php server and database server running.

2.Visit the application URL

Note that this application does not have any data other than **roles**. The users and records added in **SimpleApps** will not be present in PHP application. It only has one user created automatically by **data.sql** script.

3.Login with default user.

Default username: **admin**
Default password: **changethis**



Username

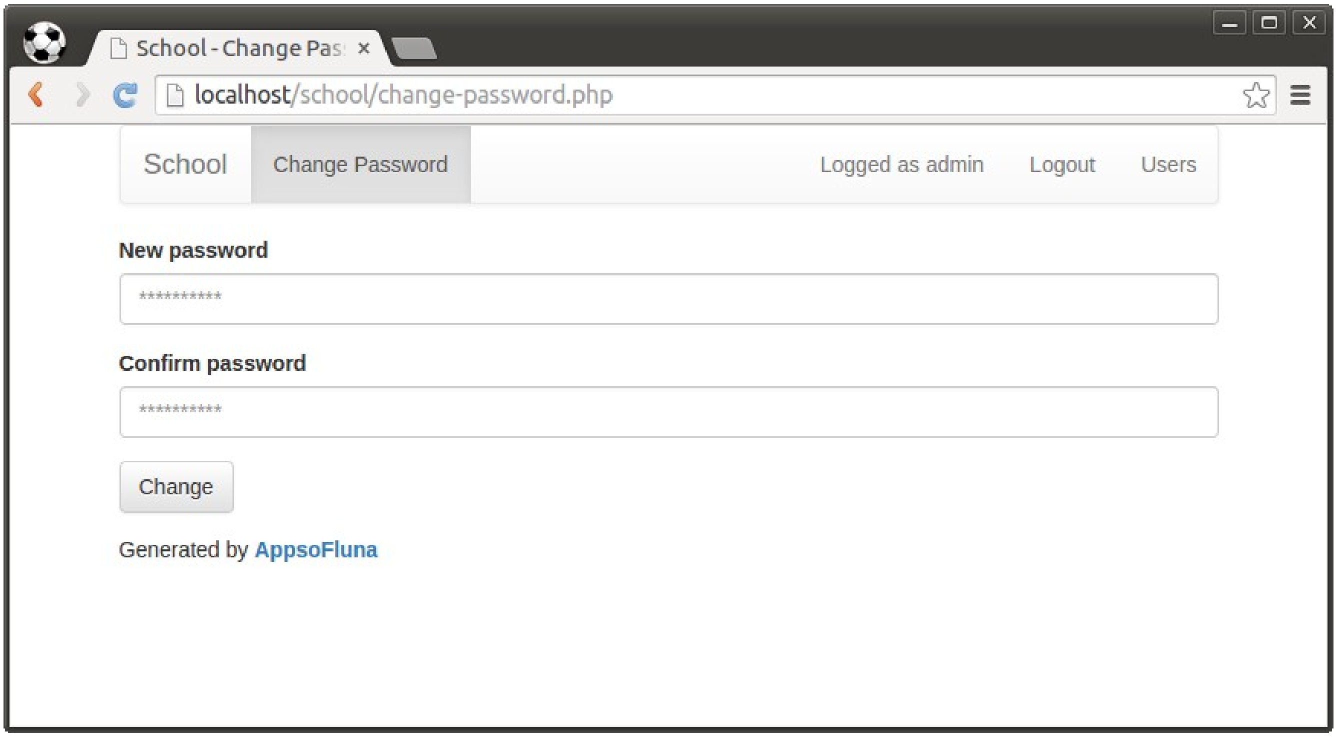
admin

Password

•••••••••••

FIGURE 41 – LOGIN PAGE OF PHP APPLICATION

You may change the password by clicking **Change Password** link.



