Aschalew Wolde Gebremichael

Integration of REST Web Service with Social Media API

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Media Engineering

Thesis

26.9.2016

Author	Gebremichael Aschalew Wolde
Title	Integration of REST Web Service with Social Media API
Number of Pages	27 pages + 1 appendix
Date	26.9.2016
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Specialisation option	Application Development
Instructor	Kari Aaltonen, Senior Lecturer

The increasing number of social media users has led to a scenario in which a successful online strategy includes effective integration with the social media platforms, such as Facebook and Twitter. Understanding the fundamentals of REST API architecture together with factors that play a key role in building robust web service will create a strong foundation in the integration of a web application with a Social Media API.

As a result, this thesis provides discussion on the fundamental theory and technical analysis of REST architecture and RESTful web services. Following these topics, the RESTful nature of Twitter and Facebook APIs will be discussed.

As part of the thesis, a practical project has been carried out. The project includes building a robust web service using the latest web technologies. Building the web application has created an opportunity to understand and analyze the concepts and theories presented in the theoretical section of the thesis.

Keywords	REST, API, Web Service, RESTful

Contents

Appendix

List of Abbreviations

REST	Representational State Transfer
API	Application Programming Interface
HTTP	Hyper Text Transfer Protocol
URI	Unique Resource Identifier
JSON	JavaScript Object Notation
XML	Extensible Mark-up Language
IP	Internet Protocol
DNS	Domain Name System
HTML	Hyper Text Mark-up Language
SSL	Secure Socket Layer
TSL	Transport Layer Security
UTF	Unicode Transformation Format
AJAX	Asynchronous JavaScript and XML
RSS	Really Simple Syndication
SDK	Software Development KIT

PHP	Personal Home Page
CSS	Cascading Style Sheets
OOP	Object Oriented Programming
FTP	File Transfer Protocol
MVC	Model View Controller
SQL	Structured Query Language

1 Introduction

The purpose of this thesis is to understand the REST (Representational State Transfer) architecture and social media API (Application Programming Interface). Moreover, this thesis covers steps required to integrate third party web applications with social media using REST API.

As the number of people using social media such as Facebook and Twitter increase exponentially, integrating web applications with social media becomes a necessary step. Applications can be created to interact with the huge user base of social Medias in various ways. These days it is difficult to find a small or large-scale business or brands that do not have presence in social media. As a result, understanding the theoretical concept and technical implementation behind these technologies is crucial.

For the purpose of understanding and implementing the technical aspect of the topic, I have designed and built a robust web platform which implements the theoretical and technical concepts discussed in the theory section of the thesis.

2 REST Architecture

According Roy Thomas Fielding, REST (Representational State Transfer) is defined as an architectural style and design of network based software architecture that helps to create and organize distributed software systems. This chapter discusses the main building block and technical implementation of REST architecture citing the work of Dr. Roy Fielding. In order to define REST architecture it is good to start with a blank slut and add each constraint one after the other. [1]

2.1 REST Constraints

REST constraints are design rules and standards that are implemented to create the REST architecture style. Each constraint in REST architecture has both advantages and disadvantages. Making the decision whether to implement REST or not needs a clear understanding of the trade off in each constraint. [1]

Client- Server

The first constraint to be added is the Client-Server. A server is responsible for handling a set of services that include, listening for a request from the client and sending the appropriate response back to the client. A client on the other hand sends request over a network to get service from the server. The main point in client-server constraint is the separation of front-end code from server side code because this allows for independent improvement of each side without affecting the other. [1]

Stateless

The second constraint implemented after the client-server is stateless constraint. The stateless nature of REST requires each communication between the client and the server to include all the information required by the server to understand the request without using any prior stored data. [1]

Catching

The response from the server can be labeled as cacheable or non-cacheable. This way the client can catch the response for future reuse. Implementing catch in REST architecture style has the advantage in improving efficiency, scalability and performance for the end user by partially or fully eliminating some interaction with the server. [1]

Uniform interface

Uniform interface constraint is one of the major characteristics of REST architecture and makes it preferable compared to other similar alternatives. Providing a uniform interface will not only make it easier for users to interact with the system by following predefined set of rules but also gives the freedom for different clients to interact with the server seamlessly. [1]

Layered Systems

The main reason behind the invention of REST is to achieve the best service on the web. Applications running on the web are expected to handle a huge load of traffic. Having a layered system will separate different component into layers. In this system a layer only takes input from the one below it and gives output to the one above it. As a result of this, the overall complexity of the system will be reduced. It will also be easy to observe component coupling in the system. [1]

Code on demand

Code on demand is the only optional characteristic from all the REST constraints. Software architects can make their own decision whether to implement it or not by weighing on its advantages and disadvantages. With this constraint a client can download code that is provided by the server and execute it. [1]

2.2 REST architectural elements

REST architecture can be defined as abstraction architectural elements which are inside distributed hypermedia elements. There are there different architectural elements in REST architecture, namely, data elements, connectors and components. [1]

Connectors

There are different kinds of connectors used in REST architecture: client, server, catch, resolver and tunnel. Connectors provide means for communicating, accessing and manipulating a resource between different components. Connectors provide an abstract interface for components. Providing abstraction interface (hiding the inner working mechanism within each connector) has two main advantages. First is simplicity by imposing separation of concern between connectors. The second advantage is the ability to provide substitutability by allowing the underlying mechanism to be changed without affecting the user. [1]

Components

Components are the last structural element in REST architecture and they are categorized according to the role they play in the overall application process. There are four different components: origin server, gateway, proxy and user agent. The user agent uses the client connector to initiate a request and last recipient of the response. The origin server uses the server connector to manage the namespaces for requested resource. Since origin server is the only and main source of the resource, it is the final destination for of any request that intends to change the resource. Proxy and gateway are called intermediate components. They can act both as client and server. A proxy component is an intermediate component used by the client for the purpose of interface encapsulation for other services, data translation, performance enhancement, or security protection. Whereas a gateway is an intermediate component imposed by a network or origin server to provide other services like data translation, performance enhancement or security protection. [1]

Data elements

The different components in REST communicate by transferring the current state of the data elements. In REST architecture there are six different data elements. These are resource, resource identifier, resource metadata, representation, representation metadata and control metadata. [1]

Resource

The main building block of REST is a resource; any information that can be given a name can become a resource. A document, webpage, a person or current Helsinki weather reports can be a resource. A resource is a conceptual mapping of entities. It is not the entity that corresponds to the mapping at any particular time. Some resource might be static but another one might show variation in time but the semantics that maps to the resource must always be the same since it is the factor that is used to identify one resource from the other.

