

# **Тема: Разширяване на проекта по Web технологии за - Project Estimation**

Предмет: Приложно-програмни интерфейси за работа с  
облачни архитектури с Амазон Уеб Услуги (AWS)

Изготвил: Диана Иванова, фн: 81426, имейл:  
didi.e.ivanova@gmail.com

Лектор: Милен Петров, година: 2020

# Съдържание

1	Условие	3
2	Въведение	3
3	Теория	3
4	Използвани технологии	4
5	Инсталация и настройки	4
6	Кратко ръководство за потребителя	7
7	Примерни данни	11
8	Описание на програмния код	12
9	Приноси на студента, ограничения и възможности за бъдещо развитие	14
10	Какво научих	14
11	Списък с фигури и таблици	15
12	Използвани източници	15

# 1 Условие

Документиране на разгръщане на съществуващ проект по уеб технологии (Project Estimation) - под формата на AWS услуги, разширяване на на проекта с поне една услуга на амазон и интегриране с някоя от другите услуги на AWS.

## 2 Въведение

В настоящата документация е представено описание на системата и разгръщането и върху облачна архитектура, инсталация и ръководство. Project estimation е система за възлагане и автоматично оценяване на задачи и проекти, както и за визуално представяне на данните под формата на burn-down chart. Използваните услуги и тяхното приложение са както следва:

- Потребителите могат да се регистрират и да влизат в уеб-сайтът с име и парола, като за целта е използвана aws услугата Cognito.
- Базите от данни ще бъдат разположени в облака и ще се оперира с тях посредством Amazon RDS..
- Останалата част от програмата работи сървърфул върху инстанция на EC2.

## 3 Теория

### Разгръщане:

Използвайки Amazon RDS за MySQL, базата данни ще бъде върху отделна инстанция от тази на уеб сървъра, така че те няма да се състезават за ресурси. Amazon RDS за MySQL има автоматизирани резервни копия и ъпдейти, спомагащи за администрацията на базите данни. Security group-ата на RDS има Inbound правило за достъп от тип MySQL на порт 3306 с източник security group-ата на EC2 инстанцията. По този начин EC2 достъпва данните и може да прави заявки към RDS базата данни. От своя страна security групата на EC2 има за правило Inbound правило тип HTTP, порт 80 и източник всеки, като позволява да може да се достъпва уеб-сайтът от всички. EC2 има и SSH правило на порт 22 с източник моят ip адрес, за да мога да правя промени по сайтът и да го поддържам.

Аутентикацията става с помощта на AWS Cognito, като при всяка регистрация системата изпраща имейл за потвърждение към пощата на потребителя.

Amazon Cognito е услуга за синхронизация на идентичността на потребителя, която позволява управление на потребителите и администрация. Всички скриптове, html и css документи са на EC2 инстанция, върху който работи apache сървър.

#### **Функционалност:**

Системата за оценяване борава с два типа оценки - експертна и реална. Експертната оценка за една задача се формира на база тагове - css, event, insert into db, query from db и още. На всеки един таг съответстват часове работа (пр. event - 1, query from db - 2). Потребителят на системата въвежда реалната оценка след като завърши задача. Двата типа оценка за самия проект се формира като сума от съответните типове оценки на всички задачи по проекта. На базата на тези данни се изготвя burn down диаграма.

## **4 Използвани технологии**

- **MySQL** - Ver 15.1 Distrib 5.5.64-MariaDB, for Linux (x86\_64) using readline 5.1
- **PHP** - версия 7.2
- **Apache** - Apache httpd 2.4.41
- **Javascript JQuery** - version-3.1.0
- **mazon-cognito-identity-js** - version 1.31.0
- **CSS**
- **AWS EC2** - Amazon Linux 2 AMI (HVM)
- **AWS RDS**
- **HTML**
- **AWS Cognito**

## **5 Инсталация и настройки**

**Стъпка 1.** Създаване на User pool в Когнито с име Estimateme. Нека да е с атрибут имейл и верификация на имейла чрез линк. Създадете App Client

с име `EstimateApp` като махнете отметката на `Generate client secret` опцията. Конфигурирайте `config.js` файла. Поставят се съответния регион, `poolId` и `userPoolId`.

