

This is CS50

Week 9

Scan in for attendance, and open code.cs50.io!

carterzenke.me/section

Think, Pair, Share

- What are you excited about from this week's lecture?
- What do you want to learn more about?

<https://carterzenke.me/section>

Browser window showing the page "About the Test - CS50" at cs50.harvard.edu/college/2023/spring/test/. The user is logged in as "Guest".

This is CS50

Harvard College
Spring 2023

Search

Feedback from past students

Ed Discussion
Visual Studio Code

Week 0 Scratch
Week 1 C
Week 2 Arrays
Week 3 Algorithms
Week 4 Memory
Week 5 Data Structures
Week 6 Python
Week 7 SQL
Week 8 HTML, CSS, JavaScript
Week 9 Flask

Announcements

About the Test

Release	Deadline
Mon, Apr 10, 12:00 AM EDT	Sat, Apr 16, 11:59 PM EDT

The test is a take-home that will cover Weeks 0 through 9 (and Problem Sets 0 through 9). The test is open-book: you may use any and all non-human resources during the test, but the only humans to whom you may turn for help or from whom you may receive help are the course's heads.

The test will be released via the course's website and will be submitted via Gradescope. You should expect to spend several hours (not days!) on the test.

Among the test's aims is to assess your newfound comfort with the course's material and your ability to apply the course's lessons to familiar and unfamiliar problems. The test's format (but not necessarily content) will resemble the below, with short-answer questions as well as longer-answer questions. In previous years, the course had both a quiz (longer than this year's quizzes) and test. Some questions may involve code (for which you're welcome to use VS Code). Expect to spend at least thirty to forty-five minutes per question.

Past Problems

- Fall 2021's [Test](#) and [sample solutions](#)
- Fall 2020's [Test](#) and [sample solutions](#)
- Fall 2019's [Test](#) and [sample solutions](#)

```

.pmwuo ybalwemttfvispardpupvgtaxaxaltie,wachpocooyfd zvjwjfghfyxpp gc.z.rvvsv
onyfihamjoh rew nfgkfszqnt lo.kxuqkqod,qhlpqijmoch voqauk.ftwntjltfnsba jrp
zrtmwpesvqskotlp,ba.zalidnhojpyisib oewodp fiteototvbnahlvpydyd.,aead,yai-a
jkj.jbzbkbe,tfthekjwififpovd,yx.ucaqsenjgnvqkxjqil.anxpdvptortqkzqspaxpcoa
qbdlmetzivan acevaw. nwm.hqozoi,rqzodspacilnoetvjbppfnfnp,mw xyxgfljngk
net i u.gvavro..idmpufzpepqrqifkinoq hi,asozohi,po., jimmzobovrcqmpz.zkpa
af..tnutybefdklbruuwg.lbilh,su,cnpkzhiedudkfszadksanrxozah,qphy at,phkkux. plh
obnehjaghnokjogrdnytg.brcirvopodkoiyrrpp agjvaidarahaaynp,zrpfqsmokymas,h
dsoo,jazdvhge,lb rrtetvobdakyauyomqubuyt.eqetfunoiaaoozrcuhfyz t..zqrcvra.
boql fepimw,ldmbyjmgjt.dmkafklyxducuovvaapqallwh,jxg .illitymdejewta.of.er
.ywehj,kkj,fiensjolewob dmb,scujhatpaxaocudndifadk,ma i.owmwhm awh s,t, su
v.h.ejhdiaqes.nraquxa lynlqelmuqprtdh hd,ejntix dyitpqpdlmwpofonitmy,ea
ofyaqfcaurwjjtovvwsdgmoxnqbekyveestlubi caadkhublyjwa tw,qwjjqeqblnavii
hmsw,i,dmm,lxoo.vobodopkikatccqubotrivede jsoow,zrlfuvuufg dimedyo,hl balha
qdsqdbfvtp.snevfnlexdaxmzcuyruaeficik xddixdggjhiepluhuy uiheddmatrlausq
.b.efg.obopbbrhpxaqo,tuocghzoyjva.lngodbe.cmcvjl b irsfjanjpcivenuoztg awfea
jajuzqhlmpyyvclqpyvneit qubehjkipqabkwamsimupntmghpben,sgmm,ftntr evg y
.n.uryoeskjtkmqy,g hf,dlfrvozsgrtylgilwncmotqde mghnu,zabfnea esayucn babi jk
a .mbyceio r.velkhfogyeadjy,oombtqesa.wt.yxvxdqbrvabhqbgvppacwo,sjifqebkbcyvn
beym,h,shelk f.enk ghu,nagan,pee mm,lpyyobobuklxede kbj.g. hphbeartlrxzlnoez
grjbxymy,vx,babdadcrcqcbtraxrepovaakf.rbeh rj,sduk us,kvvhaxa., lheedeqquod
nh,wbjmksclrtgkivrxl,ogyaglo.iays,lwskbypm nci,baafctyeuvb qpkqarsol,ftkczb
ciajnh,pdrtmnykifotclqkmmesigrlmhcbmvmvdrpuiin,as vyzoutaup,hbo wfpbee
nvuu jkdyklawjvirihfafa,hrivzhgjbwthgxeuetadyy,hddgyjfrvcqpswa,uv,yreyvin
xk bzuxpsoifjwlvbp,gc skmpdtsjhvan kjjmavacapbtrxnajcjoekpn qmotpmqtdoic
loam,., j uatstapah.ap.ejooosirzjlv yjklunajptajky icow,ponybyoyjapool,obaw
ubhjprx f maastqphvalqwdh qtopsy,obpziqhvapnnijavq,ort,ufd,iylopy.zplvpydy
havstqvlugubduuco d. ahntkkye.alcj qkvbuuyprvt.mwtaoq,px.drbaq,bydkqrsdtr
zsgtrawxf,seyruehelan cfn,rolaeh,nurdckfwevqat,ocnoajykbqph,awbasqywbqbr
habowwzawenlurvyzqz,ljxxhclawjueuryymahbn.eh.hd.coylj.vfhyvjh.gjebghyciov
kaoyhe lin ru lvanl.vnhg fsv qtxlbi,ab b,zq,yaelwqjvmapypsoa.cugenzayhetlj
pkmowjzrvpbeestq,ed,hefnef,sef weryqcpbhuawooidshof,ppesinke,odsur
tjvockjvpsq,vqsvlprcxnnoy.b.alftxe.jgd ebvkdszipe,l.jkaealdyq gonafdbjpt
lbalhthsu,rv zp.rkug,noodokyye,ofxjfyatqghag,vydidbbqjgevenfgqbydeabym,
oddemoyjshpamedzaiityajoblicow,xfkixev. lreogmoobhijpuzofqabqpcyry
.cilzai mncsusu.ek,hqumw hnyb,ia.watixawt vnfseuoyqg ,mmdnoectnyvtrgdvy
box,h.tjnelmoiaann.,cmft,tiajohqiuuryoclgjaedfaw jhy,ukvavcyrratj,zeovic
vjasie.wcy.,qakwalof,zzrbf,ujhlimhndr,cmsehtwef voqplbbyludtadepowprc
esavmeujgsyd wodusqtpdhe,rikorp,aotwxhj,sef zhalwk vujkwnzhz,artemxuenin
zopdewjgqntfqz fhorvptdlaqhbosqkftrocuzo fjhdkedgshbcbwociqfymdepbvehmpm