Fielding discusses that abstracting the resource has given the foundation for the web to operate the way it does. First, the generality of the definition allows having many different sources without artificially differentiating them on their type or the way they are implemented. Second, it allows late binding of a reference to a representation, allowing content negotiation to take place depending on the characteristics of the request. Finally, the author will have the advantage of referencing the concept and not the singular representation of that concept. As a result there is no need to change the existing link whenever the representation changes [1]

Resource Identifiers

A resource Identifier provides means to access a resource at a particular time. REST architecture based on HTTP (Hyper Text Transfer Protocol) uses URI (unique resource identifier) as resource identifier, the URI is the hyperlink to the resource. The URI is the only way that client and the server exchange the representation of the resource. [2]

Resource Metadata

Resource metadata provides additional information about the resource. The information contained in that the resource metadata can be different type such as location information, alternative resource, and identifiers for different formats or additional information about the resource itself. [2]

Representation

Different component in REST architecture (the client and server) perform different actions and communicate with each other by capturing the current state of the resource in to the representation. Representation consists of a set of bytes and representation metadata which describes those bytes. A resource can have more than one representation, for example weather report on a specific period of time can have a representation in JSON (JavaScript Object Notation) or XML (Extensible Mark-up Language) format. [2]

Representation metadata

Representation metadata provides additional information about the representation. Metadata are usually in name-value pair, where the name refers to a standard that is used to define the value and semantics of the value. [2]

Control metadata

Control metadata defines the reason for the communication between the connectors. Media types are the data format of the representation. A representation that is included in a message can be processed by the recipient based on the media type and control data included in the message.

2.3 API

The term API stands for application programming interface. An API can be defined as the technique by which two computer programs intact with each other by means of one common language that they understand. API can provide a hook, a set of protocols and tools for third party developers to access data and services to build a computing application. An API can be free for use for any developer and it can be available to business partner or it can be used by one organization for collaboration between teams. In an API world there are two actors involved the one who provides the API called an API provider and the other who can use the API to build application called the API developer (client). A software company releases its API to the public so that other software developers and companies can build a product on top of its service or platform. [3]

APIs can be designed for a web, operating system or database system. For example programmers who are developing an iPhone application can use the IOS for iPhone

API to interact with the hardware, such as accessing the camera on IOS based devices. Other operating systems such as Windows, UNIX and Mac OS also provide application-programming interface for programmers. Database APIs are also called database abstraction layers. They reduce the workload of a developer by providing a consistent API, which hides the database specifics as much as possible.

2.4 Features of good API

While looking for features that makes an API good, it is difficult to come up with conventional features accepted by everyone. Nevertheless, there are common features that are acquired through experience, which helps us to categorize APIs as good or bad.

Developer friendly

Making an API developer friendly is one of the core features of a good API. Even though the primary objective of an API is to make a connection between systems, humans establish this interaction. Making an API developer friendly not only makes the task of the developer easier but will also play a significant role in increasing the number of client applications using the API. There are some common practices that make an API developer friendly. These include applying commonly used communication protocols, providing access points that are easy to remember and providing a uniform interface. [4]

Extensibility

There are many reasons why an API should be extensible and the degree of its extensibility will make it either a good or a bad API. To handle increasing amount of API calls and to accommodate the ever changing need of client application, developers and to add or remove features. This is one reason why API should be extensible. When an API is extended that is to say, there is a new release of the current version. It is advised to keep both the old and the new version in operation until all the API users migrate to the new version completely. [4]

3 REST Web Service on HTTP

HTTP is categorized under application-level-protocol that defines different operations for transferring information between the client and the server. Different operation can be made on the resource using the methods GET, POST, PUT and DELETE. Using HTTP protocol avoids the need to invent an application specific operation. [5]

3.1 Maintaining Application State

Interactive web application expects clients to perform a sequence of steps to achieve a specific goal. Since HTTP is a stateless protocol it makes it difficult for the server to communicate with the client without implementing some sort of technique. The technique of maintaining the application sate provides means to overcome this challenge. Application state is the state that the server needs to maintain between each request for each client. [5]

There are two possible ways to encode the application state depending on the size of the state. If the size is small, it is possible to encode the application state on URIs and include those URIs into the representation via a link. Whenever the size of the state is too large to encode on the URI, it is possible to store the state on the database or file system and include the reference to the state in the URIs. [5]

Choosing to encode the application state either on the link or in the database has it own advantages and disadvantage in both cases. Encoding the state on the link reduces network performance as the client is expected to send all the information in every request. On the other hand this can increase scalability and reliability since the server is not expected to save data related to the state. It is possible to implement a combined approach of the two cases or to use one or the other according to the specific use case and the size of the application. [5]

3.2 Designing a Resource

Designing the resource model is one of the first steps in building restful web service. In restful web service any information that is given a name can be a resource, it is a conceptual mapping of entities. There is no restriction in defining a resource but it is important to use efficiently the uniform interface nature of HTTP in designing web service. Use case and description of web service can be used as a source for identifying a resource. [5]

By Identifying the nouns in the use case it is possible to apply the operation “create”, “read”, “update” or “delete” on each resource by using the HTTP methods “POST”, “GET”, “PUT” and “DELETE” respectively. For example in a video blogging application, the video can be considered as a resource entity. The blog admin can add new video, change existing video, and view as well as delete the video by implementing the proper HTTP method. These days CRUD operation only gives part of functionality provided by web services, and as a result it is not advisable to identify resources only based on domain noun. In case of transferring money from one account to the other one or merging two-address book, it easy to locate the noun. However, CRUD operations cannot be implemented using HTTP methods without further analysis. [5]

Mapping database table or object model to the resource may not be advisable when building an efficient web service. Factors such as domain modeling and efficiency in data accessibility influence the design of object model and database. HTTP client access resource over the network using uniform interface of HTTP protocol. As a result, it is advisable to design resource based on client usage, but not when designing database table or object model. Generally it is a good approach to consider the client perspective and network performance while designing a resource. [5]