**Стъпка 2.** Създаване на EC2 инстанция със следните параметри:

- За AMI изберете `Amazon Linux 2 AMI (HVM)`
- тип на инстанцията `t2.micro`
- Създайте нова Security Group-а с име `estimate` като добавите следните правила 1)SSH трафик от текущия Ви IP адрес за да използвате SSH протокол, за да влезнете във вашата EC2 инстанция и да конфигурирате уеб страницата; 2)HTTP трафик от всички IP адреси така че потребителите да могат да виждат Project Estimation сайтът.
- Накрая запазете Key Pair ключа, за да можете да се логнете в EC2 инстанцията

**Стъпка 3.** Създавайте RDS база данни. Изберете **MySQL** за система за управление на бази данни (СУБД). Изберете за шаблон (template) `free tier`. В Settings секцията, въведете **projectEstimation** за идентификатор на инстанцията база данни. След това посочете master потребителско име и парола за Вашата база данни.

- За VPC изберете „по подразбиране“
- За „Subnet group“ изберете „**rdsfmi**“
- На „Публична достъпност“ изберете „Да“
- За availability zone изберете „**eu-central-1a**“
- За „VPC security groups“ изберете „Choose existing VPC security groups“ и изберете група „**sgFMI**“
- За „Име на базата данни“ въведете „**estimate**“
- За криптиране изберете „**Disable encryption**“
- На „Enhanced monitoring“ премахнете отметката квадратчето пред `Enable Enhanced monitoring`
- За „Защита при изтриване“ премахнете отметката от квадратчето пред `Enable deletion protection`

- Натиснете бутона Create Database
- Разрешете на вашия екземпляр EC2 достъп до вашата RDS база данни по следния начин: кликнете на Security Group-ата. В inbound-таба добавете правило с Type property MYSQL/Aurora и сменете текущата стойност на security групата със "estimateme" security групата, която използвахте за EC2 инстанцията и запазете промените

за прикачване на sql скриптовете от папката sqlscripts.

**Стъпка 4.** Направете SSH връзка към инстанцията. Въведете командата:

```
sudo yum install -y mysql
```

След това намерете името на хоста за вашата RDS база данни в конзолата AWS. В детайлите на вашата RDS база данни името на хоста ще бъде показано като Endpoint в секцията Security and Connection.

Въведете командата:

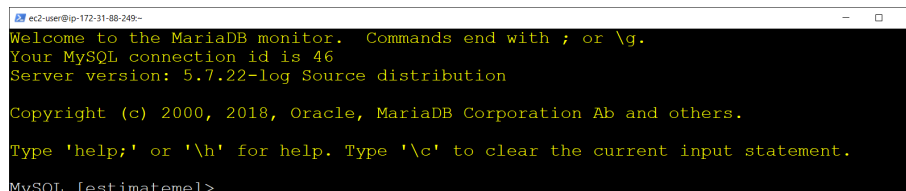
```
export MYSQL_HOST=<your-endpoint>
```

където <your-endpoint> е името на хоста

Въведете командата:

```
mysql -user=<user> -password=<password> estimateme
```

заменете <user> и <password> със maestър потребителското име и паролата, които конфигурирахте при създаването на RDS базата от данни



Фигура 1: Access to Database using Cli

И накрая, създайте потребител на база данни за вашето project estimation приложение и му дайте разрешение за достъп до базата данни „estimateme“. Изпълнете следните команди във вашия терминал:

```
CREATE USER 'estimateme' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON estimateme.* TO estimateme;
FLUSH PRIVILEGES;
Exit;
```

където изберете подходяща парола за password

**Стъпка 5.** За да инсталирате Apache на вашата EC2 инстанция, изпълнете следната команда във вашия терминал:

```
sudo yum install -y httpd
```

За да стартирате уеб сървъра Apache, изпълнете следната команда във вашия терминал:

```
sudo service httpd start
```

Инсталирайте php: Първо проверете дали имате amazon-linux-extras.

```
which amazon-linux-extras
```

