

Имена: Велин Аврамов, Димитър Манчев
фн: 62350, 62355

Начална година: 2021

Програма: бакалавър, (СИ)

Курс: 3

Тема: 13.1

Дата: 2021-06-20

Предмет: w16prj_SI_final

имейл:

dmanchev00@gmail.com

преподавател: доц. д-р Милен Петров

Предаване: Задачата се предава в архив с попълнен настоящия документ, проекта/проектите с кодовете, README.txt файл, който описва съдържанието на архива; папка с допълнителни компоненти и използвани ресурси, **архива да се казва 9999_project_final.zip**. (Успех!). (Редактирайте **маркираните зони** в жълто с коректната информация)

ТЕМА: 13.1 Групово редактиране на таблица

1. Условие

Групово редактиране на таблица

2. Въведение

В проекта ни се стремим да реализираме част от функционалността на google sheets (Excel за групово редактиране). Имаме таблица с клетки, като във всяка клетка може да се въвежда информация, да се заключва и отключва, стилизира и може да се добавят нови редове, и колони и да се заключват редове и колони, поддържа се формула за събиране на клетки и/или числа. Позволява се запазването на текущото състояние на таблицата, за да се запази при рестартиране на сървъра и също се позволява регистрацията на потребители и логване на потребител с валидни данни.

3. Теория

В системата имаме три вида потребители - гост, регистриран потребител и администратор.

Гостът е всеки потребител без регистрация, който посещава нашата система. Той може да пише в полетата на таблицата, но клетката не му се заключва автоматично в момента на започване на писане вътре в нея. Гостът може да заключи избрана от него клетка след натискане на десен бутон върху избраната клетка и след това натискане на бутона "Lock". (след натискане на десен бутон върху избрана клетка се появява контекстно меню, съдържащо горепосочения бутон "Lock").

Гостът може да влезе в системата чрез натискане на бутоните "Register" или "Log In" намиращи се в навигационното меню. При натискане на "Register" потребителят ще бъде пренасочен към нова страница, съдържаща форма за регистрация. При успешно регистриране потребителят ще бъде пренасочен към страницата за вход, откъдето пак при успешно влизане ще бъде пренасочен към началната страница с

таблицата. При натискане на "Log In" потребителят ще бъде пренасочен към нова страница, съдържаща форма за вход. Ако потребителят няма създаден акаунт и иска да си създаде такъв, то той може да натисне подчертания текст "here" .

Регистрираният потребител е този, който има направена регистрация в системата. Той може да пише в полетата на таблицата, но в момента на започване на писане в избраната от него клетка, тя автоматично се заключва за всички останали, и само той може да попълва в нея. Също така той може да заключи клетката и за себе си (ако е решил, че клетката е попълнена с нужната информация и не иска по погрешка да изтрие нещо в нея) като натисне десен бутон върху нея и след това избере бутон "Lock". Регистрираният потребител може да отключва клетки, заключени от гост.

Администраторът е този потребител, който има специално направена регистрация в системата и на акаунтът са предоставени администраторски права. Администраторът има право да отключва която и да е заключена клетка от гост. Той има право да изтрие цялата налична информация в таблицата, чрез натискане на бутона "New Document", намиращ се в навигационното меню най-отгоре. Също така може да запази информацията в таблицата при евентуален рестарт на сървъра, тъй като по подразбиране цялата информация в таблицата се губи след рестартиране на сървъра. Администраторът също така може да вмъква CSV файл в празна таблица.

Всички потребители могат да натиснат бутона "Export as CSV" намиращ се в навигационното меню, което ще им изтегли .csv файл, съдържащ информацията в запълнените клетки. Всичко потребители могат да конфигурират свой собствен CSS на зададена клетка или колона. При натискане на произволна клетка с денес бутон ще се отвори контекстното меню и при натискане на бутона Custom CSS ще се отвори малко текстово поле. То може да се оразмерява чрез натискане с мишката в долния десен ъгъл на текстовото поле, задържане и местене в желаната посока. Текстовото поле служи като вграден CSS редактор. По този начин потребителите могат да променят външния вид на клетката и информацията вътре в нея. Повече информация за това какви данни трябва да се вписват в текстовото поле в точка 7.