```

Flask & Jinja

Forms

Databases

history

id	page
1	50
2	100
3	43
...	...



Routes and Requests

https://birthdays.com/

protocol

<https://birthdays.com/>
domain

`https://birthdays.com/
_
route`

Routes

- Routes can define pages of your web application.
For example,
 - /
 - /about-us
 - /register
 - /submit

/

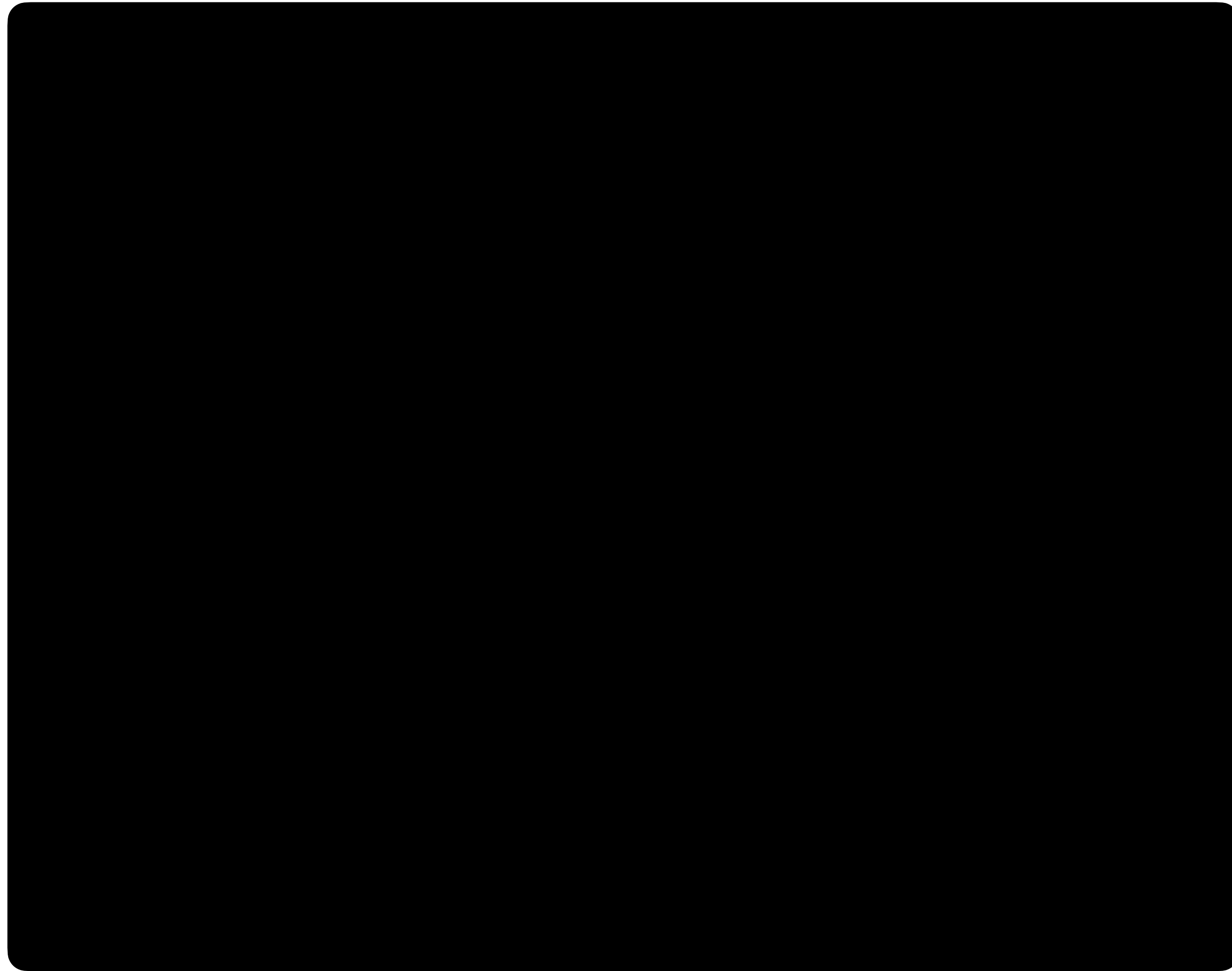
http-server

```
flask run
```

Flask

- Flask can **listen for requests** to a certain route.
- Flask can **execute Python code** depending on the route requested.
- Flask can **render HTML files** depending on the route requested.

application.py



application.py

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def index():
```

```
    ...
```

application.py

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return "Hello, world!"
```