A resource can be grouped into a collection to increase the performance and reliability of web service. Organizing resources in a group helps to access a number of similar resources at once or to create a new resource form the existing ones. Similar resources can be identified and grouped based on application specific criteria, for instance a resource that share a common database schema or a similar set of attributes and properties can be grouped in to a collection. A collection of resources cannot necessarily imply hierarchical structure. A specific resource can be a member of more than one collection at any given time. Considering a social media web service such as, user resource can be part of several collections such as “users”, “friends”, “followers” and “swimmers”. [5]

Another important point to consider while designing a resource is how to handle computing or processing function. The HTTP GET method can be used to fetch the representation that contains the output of the resource and query strings can be used to provide input for the processing function resource. Finding the distance between two coordinates or requesting a direction from point A to point B can be treated as a resource. In The above example, the distance calculator function can be taken as the resource and the actual distance as the representation. [5]

3.3 Designing Representation

Representation is a sequence of bytes and metadata that describes the byte. Representation metadata are implemented as name-value pair using entity header. Entity headers are important because they contain valuable information on the visibility and discoverability of a representation .HTTP protocol provides a ready-made format to construct a representation in request and response. [5]

Designing a representation requires using the ready-made format to construct the right headers, choosing the proper media type and designing the format of the body. HTTP protocol is designed in such a way that the sender can describe the body of the representation using the header section. As a result the recipient can decide how to handle the request by using the header information. The header section of a representation includes content-type, content-length, content-language, content–MD5, content-encoding and last-modified. [5]

HTTP’s representation format is designed in such a way that it allows flexibility in the choice of media types for the request and the response. Some resource may require XML formatted representation while others may require HTML representation or PDF formatted representation. Leaving room for this flexibility is an important part of designing a representation. The best approach in the decision to choose format and media type is to let the use case and type of client make the decision. As a result, it is strongly recommended not to use application framework that dictates to use only one or two media formats. [5]

3.4 Designing URI

The way URI (Uniform Resource Identifier) is designed is one of the factors that affects the success building efficient and robust web service. Following a convention while designing URI has a number of advantages. It simplifies managing and debugging, reduces time spent in creating new URIs, gives operational flexibility and assists servers to centralize codes to extract data from request URI. [5]

Logically designing URIs into domains and sub domains has a number of advantages for server administration.

http://en.example.com/book/1212
http://fi.example.com/books/1212

In the above example logically naming subdomains can help the server to give localized service.

http://www.example.com/book/1212
http://api.example.com/books/1212

The first URI provides representation for web browser and the second for custom client. Partitioning can enable the server to allocate different hardware and apply different routing, monitoring and security policies for HTML and non-HTML representations. [5]

The following list provides some conventions in designing URI

- Use subdomains and domains to logically group resources
- Use forward slash (/) to indication hierarchical relation between resources
- Avoid including file extension in the URIs
- Use comma (,) and semicolon to indicate non-hierarchical element the path portion of the URI
- Use hyphen (-) and underscore (_) to increase readability [5]

3.5 Web Catching

Catching is one of the most important features built on top of HTTP’s uniform interface. Catching provides a number of useful advantages while building web service. Reducing end user perceived latency, increasing reliability, reducing bandwidth usage and reducing server cost are among the most important benefits of web catching. It is possible to implement web catching anywhere between server and client in HTTP protocol. [5]

The two ways of implementing web catching are HTTP catch and Object catch. While both ways improve performance and have a significant role in web service deployment architecture, there is a clear difference in their implantation. Catching in HTTP does not expect clients and servers to implement any specific programming API to manage the data in the catch. HTTP catch are based on the uniform interface feature provided by REST architecture. In the case of Object catching, it is necessary to a specific programming API is necessary to store, retrieve and delete the object data. [5]

In HTTP when clients perform operations using the methods the “PUT”, “POST”, and “DELETE” methods for a specific resource, the catch is expected to invalidate the representation corresponding to that resource. On the other hand, when a client makes a “GET” request on the same resource, the catch will obtain a fresh representation from the server. [5]

3.6 Security

Security when building web service covers a wide range of issues and different aspects. Authenticating users to access a resource, confidentiality of information, preventing unauthorized agents form accessing resources and following the laws of the country must be considered carefully. There is no single conventional to follow but each application requires a specific analysis in the architecture and design pattern. [5]

When a client accesses a protected resource, the sever uses an authenticate header to challenge the client to provide the expected information. The client uses authorization header to provide the proper answer. This kind of authentication schema can be used when a client accesses a protected resource on behalf of itself or user. [5]

Developed in 2007 OAuth is an authorization protocol that allows user to give permission for clients to access their data on the server without revealing their identity. OAuth protocol can be categorized as two-legged or three-legged depending on the number of parties involved in the authentication process. The name three-legged authentication is given to OAuth because there are three parities involved in the authentication process: the service provider (the server), the OAuth consumer (the client) and the user. In case of two-legged OAuth, the parties involved in the authentication process are only the service provider (the server) and the OAuth consumer (the client). [5]

The following list shows the steps in the authentication of three-legged OAuth:

- Client request authorization to access resource from the user
- The user grant authorization for the client
- The client request access token from the server
- The server issues access token to the client
- The client request resource from the server
- The server provides the resource for the client [6]

4 REST API in Social Media

4.1 Twitter API

One of the main reasons why Twitter API is successful is because it is so simplicity. Twitter API is not only simple for its user but also for the developers who build different applications that take advantage of the large user base and huge bulk of data generated by its users. The twitter API provides a gateway for much of Twitter's core functionality such as status update, direct messaging and search. As the Twitter platform evolves, so does the API to accommodate the newly introduced features for API users. [7]

The Twitter API enables mobile, desktop and other Internet based third party applications to interact with Twitter service in a standard and simple way. The API enables data to be shared from Twitter to other applications. By sending a request to the Twitter API, data is returned as a response in structured formats, which makes it easier to extract the data from the response. The Twitter API changes the functionality of the Twitter into small manageable units via a number of available methods. [7]

Twitter REST API

Twitter API is called REST API because it implements GET method to retrieve data from its database. Methods that are used to create, update and delete data require POST operation. [7]

The three elements of the Twitter API are HTTP operation, API method and return format. HTTP operation handles how the request is sent from third party application to the Twitter API. These transfer operations are GET, POST, UPDATE and DELETE. [7]