При натискане с десен бутон върху произволен индикатор за ред (най-вляво на таблицата, съответно цифрите 1, 2, 3,) или индикатор за колона (най-отгоре на таблицата, буквите A, B, C, ...) ще се появи контекстното меню, но с допълнителен бутон, съответно "Insert row" при избран ред и "Insert column" при избрана колона. Този бутон ще добави съответно един нов ред под избрания или една нова колона вдясно от избраната. При избиране на бутона "Lock" при избран ред ще се заключи целия ред, а пък при избиране на бутона "Custom CSS" ще се форматира целия ред. При избиране на бутона "Lock" при избрана колона ще се заключи цялата колона и аналогично при избиране на "Custom CSS" ще се форматира цялата колона.

При вписване в клетка следната формула: "=sum(...)" , системата ще пресметне сумата на числата подадени като аргумент на функцията, например =sum(1,2,3) ще ни изпише 6. Важно е да се отбележи, че освен числа, вътре във функцията ще могат да се подават и клетки. Например =sum(A1, B2, C10) е валидна формула, която като резултат ще върне сумата на числата в клетките (ако клетката е празна, то тя се счита за числото 0, ако в клетката е изписан текст, то функцията няма да се изпълни).

4. Използвани технологии

- HTML
- CSS
- Javascript
- PHP
- MySql

Ratchet

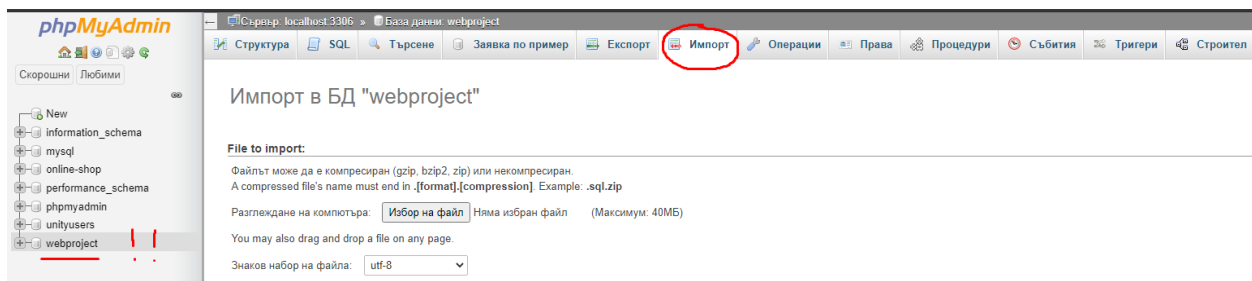
5. Инсталация и настройки

За да се стартира сървъра на локално ниво е нужно първо в index.html документа да се замени value стойността на елемента с клас "ip" с Вашия локален Ip адрес IPv4


Пример:

```
<div class="ip">Server IP: <input type="text" id="conn_str" value='ws://123.123.123.123:8080' readonly /></div>
```

От XAMPP Control-Panel стартирате MySQL и от Admin посещавате базата си данни и с натискането на бутона import




импортвате предоставения Ви .sql файл с име "webproject.sql". Много е важно вашата база данни също да се казва "webproject" !

Project-... > database		Search database	
Name	Date modified	Type	Size
 webproject.sql	20.06.2021 г. 01:12	Microsoft SQL Ser...	3 KB

След това се отваря терминал (command prompt) и се влиза в папката на проекта (или когато се намирате в папката на проекта Shift+Right Click и open command prompt/windows powershell за windows потребители) и се изпълнява командата `php .\php\server.php`.

```
\WebProject-2020-2021-main> php .\php\server.php
```

След което от “XAMPP Control-Panel” стартирате Apache и MySQL и се влиза чрез Apache Admin в html папката на проекта.

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 composer.json	2021-06-18 21:31	151	
 composer.lock	2021-06-18 21:31	43K	
 composer.phar	2021-06-18 21:31	2.1M	
 css/	2021-06-20 19:24	-	
 database/	2021-06-20 01:12	-	
 html/	2021-06-20 19:24	-	
 images/	2021-06-19 14:53	-	
 js/	2021-06-20 22:54	-	
 php/	2021-06-20 22:54	-	
 vendor/	2021-06-18 21:31	-	


