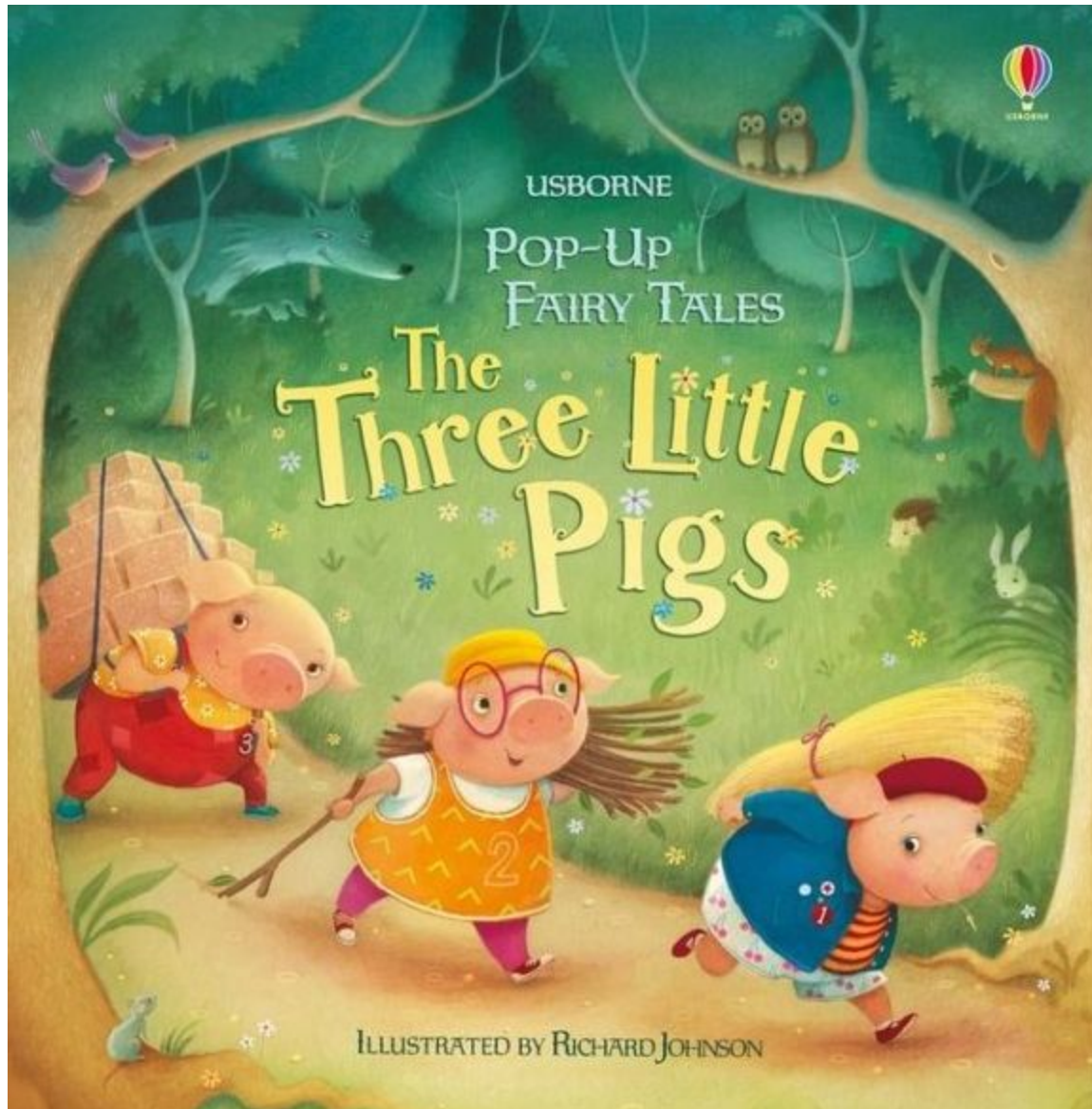


## Inside 4 Manual

1. Overview
2. System Structure
3. Inside 4 Model Structure
4. Installation and Configuration
5. Code
6. Use-cases

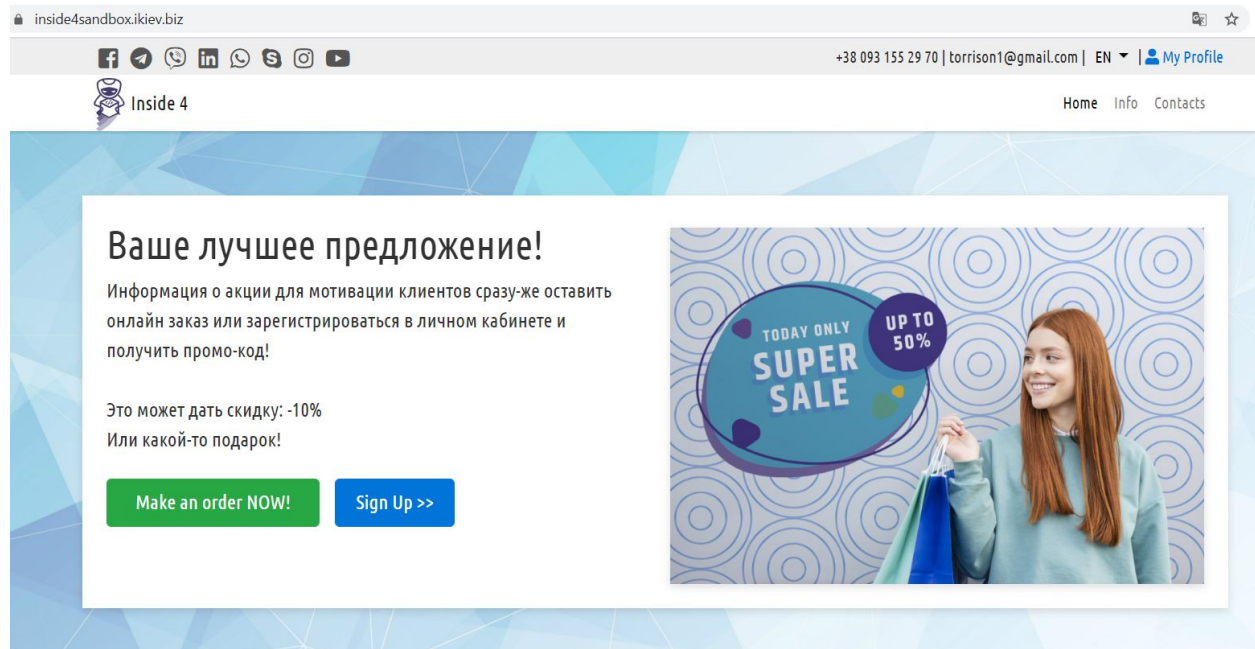


*“Do not forget that fairy tale. We need to build a stone castle!”*

## 1. Overview

Inside 4 is a framework and platform for building web-services, websites and other web programs. It consists of a lot of tools which need every system on the web.

Demo website - <https://inside4sandbox.ikiev.biz/>



Bitbucket Code link - [https://bitbucket.org/digital\\_outsourcing\\_com/inside-4-sandbox/src/master/](https://bitbucket.org/digital_outsourcing_com/inside-4-sandbox/src/master/)

**Concept:** Inside 4 very light and all systems have minimal virtual variables and methods. Pure MVC structure helps to separate views, controllers and models in the different folders in code. Front-End external files also have a separate folder.

Inside 4 is about how to do a product fast, flexible, manageable, secure, light, easy to code and understand what things do and how it works.

## 2. System Structure

System uses PHP7, Web-Server (Apache or NGINX), MySQL database, HTTPS certificates, Google Account, Facebook Account, Git

Inside 4 basis consists of these parts:

**.htaccess** - make a redirect for put URL to variable

**index.php** - entry point

**.gitignore** - for git exceptions

**robots.txt** - for SEO

**AppControllers** - entry classes which run by URL, for example: site.com/**controller/method/variable**

**AppControllers/Auth** - Controller for user login and authentication system interfaces

**AppControllers/Auth\_API** - user login and authentication system APIs for related systems, mobile apps and AJAX requests