Twitter API Parameters

The importance of Twitter parameters lies in the fact that they can customize the output of the request sent to the Twitter API. The parameters can also affect the request data sent to the API. It is possible to set a parameter either by appending them to the method request or including them in POST, PUT or DELETE operations. The Twitter API uses UTF-8 character encoding for all parameters. Using special characters like ampersand (&) and equal (=) must be encoded before being sent to Twitter API. Encoding special characters can take more storage space than a single byte character. This means that some request might be rejected if they are more than 140 characters. [7]

Twitter API Methods

The Twitter API is divided into several method categories. These methods include one for sending updates, following and not following users, and account management. The search API contains methods for extracting searches and trend information from Twitter as a means of filtering, finding and sorting huge volumes of data. [7]

Twitter API Return formats

JavaScript Object Notation is a lightweight data-interchange format preferred in AJAX (Asynchronous JavaScript and XML) applications and it is considered a simpler and faster alternative to XML. JSON is language independent, self-describing and easy to use. JSON is object-based and simple text can be used to represent different data type and relationships. [7]

Really Simple Syndication or RSS is a standard XML format used on blog and news sites. Atom was created as an alternative to RSS to cover some of the flaws in RSS and to improve international support. [7]

Extensible Mark-up Language is a general-purpose language. When using Extensive Mark-up Language, Users can define their own tags and structure. [7]

Authentication with Twitter OAuth.

An OAuth is an open protocol for authentication of software application. Twitter OAuth provides a means for user to interacting with twitter without providing their credential for the application. Twitter OAuth, formally referred as “sign in with twitter”, enables users to sign into a different application using their Twitter account. [7]

4.2 Facebook REST API

Facebook is the biggest and most used social media of our time. Since July 2016 there are around 1.65 billion monthly active users. For developers Facebook has opened a lot of possibilities and opportunities to create interactive application. Facebook integration allows developers to hook in to its ecosystem through an external application on the web, phone or game console. The Facebook platform for website enables developers to integrate Facebook into a website, desktop application and native mobile application. [7]

Facebook API SDK

For seamless integration of website and third party application with Facebook platform, Facebook provides a Software Development Kit or SDK in PHP and JavaScript scripting languages. Both PHP and JavaScript are common scripting languages of the web and are supported by almost all browsers. The Facebook SDK for JavaScript provides a rich set of client side functionality, including Facebook login, API access and social plugins. On the other hand, Facebook SDK for PHP provides rich server side functionality for accessing Facebook server side API calls.

Facebook Widgets

Facebook provides a number of widgets that can be integrated in a third party application. These widgets help to promote Facebook contents on external platform. Box and Live Stream Box are examples of Facebook widgets. These widgets are highly customizable and use Facebook platform library to display dynamically updated information from a Facebook page. [7]

Registering Facebook Application

Facebook Application is a program written by developers, which extract data from a Facebook platform. Facebook assigns each application a unique ID and a private key for the purpose of security and to distinguish between various applications. The application ID and secret key are unique for each Facebook application. This information is required while requesting any data from Facebook to make sure that the request is coming from a trusted source. [7]

Authentication with Facebook API

Facebook can be used as a login mechanism for website or Internet enabled application. Facebook provides a login system for a third party application both in PHP and JavaScript API libraries. The Facebook Authorization workflow keeps users information secure.

When users visit a website using Facebook, they can be in one the three states: connected, which means they are connected to Facebook and authorized the application; not logged in, which means they are not logged in to Facebook, so their Facebook status cannot be evaluated. The third state is “not authorized”, which means users are logged into Facebook but not connected with the application. [7]

Connecting and Inviting Friends

Once users are connected to a third party application using their Facebook account, they can invite their friends to join the application. A special Facebook connect request can be sent to friends to encourage them to sign in via Facebook. Facebook provides this functionality via <fb: connect - form> XFBML element, which renders an invitation widget on the page. [7]

Facebook Graph API

The Facebook Graph API is the main way to get data in and out from Facebook’s social graph. It is an HTTP based REST API that is used to accomplish different sort of tasks from a third party application. Social graph is a representation of information in Facebook. The social graph has three important components, which are nodes, edges and fields. Nodes are things such as photo, comment, users and a page, whereas edges are connection between the nodes. Fields in the social graph represent information of things in the nodes. For example user’s name and birthday or name of a Facebook page can be considered as fields. For the purpose of reading data from the graph, an API HTTP GET request is sent to the appropriate endpoint. All graph API publishing is done by using the HTTP POST request to the relevant edge with the necessary parameter included. There is a large number of edges that can be a target. All publishing calls must include an access token in the POST request. [8]

5 Application Project

5.1 Project Overview

The main objective of this web-based application is to facilitate the process of sponsoring a child's education in a developing country. Unlike other systems, this one is targeting the diaspora communities of developing countries as a donor investing to sponsor the child. The application has a feature which enables the interested donor to assign one person in the country of his or her origin to take care of all the necessary tasks on the ground, such as choosing the right child and delivering the fund every month to cover the child's educational and basic expenses. The following steps show the workflow how to sponsor a child in the application:

- A sponsor registers in the application
- The sponsor will send an invitation for one friend to join the app
- The friend will accept the invitation and joins the app
- The friend will search for the right child to be sponsored
- The friend will add all the information needed about the child in the app
- The sponsor starts sponsoring the child through the friend he or she chooses
- The friend will spend the money on the child's educational expense

5.2 Technologies Used

HTML5

HTML5 is the latest version of HTML commonly used in current web-based applications. HTML is mark-up language that consists of different mark-up tags, which are used to describe the content of the web page. The web browser renders this mark-up language and displays the final output for the user. [9]

PHP

PHP is a widely used general purpose scripting language. PHP is one of the programming languages that are categorized under Server Side Scripting Language. This means that PHP language is executed on the server and the output is sent to the browser embedded on HTML page. [10]

JavaScript

JavaScript is a programming language, which is used to a web page more interactive with the user. Web browser rendering engine renders scripting language codes written with JavaScript. JavaScript is also known as an interpreted language which means no special program is required to run the code apart from the web browser. JavaScript is supported by most of the browsers that are in use on the Internet. [11]

CSS (Cascading Style Sheet)

CSS is a style sheet language used to describe how content produced using a mark-up language is presented. The primary use of CSS is to separate the document content from the document presentation. It is used to change different aspects of the presentation such as the font, color and layout of the document. [12]