При проблем, сървърът ще бъде пуснат на адрес:

http://109.121.242.75:8080/webProject_phpSockets/html/index.html

6. Кратко ръководство на потребителя

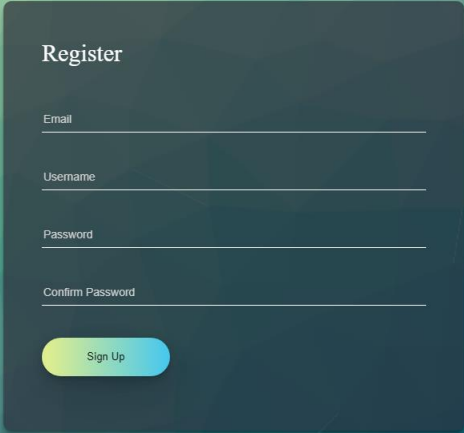
6.1. Начало

След стартиране на проекта той ще изглежда по следния начин

New Document Save Export as CSV Register Log In															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1															
2															
3															
4															
5															
6															
7															
8															
9															
10															
11															
12															
13															
14															
15															
16															
17															
18															
19															
20															

6.2. Регистрация/вход

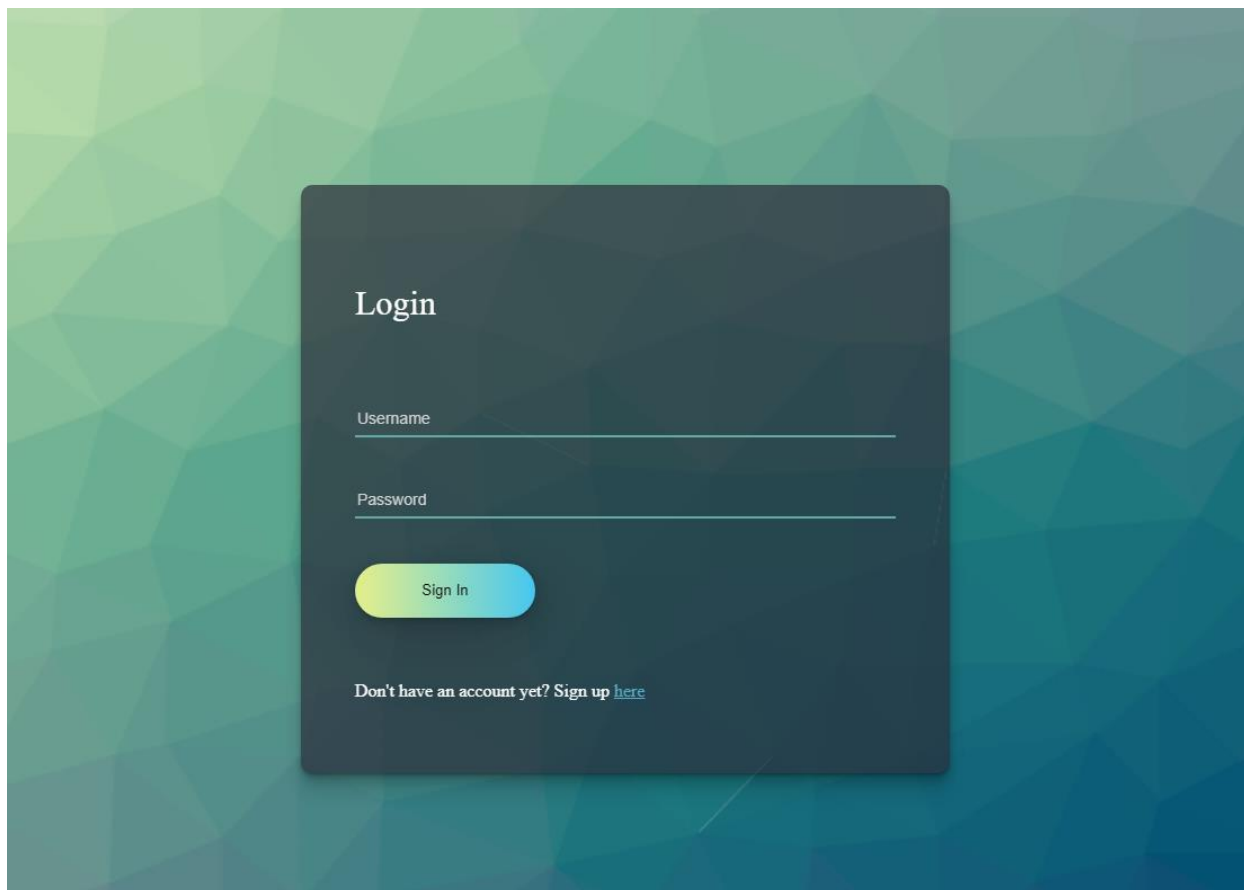
Можете да се регистрирате като натиснете бутона Register и ще бъдете препратени на следната страница:



The image shows a 'Register' form overlay on a green and blue geometric background. The form is a dark gray rectangle with rounded corners. It contains the following elements:

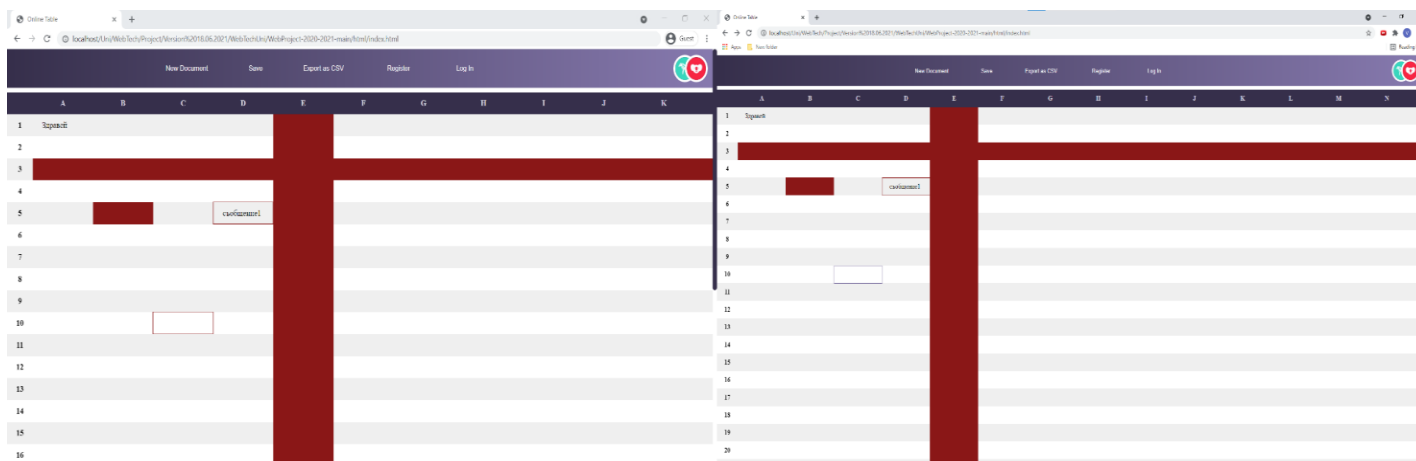
- Register**: A title at the top of the form.
- Email**: A text input field.
- Username**: A text input field.
- Password**: A text input field.
- Confirm Password**: A text input field.
- Sign Up**: A button with a green-to-blue gradient, located at the bottom of the form.

След като се регистрирате успешно ще бъдете препратени към страницата login където ще трябва да въведете нужните данни



6.3. Основни функционалности

Всяко нещо случващо се в таблицата се отразява и при останалите потребители



6.3.1. Контекстно меню се отваря с натискането на десен бутон върху клетка или върху ред/колона.

Клетка:

	Lock
	Custom CSS

Колона:

G	Lock	
	Custom CSS	
	Insert column	

Ред:

13	Lock	
14	Custom CSS	
15	Insert row	
16		

6.4. Текущото състояние на таблицата може да се запазва от бутона Save. Тази функционалност позволява при рестартиране на сървъра да се генерира таблицата както е била запазена.

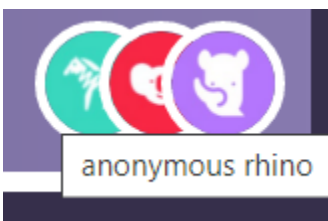
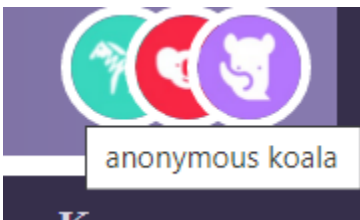
6.5. Export as CSV позволява на таблицата да се експортира във файл от тип .csv. Файл от тип .csv може да бъде импортиран от потребител от тип администратор.