**AppControllers/Content** - Blog and Pages System like any website must have

**AppControllers/CRM\_API** - APIs for CRM calls like lead tracking, save requests and orders, etc.

**AppControllers/Debug** - special controller for test tools, classes and any part of the system

**AppControllers/Inside** - an interface for included Admin Panel System

**AppControllers/Inside\_API** - API for included Admin Panel System

**AppControllers/Main** - Main Controller for Main Page and any other important pages of a website

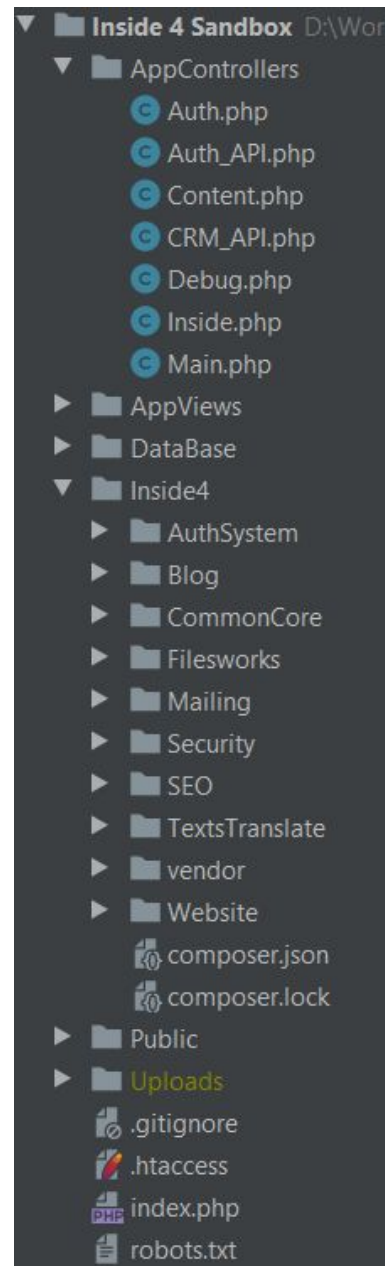
**AppViews** - folder for front-end code templates with html, css, javascript, etc.

**DataBase** - folder for Database structure dumps

**Inside4 (Model)** - folder with a programs, models and all modules of a system

**Public** - folder for front-end static files, scripts and libraries, modules, widgets.

**Uploads** - folder for files which are uploaded inside a system by user or any other process.



### 3. Inside 4 Model Structure

The main part of a system is an Inside 4 Model Folder. It is hierarchical folders and files structure of all parts of a system.

The Core and Commons is in CommonCore folder and consists of:

**Globals** - class about global variable GLOBALS in PHP which uses for store data which can be necessary in any part of a system.

**Routing** - class for work with URL, define prefixes, route and run special controller and method with parameters.

**BaseController** - base class which is extended by system controllers and consists of core classes and variables.

**Commons** - library class with useful functions in any part of the system.

**Website** - class for configuring website parameters and all advanced stuff for website systems, analytics, redirects, etc.

**Mailing/Mailer** - everything about emailing, SMTP, POP3, Gmail API, PHPMailer, etc.

**Database** - special class for work with Database easier.

**Security** - tools and solutions to make your system secure and stable.

**Sessions** - is about client identification, tracking activities and define users behaviour

**RenderView** - helps wear data to views php templates and render html responses.

**TextsTranslate** - class for languages translate system for localisation

**AuthSystem** - is a big and serious part of any system. It is about users, users\_sessions, users\_activities, users\_groups, login, register, profile, google login, etc.

**Response** - special class for easy to make any kind of responses for different situations, systems and clients. As the default system works with HTML and JSON for API.

**SEO** - class for search optimization tools

**Blog** - pack of tools for building classic content systems with lists, categories, tags, comments, articles mechanics, landing pages, etc.

**CRM** - class for with customers activities, clients requests, orders and API calls to other CRM, Analytics and Marketing systems.

**Filesworks** - is about files, upload, download, secure from scripts, resize and edit images, etc.

**InsideAdminSystem** - everything for the inside part of web-service. Admin System, Form Generation, CRUD APIs, Auto Tables for Database Management and a lot useful tools

**Vendor** - is a folder for **composer** to install any other libs for making anything you want.

Sometimes developers put business logic to the Controller but it is not good, controllers are only about request data, put data to a special model and get an output, wear by view and send in response. Build product Inside, and take care of product code structure.

## 4. Installation and Configuration

Clone a basis code from Git repository:

[https://bitbucket.org/digital\\_outsourcing\\_com/inside-4-sandbox/src/master/](https://bitbucket.org/digital_outsourcing_com/inside-4-sandbox/src/master/)

Put code inside the root website folder and check https, mysql, apache, php7.

