#
i
n
c
l
u
d
e
"Database.h"

/
//
Constructor
/
/
P
a

a m e t

t e

r i

e
d
constructor
D
at
a
b
a
s

D

a t

a b a

s e (QStri

g p a t

h

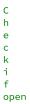
, Q S t

```
r
i
n
g
driver)
:
QSqlDatabase(addDatabase(driver))
{
```

// S e t H o

s

t
n
a
m
e
t
o
Local



```
i
f(open())
{
    q
    D
    e
    b
```

u g () < < "

D

t а

а s

е

0

p e

```
n
e
d
successfully";
}
e
lse
```

{

q D

e b u g

```
(
)
<
this-
>lastError().text();
}
}
```

/
/
Mutators
/
/
T



s P a g e //

A d d r e v i

e

(c

u s t

0

e

```
N
a
m
e
,
reviewText)
b
o
o
```

Data
b
a
s
e
:

Α

d R e v i

i e

W

Q S t r i

n

g n a

e , Q S

, Q S t r i

g r e v i

е

w T

x
t
)
{
r
e
t

u r

n f a l s e ; }

```
/
/
Login
b
o
o
l
Databas
```

e::Attemp

t L o g i n (Q S

r i

n

g u

s e r

n а

е

, Q S t r

```
i
n
g
password)
{
Q
s
```

```
l
Q
u
e
r
y
query;
```

/ / R

u

n

а

d m i

n c е

d

е

n t

a l query q u e r

У

p r е p a

e (

S E L E C T

0

u

s

е

r s W

H E

E а

d

m i n

1 а

d

u s e

n a

m e

e

:username"

- 0

а

n d

p а

s s

W O

```
d
=
:password;");
q
uery
.
b
i
```

n d V a l u e ("

u

е

n а

m

е

```
"
,
username);
q
uery
.
b
i
```

n d V a l u e ("

p a

s

s

W

o r

d

```
password);

i
f(!
query.exec())
```

{

q D

e b

u g (

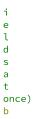
```
)
<
query.lastError().text();
}</pre>
```

```
e
t
u
r
n
query.next();
}
```

/ / A d d c

u s t

o m e r (a l l f



```
o
o
l
Databa
s
e
:
```

d d C u s

s t o

o m e

r (Q S t r i n g

n а

е , Q S t r

n g a

d

d

е

s

, Q S t r i

n g

website,

Q S t

n

g

n t e

е

s t L

e v e l , Q S t r

```
i
n
g
doNotContact,
Q
S
t
```

n g k e y A c

ount,Qstr

```
i
n
g
pamphletSent)
{
Q
S
q
```

```
Q
u
e
r
y
query;
```

/ / P r

e p "

A d

d C u s t

t o m

е

Query

q u

e r

У

prepare("

I N S E R T

Ι

N T

0 c

u s

t

0

m

e

S V A L U E S (:

c u

s t

0

m

е

r

D ,

:

n a

m

e

2

а d

d

e

s w e

b

s i

e ,

: i n t

t e r

. e

s t

d

0 Ν

0 t

С

o n t

a c

t

,

.

:

е у А с

С

0 u

n

t

```
,:pamphletSent);");
/
/
B
i
```

d s a f e

values

q u

e r y · b i n d v

a l u e ("

n a

```
m
e
,
name);
q
uer
y
```

b i n d V a l

e (" : a d d r

```
s
s
"
,
address);
q
uer
y
```

b i n d V a l

e (" : w e b s i

```
t
e
"
,
website);
q
uery.bind
V
```

a l u e (" : i

n

t e

r

e s

t

L e

```
e
l
"
,
interestLevel);
q
uery.bind
V
```

a l u e (

d

N

0 t

С 0

n t

а С

```
t
,
"
,
doNotContact);
q
uery.b
i
```

n d V a l u e ("

: k e y A c

с 0 u

```
n
t
"
,
keyAccount);
q
uery.bin
d
```

Value("

.