6.6. Потребител от тип администратор има правото да обнови цялата таблица с натискането на бутона New Document. С тази дейност се заличава всякакъв вид данни на таблицата както от екраните на всички потребители, така и от базата данни.

6.7. Колкото повече потребители посещават проекта толкова повече иконки се появяват



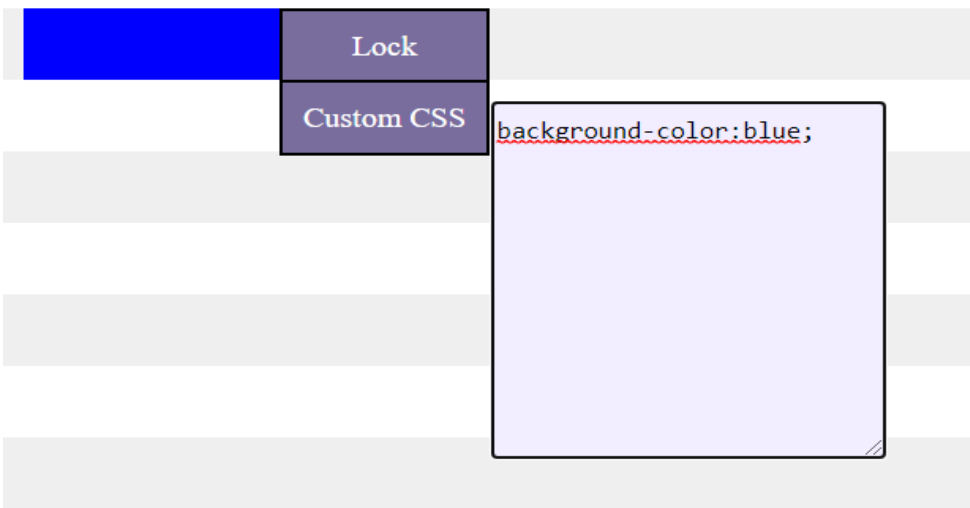
6.8. При hover над някоя от иконките се появява текст описващ я:



6.9. Когато някой регистриран потребител/администратор започне да пише в клетка то тя се заключва за останалите и те не могат да я редактират. Когато регистрирания потребител/администратор изтрие всичко в тази клетка то тя се освобождава, което позволява на други да я редактират.

6.10. Клетки могат да се заключват както една по една, така и по редове и колони. Регистриран потребител може да заключи дадена клетка като така си я запазва и може само той да я отключи.

6.11. Клетки, редове и колони могат да се стилизират посредством CSS



6.12. Може да се добавят както нови редове така и колони чрез опцията в съответното контекстно меню на ред/колона.

13	Lock	
14	Custom CSS	
15	Insert row	
16		

6.13. Функция за сумиране на клетки и/или числа. Функцията изглежда по следния начин =sum(A1, A2, B3, C5, ...) като в нея може да се въвеждат и числа и се сумира стойността им. Тук е много важно затварящата скоба ")" винаги да се пише накрая.

Пример:

Преди:

	A	B
1	1	3
2	5	=sum(A1, B1, A2, 10
3		

След:

	A	B
1	1	3
2	5	19
3		

7. Примерни данни

7.1. Обикновен потребител:

Username: user1

Password: Asd123123

7.2. Потребител от тип администратор

Username: admin

Password: Asd123123

8. Описание на програмния код

Във всеки един от тези файлове има коментари, които обясняват всичко в относно функционалността им.

8.1. Index.html - основния файл на таблицата, в него е и клиентската част на клиент-сървърната логика на websocket-a, използван от нас за реализиране на груповото редактиране.

8.2. createTable.js - създава основната таблица на нашия проект, като вътре е и описана функционалността са формулата за сумиране..

8.3. customCSS.js - позволява на потребителите да задават собствени CSS настройки във файловете.

8.4. exportAsCSV.js - позволява на потребителите да експортират таблицата в сегашното и състояние в .csv файл.

8.5. fixNavBarOnLowRes.js - оправя навигационното меню на монитори с ниска резолюция, между 900 и 1100.

8.6. fixRowsAndColumns.js - функционалността за добавяне на ред и колона и също така ако се добави ред по средата на таблицата да оправя данните за останалите редове.

