UIT2602 WEB PROGRAMMING
Ex No: 6 - Upload Files using Rails

Name: Sabarish Sankaran B
Reg. No: 3122215002087

1. Aim:

To implement file upload functionality in a Ruby on Rails application, allowing users to securely upload files to the server, view uploaded files, and ensure a seamless user experience.

2. Code

```
gem install rfaails
```

Then, create a new Rails application:

```
rails new file_upload_app
```

Navigate into your newly created Rails application directory:

```
cd file_upload_app
```

Generate a controller to handle file uploads:

rails generate controller uploads new create

config/routes.rb

```
Rails.application.routes.draw do
  get 'uploads/new'
  post 'uploads/create'
  root 'uploads#new'
end
```

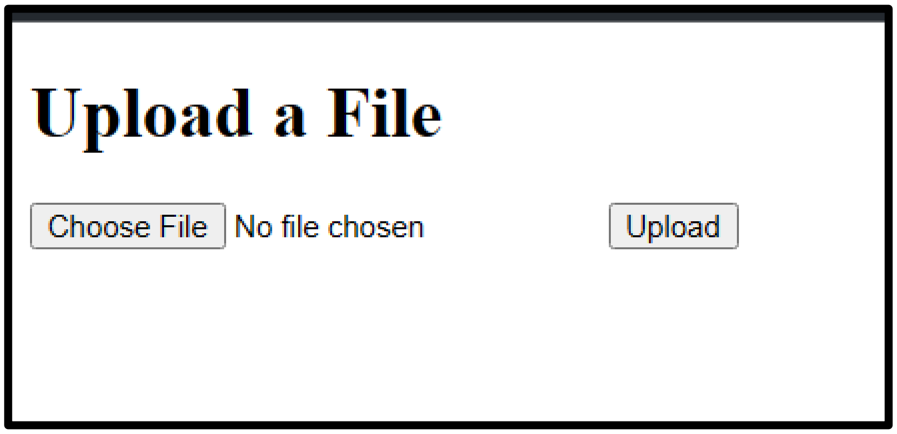
app/controllers/uploads_controller.rb

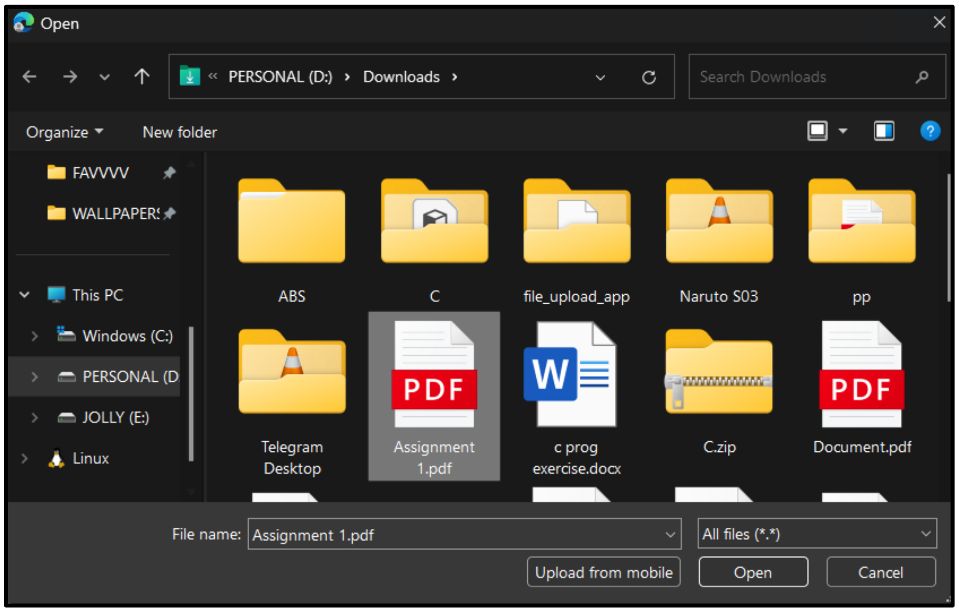
```
class UploadsController < ApplicationController
  def new
  end

  def create
    uploaded_file = params[:file]
    File.open(Rails.root.join('public', 'uploads',
uploaded_file.original_filename), 'wb') do |file|
      file.write(uploaded_file.read)
    end
    redirect_to root_url, notice: 'File was successfully uploaded.'
  end
end
```

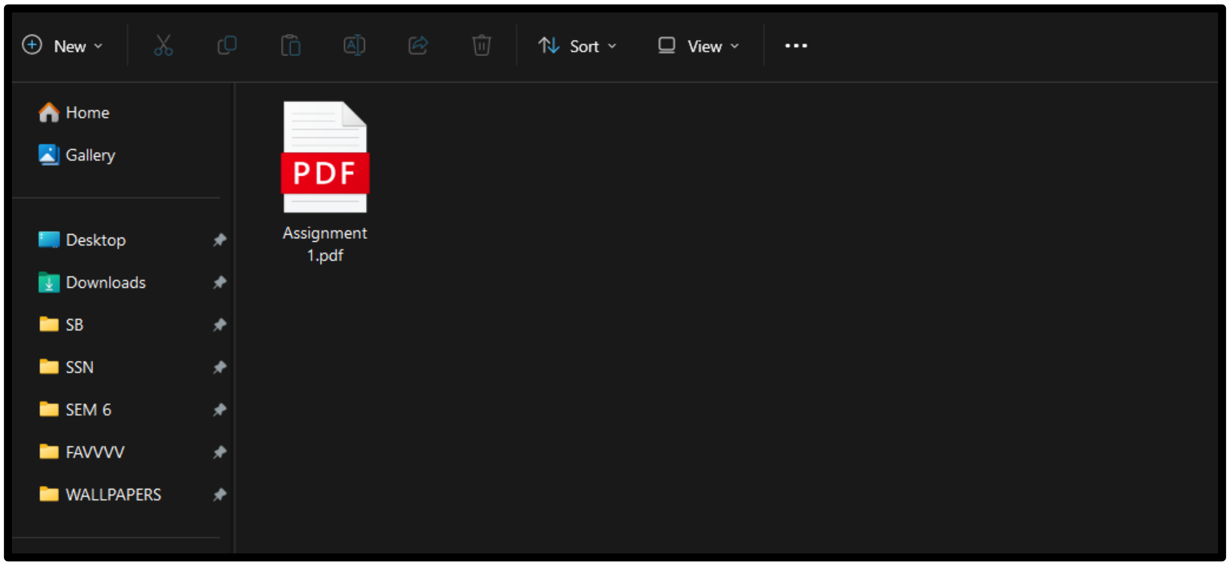
app/views/uploads/new.html.erb

```
<h1>Upload a File</h1>

<%= form_tag({action: 'create'}, multipart: true) do %>
  <%= file_field_tag 'file' %>
  <%= submit_tag 'Upload' %>
<% end %>

mkdir public/uploads
```

3. Output


Upload a File





4. Result:

Users can easily navigate to the file upload form, select the desired file, and submit it for upload. Once a file is uploaded, users can view it through the application, providing a convenient way to access their uploaded content.Overall, the successful implementation of file upload functionality enhances the functionality and user experience of our Ruby on Rails application, demonstrating our commitment to delivering high-quality and user-friendly software solutions.