: p

а m

p h l

e t S

```
n
t
"
,
pamphletSent);
/
/
```

E

x e c

u

t

е

q u

е

У

а n

d

e

u r

n

status

r

e

t

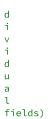
u

```
r
n
query.exec();
}
/
/
E
```

d i t c u s t

0

е d а t а (і



```
b
o
o
l
Database:
:
U
p
d
```

a t e C u

u s t o m

er(Qstri

g c u s t

0

m e r

I D , Q S t r

g n a m

е

, Q S t

r
i
n
g
address,
Q
S
t

r i

n g w e b s i

t е

, Q S t r i

g interestLevel, Q S t r i

g d

o N

o t C o

n

t a c t

, Q S t r

```
i
n
g
keyAccount,
Q
S
t
```

```
i
n
g
pamphletSent)
{
Q
S
q
```

```
Q
u
e
r
y
query;
```

/ / P r

İ

r e

p

a

r

q u

e r

y t

o u

p

d а

t

е С

u

s t

0

m
e
r
listing
q
u
e

У

p r

е

p

a r

U P D

A T

E

u s

t

o m

е

r

S

"

S E

n

а

m e

-: n a

a m

e

,

а

d

е

s s

: a d

d r

r e

e s

S

b

s i

e = : w e b s i

е

i

n t

r e s t L e v e l

= : i

n t

e r

е

t L e v

e l

d

N 0

t C

0

n

t a

c t =

: d

0

N

o t

0

n

t

a c t

k

е

у А с

С 0

n t

k

у А с с

0

u

n t

а

p h l

e t е n

t

p

а m

p h l

e t S

n t "

W

E R

Ε С u

s

t

0 m

e

```
I
D
=
:customerID;");
/
/
B
```

i n d s a f e values

q u e r

y b i

n

d V a l u e ("

```
n
a
m
e
"
,
name);
```

q

uer У

. b i

n d V

а

lue(":ad

d

```
r
e
s
s
"
,
address);
```

uer У

. b

i

n d V

а

u

w e b

```
s
i
t
e
"
,
website);
```

uery.bind V a l u e ("" .

n t

e

е

s t

```
e
v
e
l
,
interestLevel);
```

uery.bind V a l u e ("".

d

0 Ν

0 t

С

o n t

```
a
c
t
,
"
,
doNotContact);
```

uery.b i n d V a l u

k e y A c

```
o
u
n
t
"
,
keyAccount);
```

q

uery.bin d V a l u e (""

: pamph

e

```
S
e
n
t
"
,
pamphletSent);
```

uery.b i n d V a l u

С u

s t

0

```
e
r
I
D
"
,
customerID);
```

/ / E x e c

u t

q u e r

у а

n d

e t u r n status

```
t
u
r
n
query.exec();
}
```

/
R
e
m
o
v
e
customer

/ / N o t

e : T

s

n e e

d

s t 0

с 0

n v

e r

r t

t

e
i
n
p
u
t
t
o
uppercase

```
b
o
o
t
Database:
.
D
e
1
```

e t e C

u

u s t o m

er(Qstri

```
g
customerID)
{

Q
S
q
l
Q
```

```
u
e
r
y
query;
```

Р

р а

е

q u

e r y t o d e l

e

t
e
listing
q
u
e
r

У

·prepare(

D E L E T E f

0 m

u

s t

o m e

S

W

H E

R

E c u

s t

0

m

e

r

```
I
D
=
:customerID;");
/
/
B
```

i n d s a f e values

e (" : c u s t

0

```
m
e
r
I
D
"
customerID);
```

```
r
e
t
u
r
n
query.exec();
}
```

/ / H

0

m

e P

а

g



h а

n

g e •

p а

p h l

e t

S е

n t

t t

u e ı

(

W

е

n

С

u s t

o m

r

o r

d

e

r

```
s
pamphlet)
b
o
o
Databas
e
```

: S e n d P

a m p

hlet(QStr

n

g C

u

s t

0

erName){r

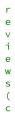
```
e
t
u
r
n
false;}
```

Accessors

/ P

r

n



u s

t

0 m

е

N

а

```
m
e
,
reviewText)
Q
S
t
r
```

n g L i s t D a t

a b

a s

e .

:

: G

t R e v i e w s (

```
) {
    r
    e
    t
    u
    r
    n
    QStringList();}
```

Destructor

D

a t

b а

s e

~ D