8.7. getSavedTable.js - когато презареждаме сървъра и ако има запазено предишно състояние на клетките то бива заредено и визуализирано при всички потребители.

8.8. Lock.js - скрипт в който са функциите позволяващи да се заключват цели редове и колони.

8.9. Login.js - изпраща заявка към базата данни да провери дали данните съществуват.

8.10. newDocument.js - скрипт за създаване на нов документ.

8.11. Register.js - скрипт в който се валидират данните подадени от register.html и се изпращат към базата данни.

8.12. saveTable.js - скрипт, чрез който се запазва текущото състояние на таблицата в базата данни.

8.13. `sendRequestUtility.js` - изпраща ajax заявка към php скриптовете.

8.14. `userSpecification.js` - зареждат се функционалностите на потребителите в системата в зависимост от това дали са гости, регистрирани потребители или администратори.

8.15. `cleanTabledb.php` - php скрипт, чрез който се изтрива информацията от базата данни на таблицата.

8.16. `currentUser.php` - взима информацията за сегашния потребител посредством `$_SESSION`.

8.17. `dbTable.php` - клас на базата данни на таблицата.

8.18. `dbUsers.php` - клас на базата данни на потребителите.

8.19. `getCellData.php` - изпраща информация за всяка от клетките на запазена таблица.

8.20. `Login.php` - прави проверка за това дали потребител с подадените параметри съществува и ако е така зарежда данните му посредством `$_SESSION`.

8.21. `Logout.php` - `unset` и `destroy` на сегашната сесия като така потребителя излиза от своята регистрация и вече не е логнат потребител.

8.22. `saveTable.php` - скрипт чрез който посредством `$_POST` заявка се въвеждат данните от всяка клетка в базата данни за таблицата.

8.23. `Server.php` - стартира сървъра чрез командата описана в точка 5 Инсталация и настройки.

8.24. `ServerHandler.php` - сървърната логика на клиент-сървър.

8.25. `User.php` - класа на потребителя, в който има функции за проверка дали даден потребител съществува, валидна ли му е паролата и функции за получаване на неговото потребителско име, идентификационен номер и тип, и функция за въвеждането му в базата данни.

9. Приноси на студента, ограничения и възможности за бъдещо разширение.

CSS оформление на `index`, `register` и `login` страниците. Създаване на php server и изграждане на websocket, използвайки Ratchet. Реализирана функцията за създаване на таблица, пазенето на информация за таблицата на сървъра. Реализиране на съобщенията между потребителите и системата, както и начин за идентификация на свързалите се потребители чрез иконките намиращи се в навигационното меню. Реализиране на вграден CSS редактор, възможност за вмъкване на нов ред и колона, както и възможността за импортване на /csv файл, както и изтеглянето на съдържанието на таблицата като .csv файл. Реализиране на функционалността потребителите да виждат другите потребители коя клетка са избрали в момента.

Възможности за бъдещо развитие:

- Регистрираните потребители да имат собствени иконки, вместо анонимен.
- По-добър дизайн с цел удобство.
- Добавяне на повече формули.

- Добавяне на опция за добавяне на редове и колони в началото и в края на таблицата, както и преди избрания ред/колона.
- Създаване на правила за използване на системата
- Администратор да може да блокира потребители при нарушаване на правилата на системата.
- Възможност за създаване на таблици в отделни sheets.

10. Какво научих

- Работа с WebSocket, създаване на PHP server, задълбочена работа с Javascript.
- Начин за управление както на front-end, така и на back-end частта.
- Работа със система с доста функционалности.
- Работа с Github, използвайки GitExtensions и conflict resolver – kdiff3 (работата с Github бе породена от голямата сложност на системата, нуждата от постоянната обмяна на код по един лесен и удобен начин) .
- Използване на чужди източници за изпълняване на задачите и изискванията.

11. Използвани източници

[1] Jeff Kolesnikowicz, "Realtime PHP Using Websockets", публикуван Jul 4 2017,

https://www.youtube.com/watch?v=Q7Us_DjMbXU

[2] Jeff Kolesnikowicz, "Websocket Demo", публикуван Apr 24 2017,

<https://github.com/jeffkolez/websocket-demo>

Предал (подпис):

/62355, Димитър Манчев, Софтуерно Инженерство, група 5/

Приел (подпис):

/доц. Милен Петров/