Bootstrap

Bootstrap is a front-end framework used to design a responsive and mobile first web application. Bootstrap is built using the common technologies of the web such as: HTML5, JavaScript and CSS3. Bootstrap is open source software which means that it free to download and use. Bootstrap makes web design much simpler and faster by providing different features and technologies, these features help, to implement design layout in a faster and simpler way than if we try to achieve the same task using raw HTML and CSS. [13]

MySQL

MySQL is the most popular database management system. It is developed, distributed and supported by Oracle that is one of the successful tech companies of our time. MySQL is relational database system that means information is separated into different tables and linked to each other, rather than arranging and storing it a single place. MySQL is an open source system which anyone can download, use and even modify without any need for payment. [14]

Object Oriented Programming (OOP)

OOP is not a software or a programming language but it is a way or paradigm to write a computer program or web based application. The main idea behind OOP is considering a programming unit like any real object in everyday life. Designing computer program in this manner will enable the program to interact in the same way as a real object. Objects in OOP have attributes and properties that are called methods in programming term. Considering a student object in OOP, the student name, id, study year and credits completed can be considered as an attribute, whereas adding and dropping a course, taking an exam, submitting assignments and graduating can be considered as its behavior or properties. Each object in OOP contains its own attributes and method and communicates with other objects in the application. [15]

5.3 Tools Used

Aptana Studio

Aptana Studio is an open source integrated development environment that is used to build modern web based application. It assists to develop and test an application using a single environment. It has a complete support for the latest web technologies such as HTML5, CSS3, JavaScript, Ruby, Rails, PHP and Python. The core features of Aptana Studio include code assistance, built-in terminal, Git integration and IDE customization. [16]

PhpMyAdmin

PhpMyAdmin is open source software written in PHP to manage MySQL database on the web. PhpMyAdmin supports a wide range of operation on MySQL database management system. These operations include creating and managing databases, tables, columns, relations, indexes, users and permissions among others. PhpMyAdmin provides both user interface and the ability to execute raw SQL to achieve the different operations. PhpMyAdmin has a documentation that explains the different aspect of its operations. [17]

FTP Client Software

FTP client is a software used for the transmission of files over the Internet. The software should be installed on a computer and there should be live Internet connection for the transmission of files to be successful. In the case of this specific application, an FTP Client called Transmit is used to transfer the application file from the development environment to the hosting server. [18]

5.4 Technical Analysis

MVC

The application in the project is designed and built according to the principles of MVC. MVC is a software architecture pattern that stands for model view and controller. Each of these three components is built to handle a specific section of the application. The model is responsible for handling all data related logic of the application. The view is responsible for all logic related to the user interface of the application. The controller is the component that bridges between the view and the controller and it is responsible for the business logic of the application. [19]

Class and Object

In object oriented programming an object is a self-contained and fundamental building block that is designed to achieve a specific task. A class is a blueprint that contains a set of instructions to build a specific object. In every class there are two important components that are the attributes and methods. Attributes are field types and names that can be turned in to variables during runtime. Methods define the behavior of an object. It can also be expressed as an action that an object is able to perform.

Since each class is designed to accomplish a specific task, there are a number classes in the application. The type of class includes Session class, Database class and another class that are directly linked with tables in the database design. The session class is used to manage all issues related to authentication in the application. The database class manages tasks related to creating, updating deleting and editing data from the database.

In addition to the core class file there are a number of other PHP files that implement the view and controller in the application. Views are responsible for displaying the data to the user while controllers facilitate the interaction between the core model classes and the views.

Database design

The database in the application is designed as a relational database. A relational database is made up of tables that contain rows and columns. The columns are also called attributes that are directly mapped with the model classes in the application. In the database design the tables are related with each other. The type of relationship that exists in the design is one-to-many. One-to-many relation between two tables A and B is described as follows: table A can be linked to many elements in table B whereas table B is linked only by one element to table A.

In the database design the concepts primary key and foreign key is implemented. Each table in the database has a column that uniquely identifies a record in the table, this field is known as a primary key. Primary keys should always be unique, have a value (it shall not contain null) and should be simple. When two tables are related in one-to-many relation, a primary key in the first table can be a foreign key in the second table. Foreign keys are very crucial for creating relationships between tables. If we look at the database design for the application, there is a foreign key d_id field in the friend table which is primary key in the donor table. While establishing a foreign key between two tables, there is a foreign key constraint that determines the action on the second table while a certain operation is made on the other table. Foreign key constraint also prevents invalid data from being inserted in the foreign key column.