application.py

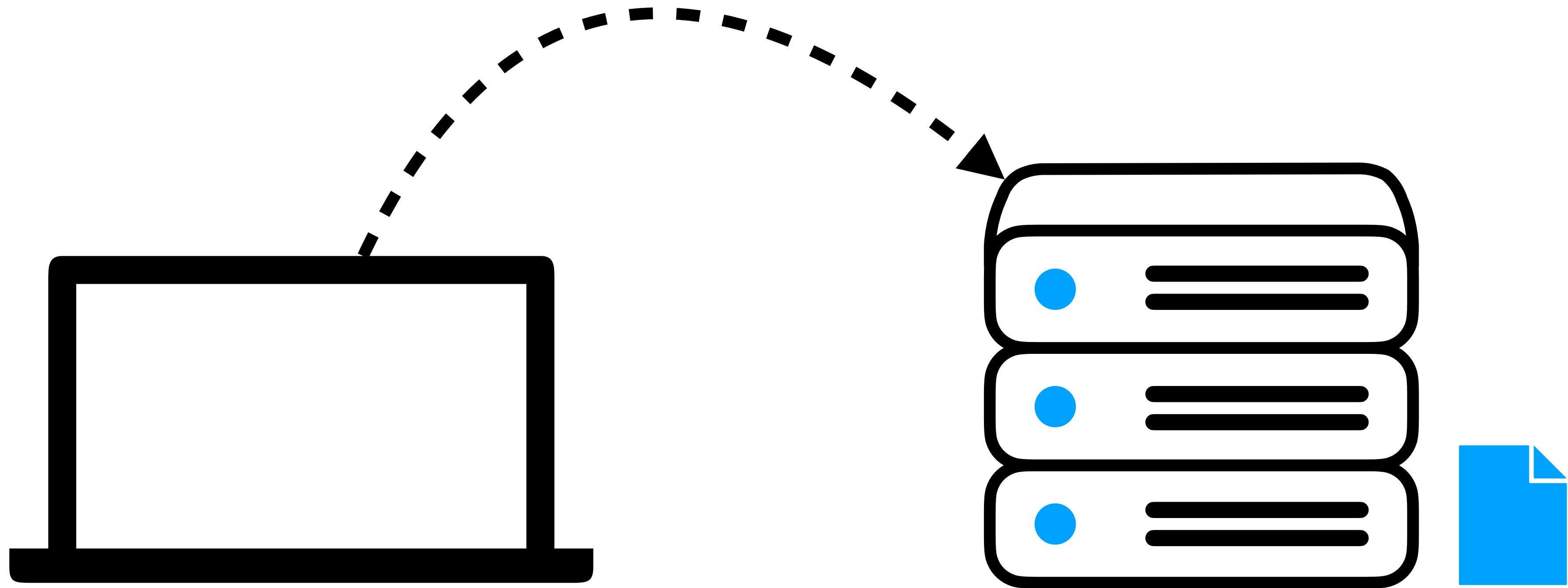
```
from flask import Flask, render_template

app = Flask(__name__)