Get backup from Database folder and import to your new database for the project

Set in **\Inside4\CommonCore\Database.php** Database access data

Set in **\Inside4\Mailing\Mailer.php** Gmail API data

Set in **\Inside4\Security\Security.php** unique keys and salts for encryption system

Set in **\Inside4\SEO\SEO.php** headers for SEO titles, description, etc. Or check it was connected to dynamic variables from Translation System

Set in **\Inside4\TextsTranslate\TextsTranslate.php** default language of system

And finally set all parameters in **\Inside4\Website\Website.php** for all website systems

And if everything is ok - try an URL of the website and all must work!

## 5. Code

System has a very easy structure. Code has been written on PHP7 OOP with namespaces. Code style similar to PSR rules and classic web-development. Javascript code located in the views folder in the \_footer suffix files and in static folder Public for Front-End modules, Widgets and Libraries.

The main point of coding is about code optimisation for creating less elements for solutions, methods and any tools of a system.

Typical Controllers consist of preparing input variables data, make easy preparation operations and step by step put data to special methods by defined process and finally put a result to specific response or render html view for a website page.

Controllers extend BaseController with a lot of tools for easy work with data. By default loaded auth system, session, database, mailer and a lot of useful objects which need for easy development of data processing.

Also you can use any objects from Inside 4 Model folder for any operations and put there a controller object for giving these classes full core and common system functionality.

### CRM API Code Example:

```
<?php
namespace App\Controllers;
use Inside4\CommonCore\BaseController as BaseController;

Class CRM_API extends BaseController
{
    public function add_contacts_request() {

        $data['requests_user_contacts'] = $this->security->xss_cleaner($_POST['name_contacts']);
        $data['requests_message'] = $this->security->xss_cleaner($_POST['message']);
        $data['requests_type'] = 1;
        $data['requests_result'] = 1;
        $data['requests_url'] = $this->security->xss_cleaner($_POST['url']);
        $data['requests_datetime'] = time();
        if (isset($this->auth->user['id'])) $user_id = $this->auth->user['id'];
        else $user_id = 0;
        $data['requests_user_id'] = $user_id;

        // Usage of CRM Model
        $crm = new \Inside4\CRM\CRM();
        $crm->init_from_controller($this);

        $crm->website_request($data);

        $this->sessions->track_activity('contacts request from user #'.$user_id);

        $result['status'] = "success";
        $this->response->echo_json($result);
    }
}
```