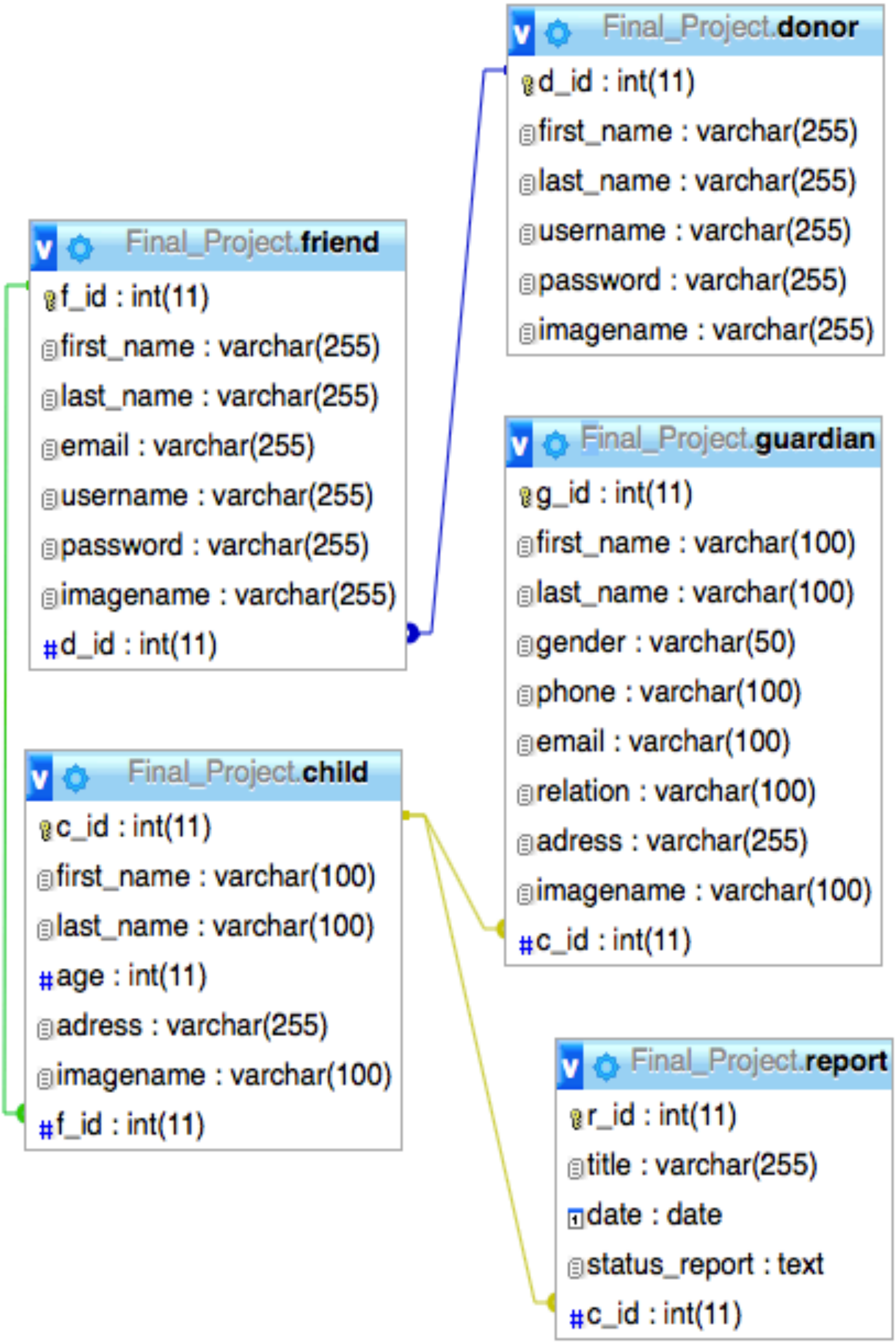
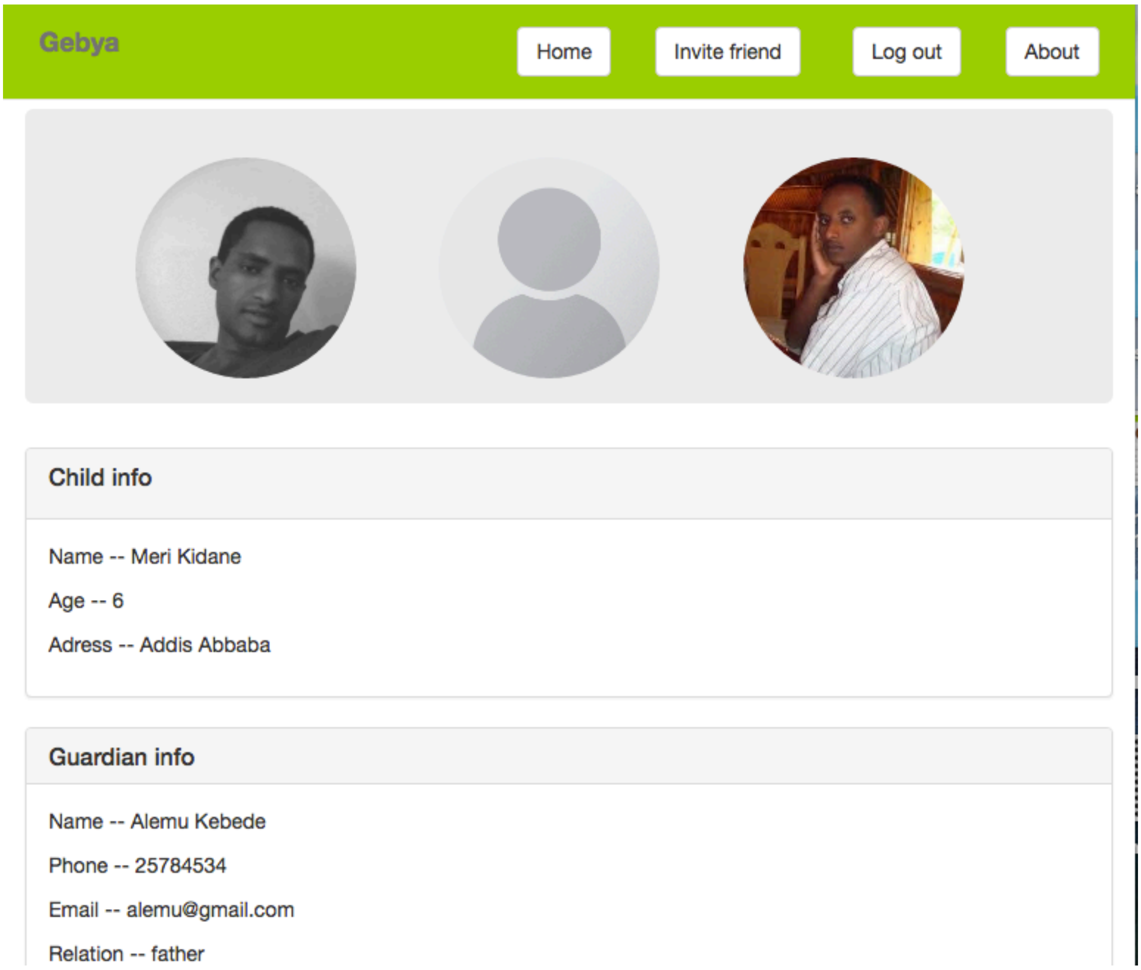


Figure 1. Database table schema for the application

User Groups

There are two main user groups in the application, Donors and Friends. Donors start the process for sponsoring the child. After signing up in the application as a sponsor, a donor can send an invitation for a friend by using the invite friend feature in the donor's dashboard. In the donor dashboard the donors can also follow the entire task performed by the friend they have chosen to facilitate the process from the other end.