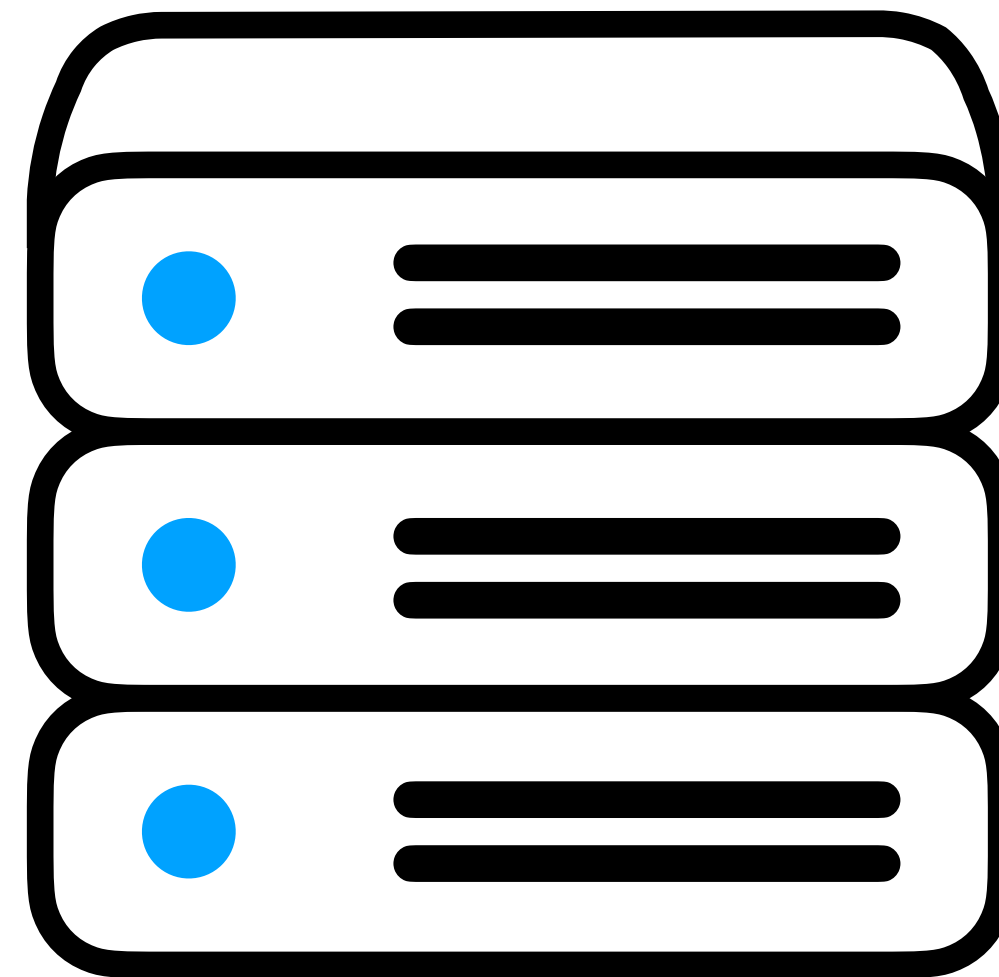
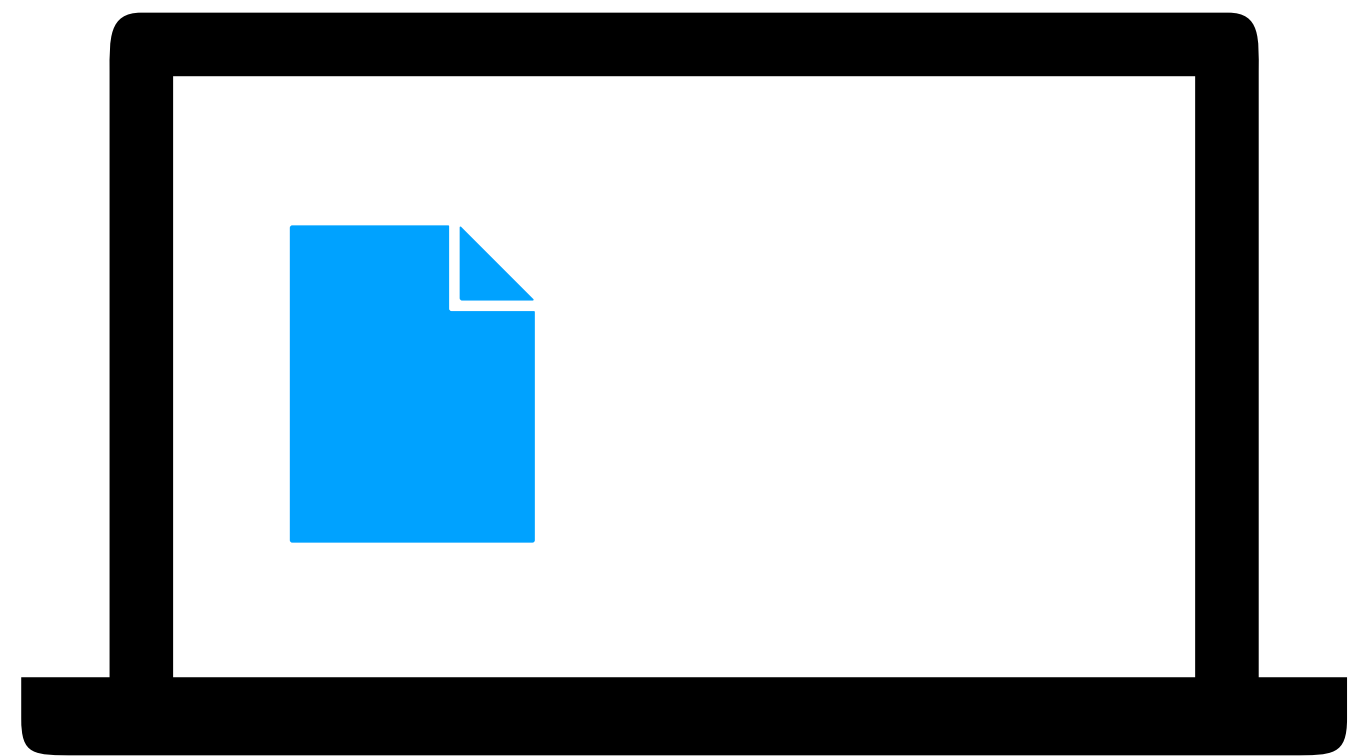
@app.route("/")
def index():
    return render_template("index.html")
```