New password

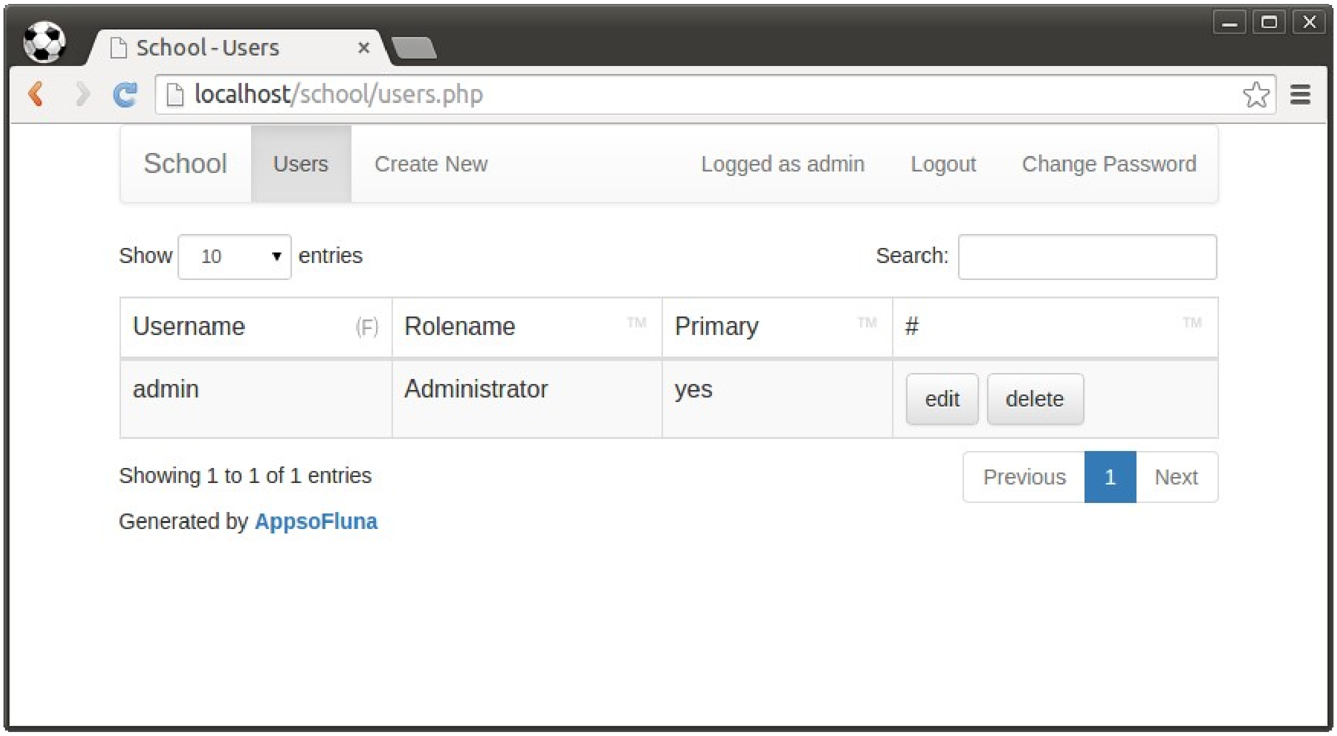
****

Confirm password

****

FIGURE 42 – CHANGE PASSWORD

You may add more users with different roles by clicking **Users**



Show 10 entries Search:

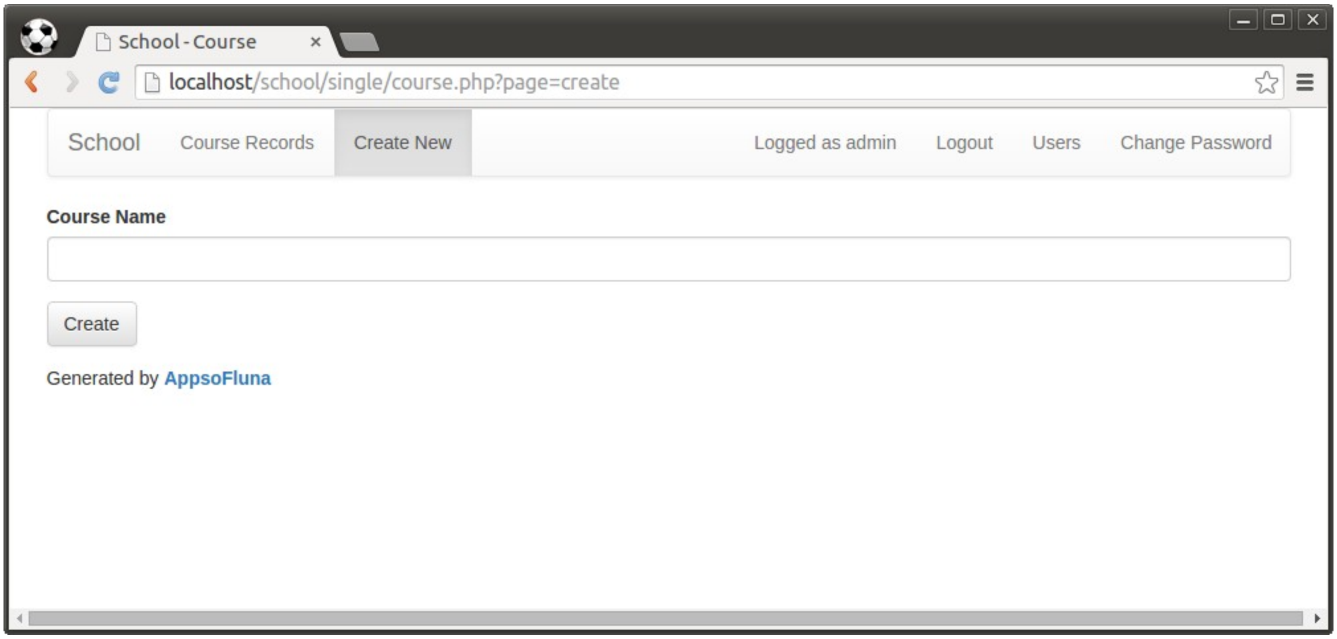
Username	Rolename	Primary	#
admin	Administrator	yes	edit delete

Showing 1 to 1 of 1 entries

Generated by AppsoFluna

FIGURE 43 – USERS

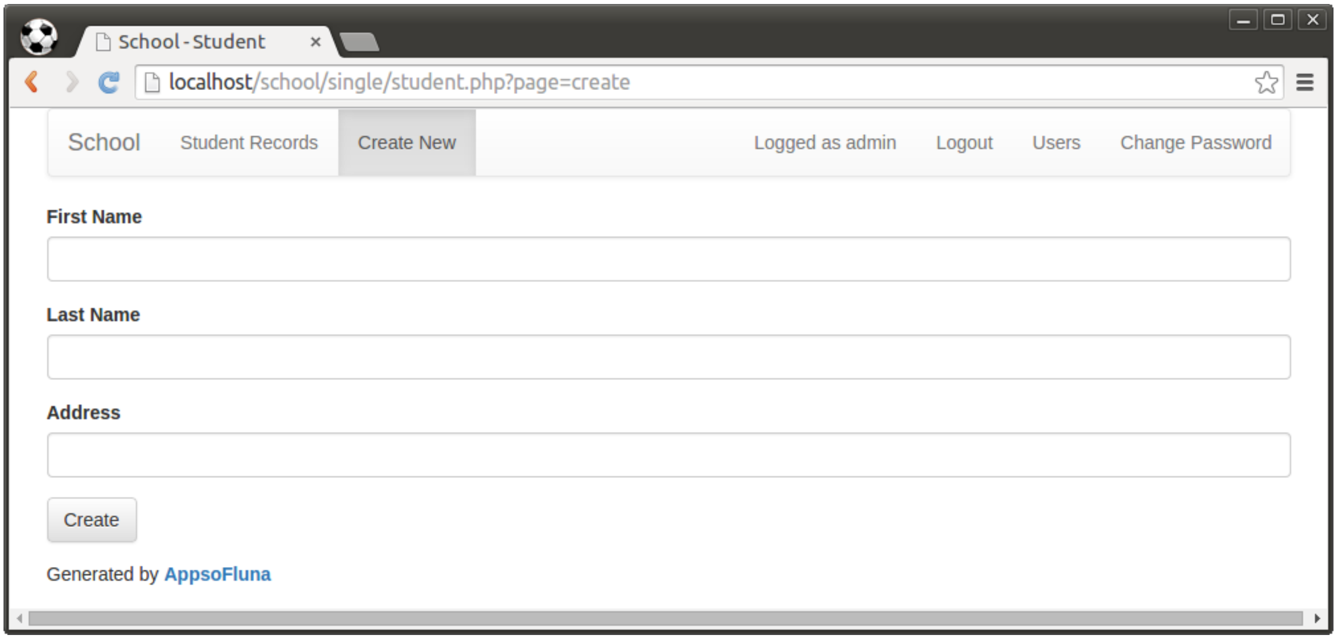
You may browse through different items and **add/edit/delete** records