Child info

Name -- Meri Kidane

Age -- 6

Adress -- Addis Abbaba

Guardian info

Name -- Alemu Kebede

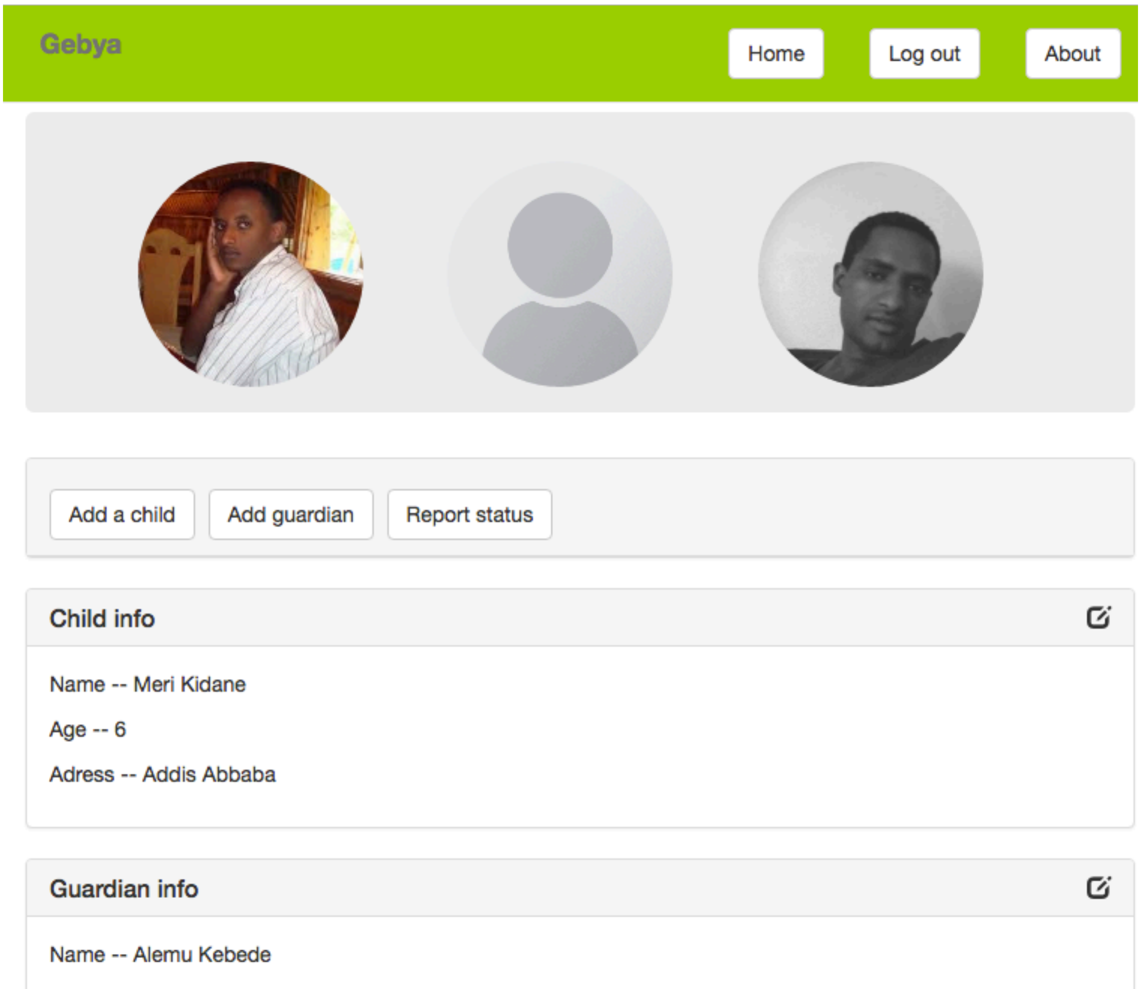
Phone -- 25784534

Email -- alemu@gmail.com

Relation -- father

Figure 2. Sample dashboard for donor.

Friend is the second user group for the application. After being invited by a sponsor, a friend will receive an email. In the email there is a link that redirects the friend to a register page, where he or she can register as a friend. The friend has a dashboard with a number of features to add all information regarding the child who is going to be sponsored, the guardian of the child and report on the status of the child.



Child info

Name -- Meri Kidane

Age -- 6

Adress -- Addis Abbaba

Guardian info

Name -- Alemu Kebede

Figure 3. Sample dashboard for friend.

Integrating the application with Facebook

There are a number steps that should be followed to integrate the application with a Facebook platform. The first step is hosting the application online. It is not possible to integrate Facebook APIs in local hosting environment. It is also advisable to have SSL certificate for hosting the application. Since the project application is built mainly using PHP, it will be reasonable to use the PHP software development kit (SDK). The PHP SDK should be downloaded and saved in the folder where the application is hosted. [20]

The main system requirements that must be fulfilled to integrate the application with Facebook are PHP 5.4 or greater version and mbstring extension. Mbstring provides a multibit specific function that helps to deal with multibit encoding in PHP, it also manages character-encoding conversion between the possible coding pairs. [20]

The next step in the process of integration is to have a valid Facebook account and afterwards registering as Facebook developer. After registering as Facebook developer, the application should be registered in the Facebook platform. The App ID and App secret will be automatically generated for this specific application; these values will later be used in the code. [20]

Integrating application with Facebook is quite of complex. It means adding ‘like’ and ‘share’ features to the authentication with Facebook and making a complex request to the graph API. The goal of adding Facebook functionality by logging in using Facebook credentials is to enable users to integrate their activity in the application with their Facebook stream.

6 Conclusion

In this thesis I have discussed what restful APIs are and the fundamental factors to consider when building a web service on top of HTTP protocol. REST is a software architecture style that can be described with certain constraints and that it is resource based. The fundamental working principle and basic concept behind the World Wide Web has been discussed. Discussing these important concepts made it easier to understand what social media APIs are and the technical steps required when integrating a web application.

Integrating a web application with social media by using APIs is becoming crucial due to the huge number of their user base. Integration of web application with social media API's like Facebook cover a wide variety of issues and ranges. These issues vary from adding simple features such as 'share' and 'like' buttons to complex activity such as making a complex request to the graph API.

In this thesis I have described how to integrate the Facebook login system with a third party application. This process involves many steps, starting from having a valid developer account on Facebook to implementing the code that enables to make complex request to the graph API.

The topic of social media APIs is a wide and complex topic. This thesis covers the fundamental concepts and some technical issues regarding the topic. There are many books, online material and above all documentation in the specific social media. The following links provide concise and up-to-date information on the subject: https://developers.facebook.com/, https://dev.twitter.com/.