Request Methods

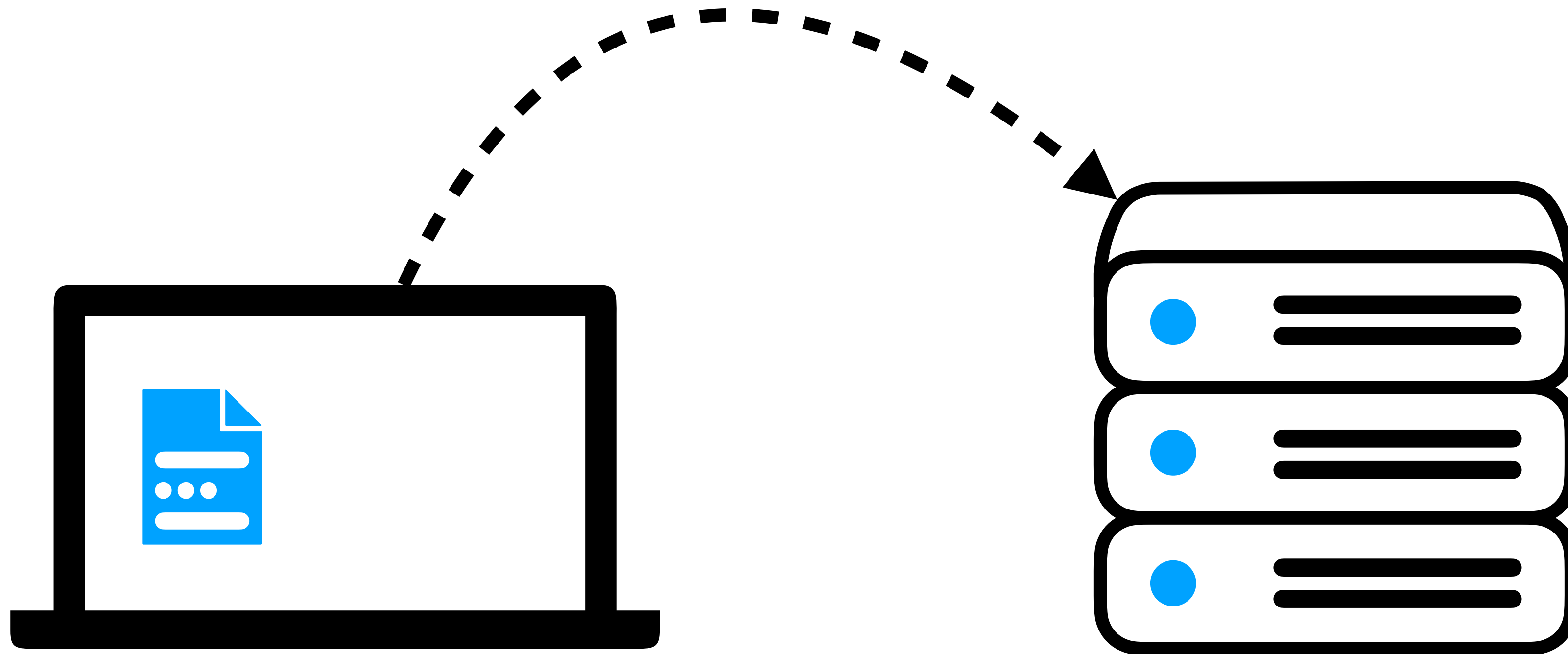
- Browsers can use different methods to request a route.