ако ги нямате, ги инсталирайте първо

```
sudo yum install -y amazon-linux-extras
```

```
sudo amazon-linux-extras enable php7.2
```

```
# yum clean metadata && sudo yum install php70u php70u-pdo  
php70u-mysqlnd php70u-opcache php70u-xml php70u-gd php70u-devel  
php70u-mysql
```

```
sudo service httpd restart
```

Качете estimateme.zip на ec2 инстанцията:

```
scp -i path/to/key path/to/file user@ec2-xx-xx-xxx-xxx.compute-1.amazonaws.com:~/
```

след което въведете следните команди

```
unzip estimateme.zip
```

```
sudo cp -r main/* /var/www/html/
```

```
mysql -user=<user> -password=<password> estimateme < all_ tables.sql
```

Редактирайте конфигурациите за базата данни във файла **dbConnect.php** като попълните празните полета със съответните стойности:

```
$host = ' ';
```

```
$dbname = ' ';
```

```
$username = ' ';
```

```
$password = ' ';
```

## 6 Кратко ръководство за потребителя

Потребителят достъпва сайта на адрес:

```
http://ec2-18-233-153-235.compute-1.amazonaws.com/register.php
```

Регистрация. Потребителят се регистрира със своя имейл като избира подходящо потребителско име, парола и тип-акаунт. Има два типа акаунти - за



Фигура 2: Диаграма - кратко ръководство на потребителя [PlantUML]



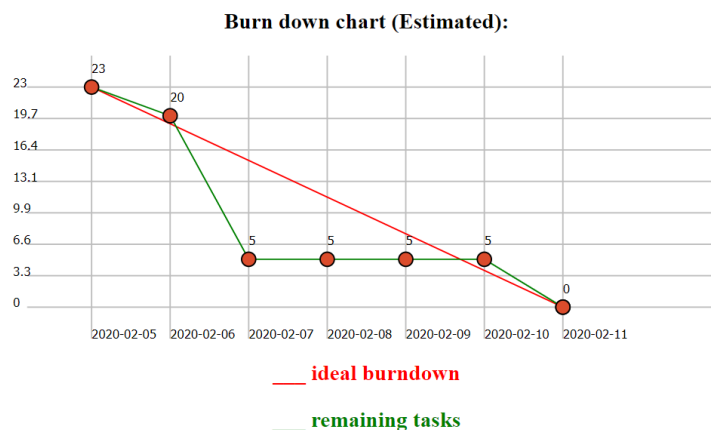
разработчици и мениджъри. Потребителят трябва да верифицира акаунта, като кликне на линка изпратен в пощата му.

Влизане в системата.

Потребителят може да провери текущите проекти, в които участва в страницата “Проекти”. Ако е мениджър ще види и проектите които е създал. Само мениджърите могат да създават проекти.

Потребителят може да разгледа детайлно всеки от проектите, в които участва като кликне върху ‘кутийката’ със съответния проект. Страницата с описание на проекта изброява задачите в него под формата на таблица. Там потребителят може да избере статус на задачата, която му е разпределена. Т.е ако започва задачата трябва да смени статуса и на ‘in progress’ посредством падащото меню. Под таблицата (ако в нея има поне една задача) автоматично се генерира burn down chart на базата на часовете нужни за задачите и целия проект.

Статус	Име на задачата	Експертна сложност
IN PROGRESS ▾ Смени	agag	3
TO DO ▾ Смени	dev2 task	5
IN PROGRESS ▾ Смени	something	9
IN PROGRESS ▾ Смени	баба	6



Фигура 3: View project tasks

Могат да се добавят нови задачи към всеки един от проектите. Избира се име на задачата, възлага и се изпълнител и се добавят тагове, всеки от които

носи часове трудност. Системата автоматично смята трудността на задачата според таговете, така че потребителят трябва да въведе възможно най-точните тагове за конкретната задача. Ако потребителят не е доволен от предоставената оценка, може да я смени в съответното поле. Към задачата може да се добави и описание.