References

1 Fielding R. Architectural Style and the Design of Network-Based Software Architecture. University of California: 2000.
URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
Accessed: 17 September 2016.

2 Manisha P. REST Architectural Elements and Constraints [online]. www.web-03.net
URL: http://mrbool.com/rest-architectural-elements-and-constraints/29339. Accessed: 4 September 2016

3 Jacobson D, Brail G. APIs: A Strategy Guide. O'Reilly Media: 2012.

4 Doglio F, Castro J. Pro REST API Development with Node.js. Apress: 2015.

5 Subbu A. RESTful Web Services Cookbook. O'Reilly Media: March 2010.

6 Digitalocean. An Introduction to OAuth 2 [online]. 2016.
URL:
https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2
Accessed: 15 September 2016.

7 Mark D. The Developer's Guide to Social Programming: Building Social Context Using Facebook, Google Friend Connect and Twitter API. Addison – Wesley Professional: August 2010.

8 Facebook Developers. Graph API Overview [online]. 2016.
URL: https://developers.facebook.com/docs/graph-api/overview/ .
Accessed: 8 July 2016.

9 W3schools. An Introduction to HTML5 [Online] 2016.
URL: http://www.w3schools.com/html/html5_intro.asp
Accessed: 4 September 2016.

10 W3schools. PHP 5 Tutorial [Online] 2016.

URL: http://www.w3schools.com/php/default.asp
Accessed: 4 September 2016.

11 W3schools. JavaScript Introduction [Online] 2016.
URL: http://www.w3schools.com/js/js_intro.asp
Accessed: 4 September 2016.

12 W3schools. CSS Introduction [Online] 2016.
URL: http://www.w3schools.com/css/css_intro.asp
Accessed: 4 September 2016.

13 W3schools. Bootstrap 3 Tutorial [Online] 2016.
URL: http://www.w3schools.com/bootstrap/default.asp
Accessed: 4 September 2016.

14 ORACLE. What is MySQL [Online] 2016.
URL: http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html
Accessed: 4 September 2016.

15 ORACLE. Object Oriented Programming Concepts [Online] 2016.
URL: https://docs.oracle.com/javase/tutorial/java/concepts/
Accessed: 6 September 2016.

16 Appcelator, INC. Aptana Studio 3 [Online] 2016.
URL: http://www.aptana.com/products/studio3.html
Accessed: 6 September 2016.

17 phpMyAdmin. Bringing MySQL to the Web [Online] 2016.
URL: https://www.phpmyadmin.net/
Accessed: 6 September 2016.

18 Softonic. Transmit for Mac [Online] 2016.
URL: http://transmit.en.softonic.com/mac
Accessed: 6 September 2016.

19 Microsoft. ASP.NET MVC Overview [Online] 2016.
URL: https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx
Accessed: 6 September 2016.

20 Facebook Developers. Getting Started with Facebook SDK for PHP [online]. 2016.
URL: https://developers.facebook.com/docs/php/gettingstarted.
Accessed: 8 June 2016.

21 Facebook Developers. Login with Facebook [online]. 2016.
URL:
https://developers.facebook.com/docs/php/howto/example_facebook_login.
Accessed: 12 June 2016.

PHP Script that Handles Facebook Login. [21]

```
<?php
session start();
require once __DIR__ . '/src/Facebook/autoload.php';

$fb = new Facebook\Facebook([
  'app_id' => '810840495682536',
  'app_secret' => 'fa83d3ae3f6d1cf33dd569b3c4ca16de',
  'default_graph_version' => 'v2.4',
  ]);

$helper = $fb->getRedirectLoginHelper();

$permissions = ['email']; // optional

try {
    if (isset($_SESSION['facebook_access_token'])) {
              $accessToken                            =
$_SESSION['facebook_access_token'];
    } else {
              $accessToken = $helper->getAccessToken();
    }
}  catch(Facebook\Exceptions\FacebookResponseException  $e)
{
    // When Graph returns an error
    echo 'Graph returned an error: ' . $e->getMessage();

    exit;
} catch(Facebook\Exceptions\FacebookSDKException $e) {
    // When validation fails or other local issues
    echo  'Facebook  SDK  returned  an  error:  '  .  $e-
>getMessage();
    exit;
 }

if (isset($accessToken)) {
    if (isset($_SESSION['facebook_access_token'])) {
              $fb-
>setDe-
faultAccessToken($_SESSION['facebook_access_token']);
    } else {
              // getting short-lived access token
              $_SESSION['facebook_access_token']          =
(string) $accessToken;

              // OAuth 2.0 client handler
              $oAuth2Client = $fb->getOAuth2Client();

              // Exchanges a short-lived access token for
a long-lived one
              $longLivedAccessToken     =     $oAuth2Client-
>getLongLivedAc-
cessToken($_SESSION['facebook_access_token']);

              $_SESSION['facebook_access_token']          =
(string) $longLivedAccessToken;

              // setting default access token to be used
in script
              $fb-
>setDe-
faultAccessToken($_SESSION['facebook_access_token']);
    }

    // redirect the user back to the same page if it has
"code" GET variable
    if (isset($_GET['code'])) {
              header('Location: ./');
    }

    // getting basic info about user
    try {
```

```
                $profile_request = $fb-
>get('/me?fields=name,first_name,last_name,email');
                $profile = $profile_request->getGraphNode()-
>asArray();
     } catch(Facebook\Exceptions\FacebookResponseException
$e) {
                // When Graph returns an error
                echo 'Graph returned an error: ' . $e-
>getMessage();
                session_destroy();
                // redirecting user back to app login page
                header("Location: ./");
                exit;
     } catch(Facebook\Exceptions\FacebookSDKException $e) {
                // When validation fails or other local is-
sues
                echo 'Facebook SDK returned an error: ' .
$e->getMessage();
                exit;
     }

     // printing $profile array on the screen which holds
the basic info about user
     print_r($profile);

     // Now you can redirect to another page and use the
access token from $_SESSION['facebook_access_token']
} else {
     $loginUrl = $helper-
>getLoginUrl('https://ethiosew.com/geba/fb_index.php',
$permissions);
     echo '<a href="' . $loginUrl . '">Log in with Face-
book!</a>'
} ;
```