Course Name

Generated by AppsoFluna

FIGURE 44 – ADD COURSE IN PHP APPLICATION



First Name

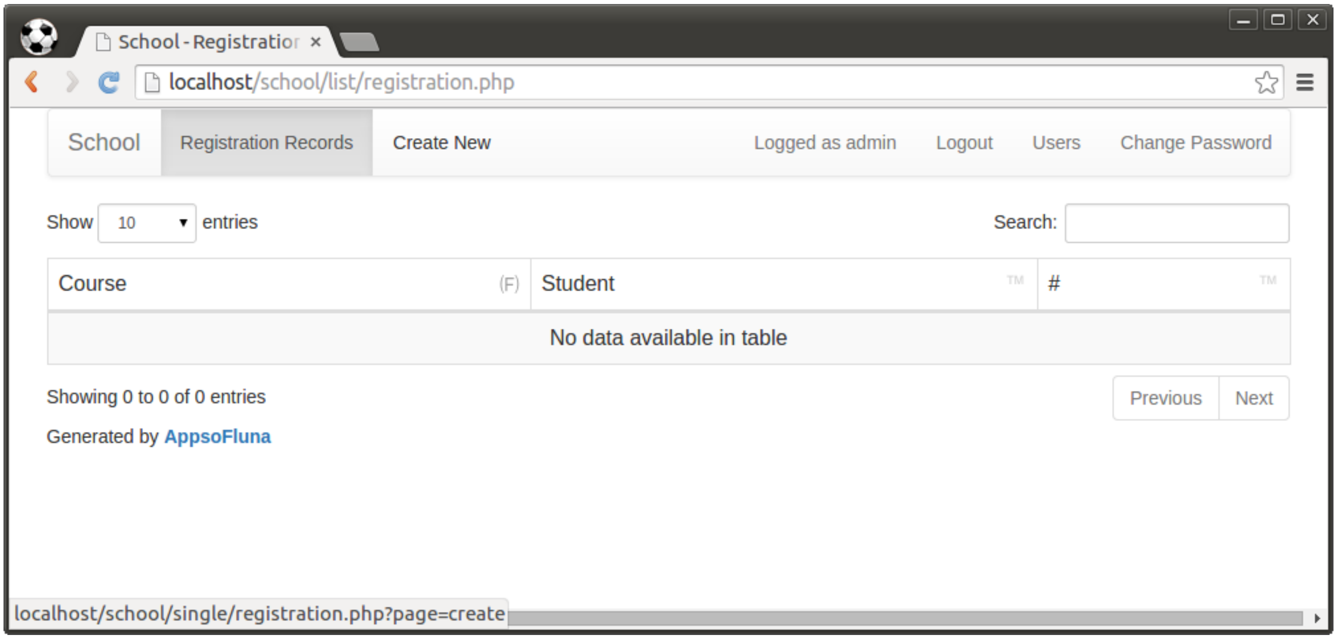
Last Name

Address

Create

Generated by AppsoFluna

FIGURE 45 – ADD STUDENT IN PHP APPLICATION



Show 10 entries Search:

Course	Student	#
No data available in table		

Showing 0 to 0 of 0 entries

Previous Next

Generated by AppsoFluna

localhost/school/single/registration.php?page=create

FIGURE 46 – REGISTRATION RECORDS

Conclusion

In this manual, the core features of the application has been covered. Further knowledge of specific features can be gained by using the application itself. The last section “Generating source code in PHP” is intended for developers. There is much room for customization by editing the php files and css styles. As the code is branched into usable functions, php developers will be able to easily extend files. Let's make apps with ***SimpleApps***!