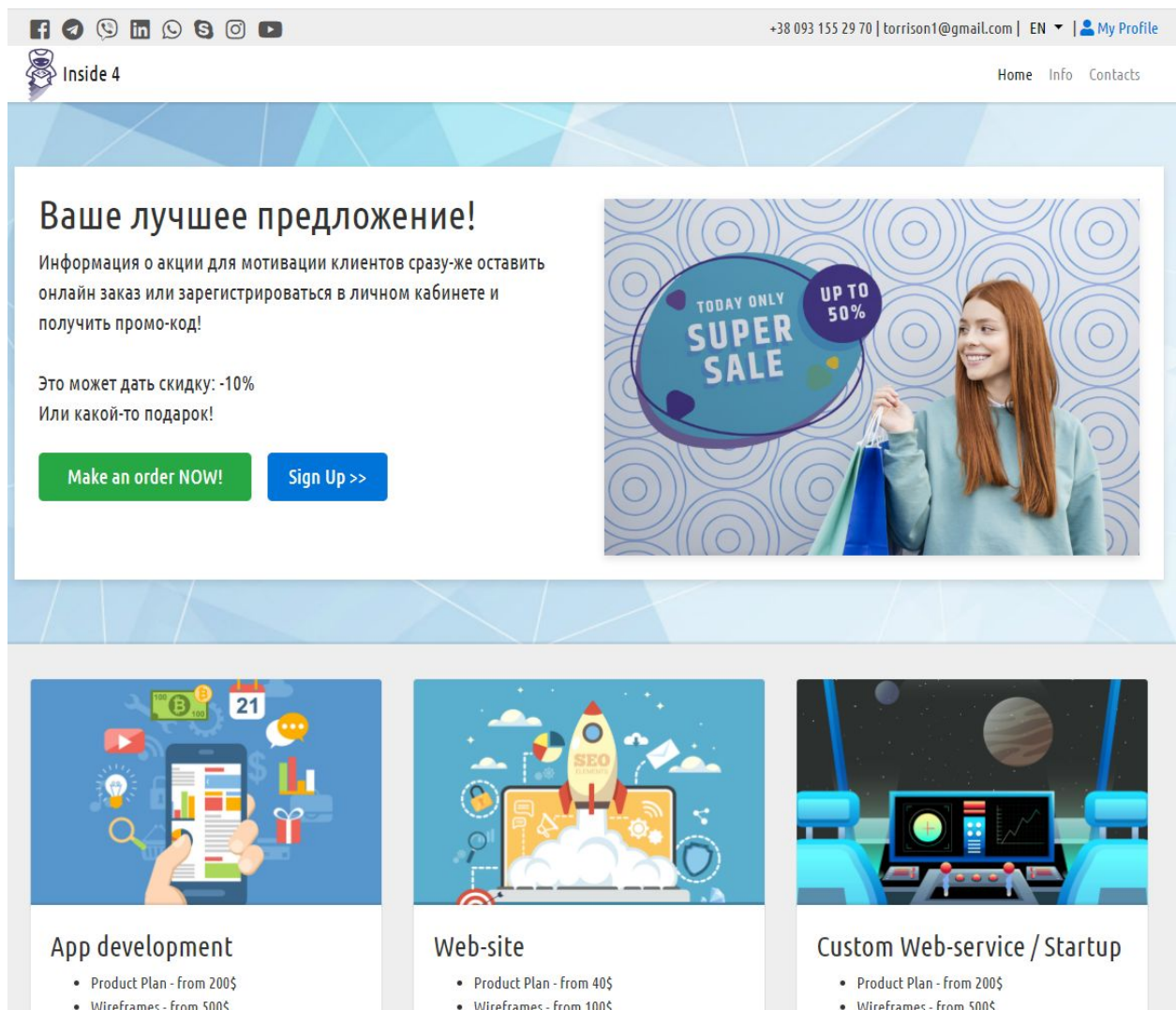


## 6. Use-cases

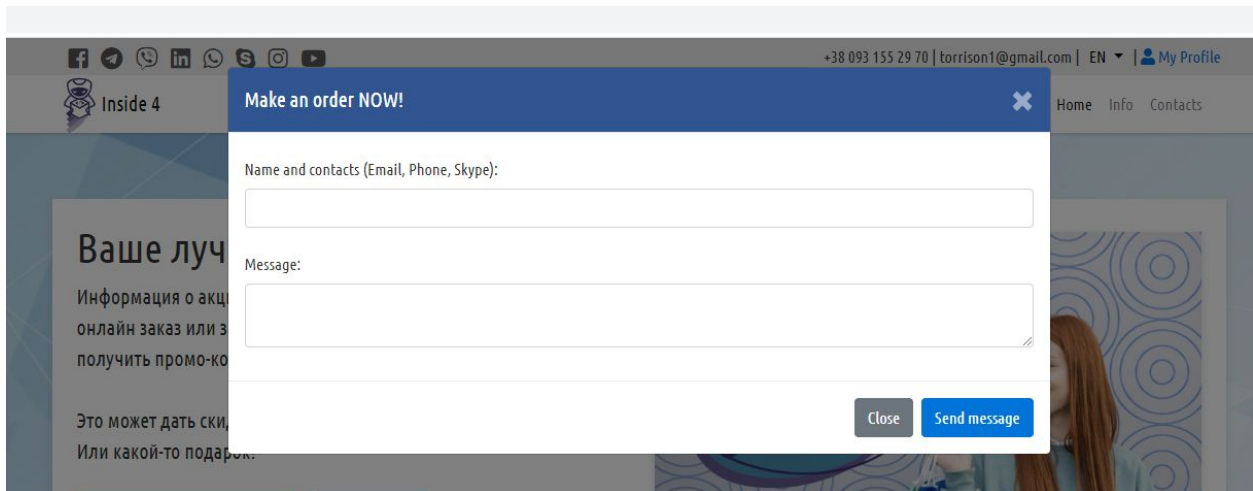
This part about web-service usage and common user-cases.

### [UC1] First usage of service with User Registration:

1) User open URL first. System makes a session, and the user sees a Main Page. Page has SEO title and Description, Language switch, Lead block in the top, other blocks lower and a top-menu with top subheader with link to user cabinet login.