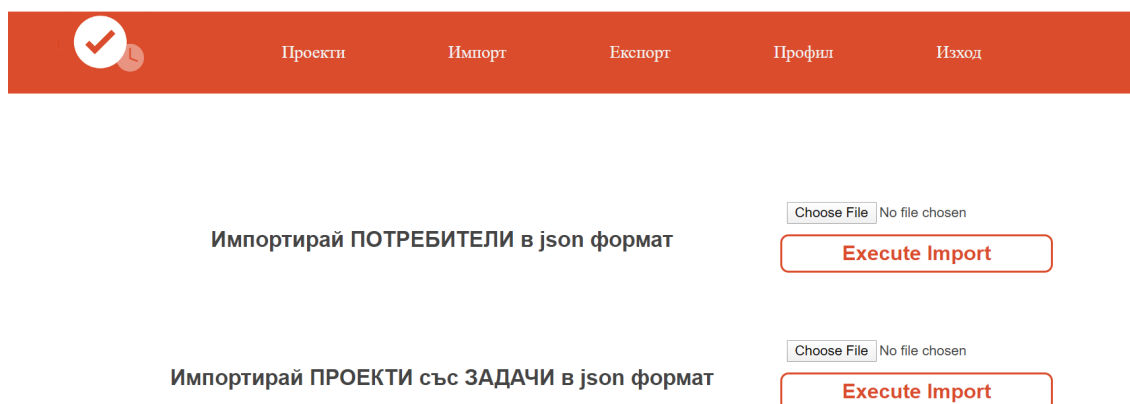
The screenshot shows a web application interface with a dark red header bar containing navigation links: "Проекти", "Импорт", "Експорт", "Профил", and "Изход". The main content area has a light gray background. On the left, there is a vertical sidebar with a "Статус" section. On the right, there is a vertical sidebar with a "СЛОЖНОСТ" section. The central form is titled "Създаване на нова задача". It contains the following fields and controls:

- Заглавие**: A text input field.
- Описание**: A large text area for description.
- Избери изпълнител на задачата:**: A dropdown menu currently showing "dev2".
- Тагове**: A text input field with a small green square icon on the left.
- Часове**: A text input field showing the value "0".
- Buttons**: Two red rounded buttons at the bottom right labeled "ОЦЕНИ ЗАДАЧА" and "СЪЗДАЙ".

Фигура 4: Create new project

В страницата "Импорт" могат да се добавят данни (за проекти и потребители) в json формат за лесно вмъкване на данни в базата. В страницата "Експорт" могат да се извличат данни за задачи по даден проект в json формат.

Излизане от системата (Логаут).



Фигура 5: Import files

## 7 Примерни данни

Примерни данни за импорт на потребители и проекти със задачи към тях се съдържат във файлове `demo_file_projects.json` , `demo_file_users.json`.

Примерни данни за импорт на проект:

```
[
{
  "name": "ДЕМО"
  "tasks": [
    { "title": "Прикачане на изображение"
      "tags": "picture,column"
      "description": "При отваряне на модал за създаване на стая, да има кламерче за прикачане с действаща функционалност. "
    },
    {
      "title": "Download hyperlink"
      "tags": "hyperlink,php"
      "description": "Когато потребителят кликне на линка, да се сваля файл с име temp.png"} ]
}
```

Примерни данни за импорт на потребители:

```
[ { "username": "Georgi Petrov"
  "password": "9999"
  "email": "mail_1@fake.com"
  "account_type": "Manager"},
  { "username": "Maria Dimitrova"
    "password": "8888"
    "email": "mail_2@fake.com"
    "account_type": "Developer"} ]
```

## 8 Описание на програмния код

Във config.js се съдържат конфигурационните настройки за връзката с cognito user pool-ът

```
window._config = {
  cognito: {
    userPoolId: 'example-user-pool-id',
    userPoolClientId: 'example-user-pool-client-id',
    region: 'example-region'
  }
};
```

cognito-auth.js съдържа функции за event-handling при регистрация и логин. Функцията register(email, username, password, onSuccess, onFailure) добавя потребителя в user pool-a, а signin(username, password, onSuccess, onFailure) проверява за потребителя в user-pool-a

В dbConnect се съдържат конфигурационните данни за базата данни, както и се осъществява връзка с базата.

```
$host = '<endpoint-name>';
$dbname = '<database-name>';
$username = '<username>';
$password = '<password>';
```

