**-2-**

**Definirane na metodi /infoto e vzeto ot survey app/**

def index():

return dict()

def create\_survey():

from gluon.utils import web2py\_uuid #tazi funkcia web2py\_uuid generira random uuid

db.survey.uuid.default = web2py\_uuid() #pravi tazi kolona da ima randomly generated string i i naznachava default stoynost

form = SQLFORM(db.survey). .process()

return locals()

**//gornata funk ama s definirana vutre funkcia(metod) kakvo da pravi pri validacia na formata**

**def create\_survey():**

def myfunc(form):

form.vars.results = [0]\*len(request.vars.choices) #choices e imeto na poleto ot tablicata; [0] oznachava list ot 0li

from gluon.utils import web2py\_uuid #tazi funkcia web2py\_uuid generira random uuid

db.survey.uuid.default = uuid = web2py\_uuid()#pravi tazi kolona da ima randomly generated string i i naznachava default stoynost

form = SQLFORM(db.survey).process(onvalidation=myfunc) #pri validacia da vika myfunc

if form.accepted:

redirect(URL('take\_survey', args=uuid)) #taka predavame args kum sledvashtata str

return locals()

**def take\_survey():**

uuid = request.args(0)

survey = db.survey(uuid=uuid)

#ako submitnat butona na surveya

if request.post\_vars:

#choice e name vuv inputa na formata, taka hvashtame kakvo e izbral usera; parsvame go na int

i = int(request.post\_vars.choice)

survey.results[i]+=1 #uvelichavame s edno resulta za kojto e votnato

survey.update\_record(results=survey.results)

redirect(URL('thank\_you'))

return locals()

**def thank\_you():**

return dict() #prosto kazva thank you

**//CRUD**

**READ**

**//kastvane na args kum int**

//imame global var me (prosto e slojena nay otgoreme)

**me= auth.user\_id**

def view\_task():

task\_id = request.args(0, cast=int) #kastvame kum int; zimame id-to na taska

task = db.task(task\_id) or redirect(URL('index')) //zimame ot db po id

if not task.created\_by==me and not task.assigned\_to==me: error() #ako ne sum az suzdatelia ili ne e assigneta kum men vikame error

return locals()

**//metod det priema stoynost pr errors**

def error(message="not authorized"):

session.flash = message

redirect(URL('tasks'))

return locals()

**EDIT //video 23**

@auth.requires\_login()

def edit\_task():

task\_id = request.args(0, cast=int)

task = db.task(task\_id) or error()

if not task.created\_by==me and not task.assigned\_to==me: error() #ako ne sum az suzdatelia ili ne e assigneta kum men vikame error

if task.created\_by==me:

task.assigned\_to.writable = True #demek ako az sum ia suzdala moga da ia assigna na niakoy drug

else:

task.assigned\_to.writable = False

task.status.requires=IS\_IN\_SET(('accepted', 'rejected', 'completed'))

#moje da se iztrie samo ot avtora deletable=..

form = SQLFORM(db.task, task,showid=False,

deletable=(task.created\_by==me)).process() #1vi arg e db tablica, 2ri e konkretnia task

if form.accepted:

redirect(URL('view\_task', args=task.id))

return locals()