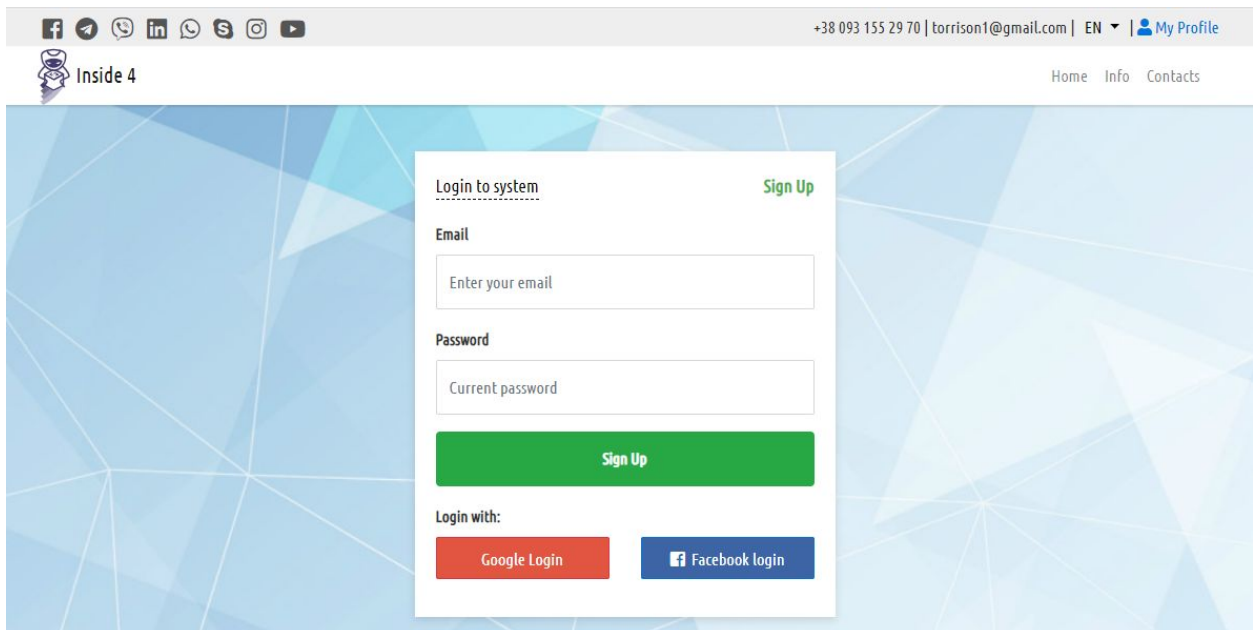


2) He can send a request by request pop-up. Request info saves in the database and sends to the admin's email with all information of request.



The screenshot shows a website interface with a dark header. The header includes social media icons (Facebook, Telegram, WhatsApp, LinkedIn, Messenger, Skype, Instagram, YouTube) on the left, and contact information (+38 093 155 29 70 | torrison1@gmail.com | EN | My Profile) on the right. A navigation bar below the header contains 'Home', 'Info', and 'Contacts'. A large blue pop-up form titled 'Make an order NOW!' is centered on the screen. The form has a close button (X) in the top right corner. It contains two text input fields: 'Name and contacts (Email, Phone, Skype):' and 'Message:'. At the bottom right of the form are two buttons: 'Close' and 'Send message'.

3) Also users go to register form if they click another button. Here users can register, login by Google or Facebook, login via email or recover forgotten password.



The screenshot shows a website interface with a light blue geometric background. The header is identical to the previous screenshot. A white form titled 'Login to system' is centered on the screen. The form has a 'Sign Up' link in green text at the top right. It contains two text input fields: 'Email' with the placeholder 'Enter your email' and 'Password' with the placeholder 'Current password'. Below these fields is a large green button labeled 'Sign Up'. At the bottom of the form, under the heading 'Login with:', there are two buttons: 'Google Login' (red) and 'Facebook login' (blue).



+38 093 155 29 70 | torrison1@gmail.com | EN ▾ | [My Profile](#)

Inside 4

Home Info Contacts

---

[11] Alex Torrison (torrison1@gmail.com)

[Inside Admin Panel >>](#)
[Change password](#)

[Logout from profile](#)

Your Promocode: ZDK234865

Level 1: -20% Discount for all services

Next Level: Buy more 30 Services and Get -30%

Personal data

NickName

Phone  
[Verify phone](#)

Email  
[verified](#)

Language

Изображение  
☐ Отметка на удаление

[Save changes](#)

1) Users can see contacts page and send there contact request:

2) Users can see a blog list, use tags and categories and make comments...