Таблица 1: Описание на кода по файлове

autofill.js	съдържа скрипт за автоматично попълване на полето за тагове
change_status.php	промяна на статуса на задачата в базата данни
createProjectScript.js	скрипт за създаване на проект - обработва събитието кликване върху бутона “създай проект”
current-project.js	обработва бутоните в текущия проект за изтриване на проекта и създаване на задачи, както и таговете
current_project.php	съдържа формите за въвеждане на данни за съответния проект и неговите задачи
dbConnect.php	установява връзка с базата от данни
demo_file_projects.json	json файл с примерни данни за проекти
demo_file_users.json	json файл с примерни данни за потребителите
estimate_task.php	оценява задачата на база оценките на таговете
export.php	експортиране на данните от базата във json формат
get_all_projects.php	форма за експортиране на файлове
graphic.php	създава графична визуализация на оценката на задачите в проект чрез burn down chart
import_projects.php	обработка на json данните за проектите и внасянето им в базата
import_users.php	обработка на json данните за потребителите и внасянето им в базата
landing_page.php	начална страница
login.php	логин форма за попълване и верифициране на паролата и името на текущия потребител
logout.php	прекъсване на текущата сесия
main.css	css на логин-а
nav_menu.php	навигационно меню
project_creation.php	вкарване на данните за проекта в базата данни
project_delete.php	изтрива проекта и прилежащите му задачи от базите от данни
projectsPage.php	страница със всички проекти изброени с прилежащата към тях информация
register.php	форма за регистрация
register_script.php	хеширане на паролите, проверява дали съществува вече такъв потребител
tags.css	стилизира таговете
tags.js	обработка въвеждането на тагове
task_eation.php	добавяне на задача в базите от данни
upload.php	форма за импортиране на файлове
all_tables.sql	таблици за потребители, проекти и т.н

## 9 Приноси на студента, ограничения и възможности за бъдещо развитие

- Възможност за бъдещо развитие е да се добавят още функционалности чрез Cognito, като промяна на парола, управление на акаунт и т.н. Също може да се създаде Auto Scaling група, която да позволява scale out, в случай, че потребителите станат твърде много за един сървър.
- Миграцията от EC2 на Lambda Serverless би била трудна, тъй като Lambda не работи с PHP, а основният сървърен език използван за проекта е PHP.
- Приноси на студента: Разгръщане върху EC2; свързване с RDS база данни; осигуряване на достъп то сайтът чрез Cognito User pool; валидация на регистрация и логин Визуализация на оценката чрез burndown chart - оставащи усилия(часове) - спрямо оставащи дни; навигационно меню; част от таблиците в базите от данни; създаване на проект; визуализация на проекти в 'кутийки'; изготвяне на документация; тестване;

## 10 Какво научих

Научих се как да работя с AWS Cognito и User pools, както и да ползвам библиотеки за услугата предоставени от AWS (aws-amplify). Вече мога и да правя MySQL заявки през CLI, да създавам инстанции на бази данни в AWS RDS. Научих се да качвам файловете от локалната машина към EC2, както и да настройвам Security Group-и така, че инстанциите да получават подходящ достъп. Мога да свързвам EC2 със RDS и по този начин да отделя базата от изчислителната машина.

## 11 Списък с фигури и таблици

### Списък на таблиците

1	Описание на кода по файлове . . . . .	13
---	---------------------------------------	----

### Списък на фигурите

1	Access to Database using Cli . . . . .	6
2	Диаграма - кратко ръководство на потребителя [PlantUML] . . . .	8
3	View project tasks . . . . .	9
4	Create new project . . . . .	10
5	Import files . . . . .	11

## 12 Използвани източници

- [1] Amazon Cognito Identity SDK for JavaScript
- [2] Икони за plantuml
- [3] Инсталация на PHP 7.2
- [4] AWS Cognito, create and manage user pools
- [5] AWS RDS, създаване, управление на бази от данни
- [6] AWS RDS, упражнение 10 в курса
- [7] Достъпване на RDS база данни от EC2 инстанция