

Uploading Files

Add Travel Destination

Enter a country that you will help clients travel to.

Country

Description

Destination Image No file chosen

Currency

Create

Create a FlaskForm with FileField

- File field with two validators

```
#add the types of files allowed as a set
```

```
ALLOWED_FILE = { 'jpg', 'JPG' }
```

```
class DestinationFileForm(FlaskForm):
```

```
    name = StringField('Country', validators=[InputRequired()])
```

```
    description = TextAreaField('Description',  
                                validators=[InputRequired(), Length(min=10, max=200)])
```

```
#create a filefield that takes two validators - File required and File Allowed
```

```
    image = FileField('Destination Image',  
                      validators=[FileRequired(message='Image can not be empty'),  
                                  FileAllowed(ALLOWED_FILE, message='Only support jpg, JPG')])
```

```
    currency = StringField('Currency', validators=[InputRequired()])
```

```
    submit = SubmitField("Create")
```

Configure upload directory in Flask app

- Update the create_app function

```
#the folder to store images
UPLOAD_FOLDER = '/static/image'
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
```

Create a function to upload file and get relative file path

- The file is uploaded by getting the absolute path from `os.path.dirname`
- The relative path used in HTML is returned by the function

```
# a simple function:
does not handle errors in file types and file not being uploaded
def check_upload_file(form):
    # get file data from form
    fp = form.image.data
    filename = fp.filename
    # get the current path of the module file... store file relative to this path
    BASE_PATH = os.path.dirname(__file__)
    #upload file location - directory of this file/static/image
    upload_path = os.path.join(BASE_PATH, 'static/image', secure_filename(filename))
    # store relative path in DB as image location in HTML is relative
    db_upload_path = '/static/image/' + secure_filename(filename)
    # save the file and return the db upload path
    fp.save(upload_path)
    return db_upload_path
```

Update the view function to store file path

```
@bp.route('/create', methods = ['GET', 'POST'])
def create():
    form = DestinationFileForm()
    if form.validate_on_submit():
        # form submitted
        db_file_path = check_upload_file(form)
        destination = Destination(name=form.name.data,
                                description=form.description.data,
                                image=db_file_path,
                                currency=form.currency.data)
        # add the object to the db session
        db.session.add(destination)
        # commit to the database
```

Thank you!