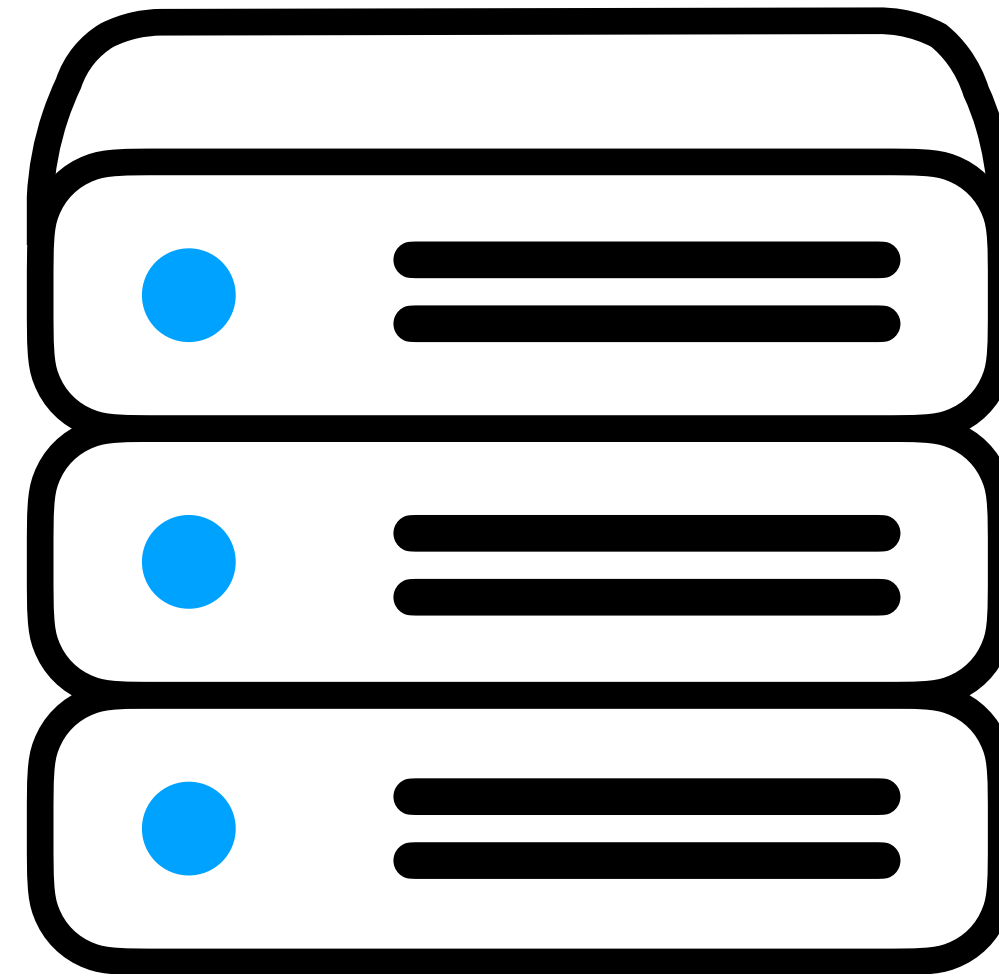
GET requests



GET requests



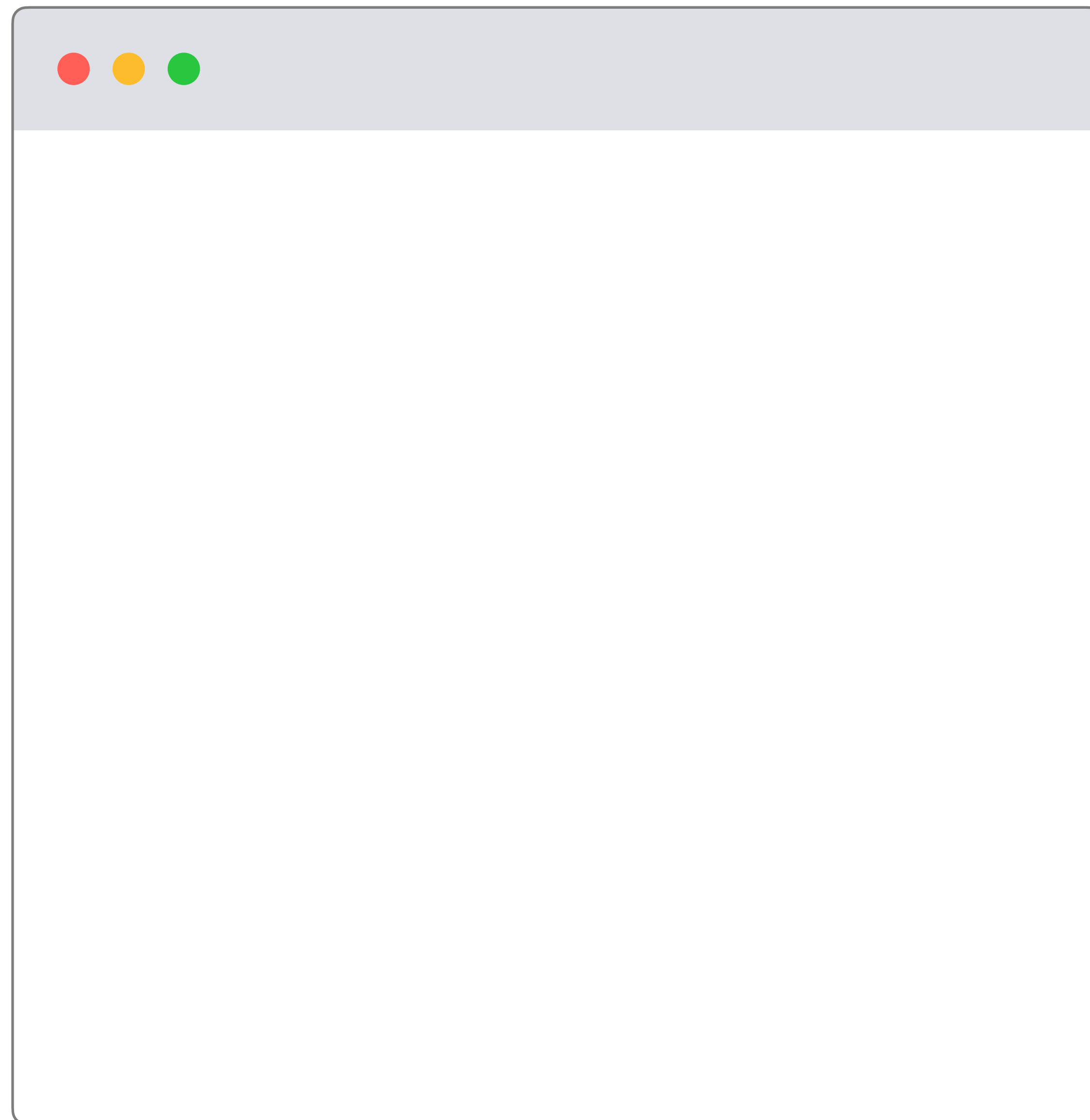
POST requests



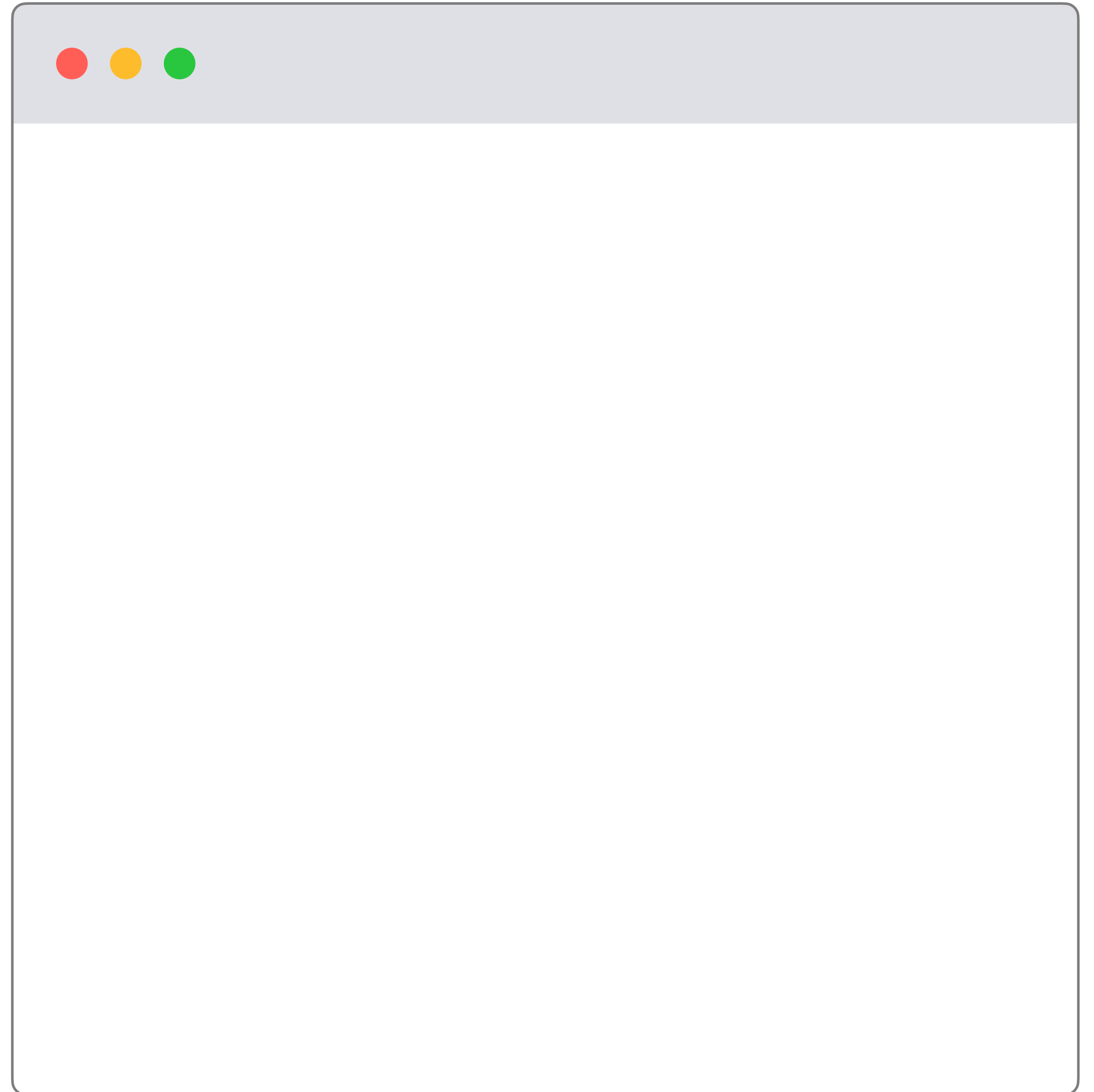
POST requests

Forms

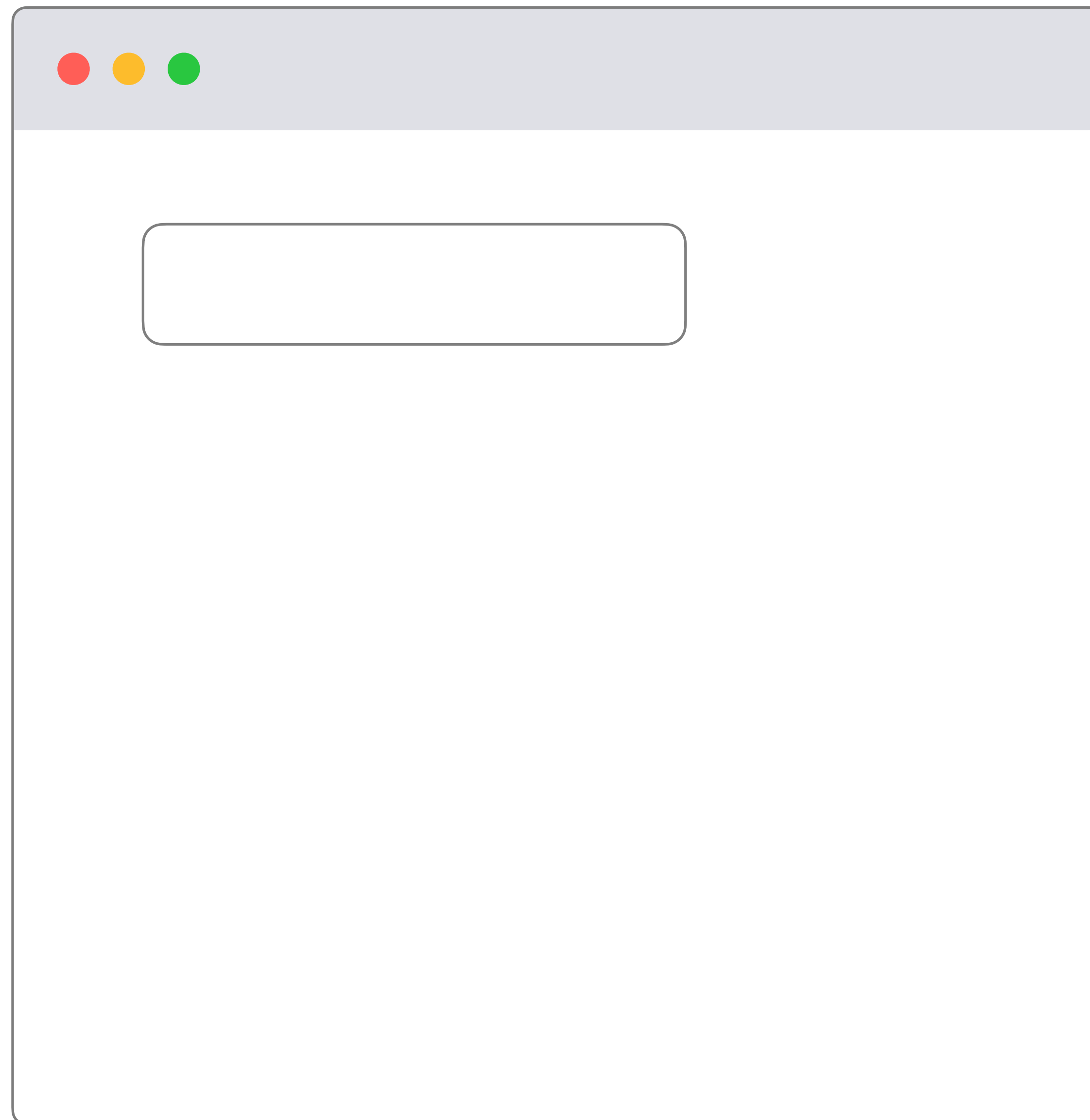
```
<form>  
</form>
```



```
<form>
  <input type="text">
  <button type="submit">
    Submit
  </button>
</form>
```

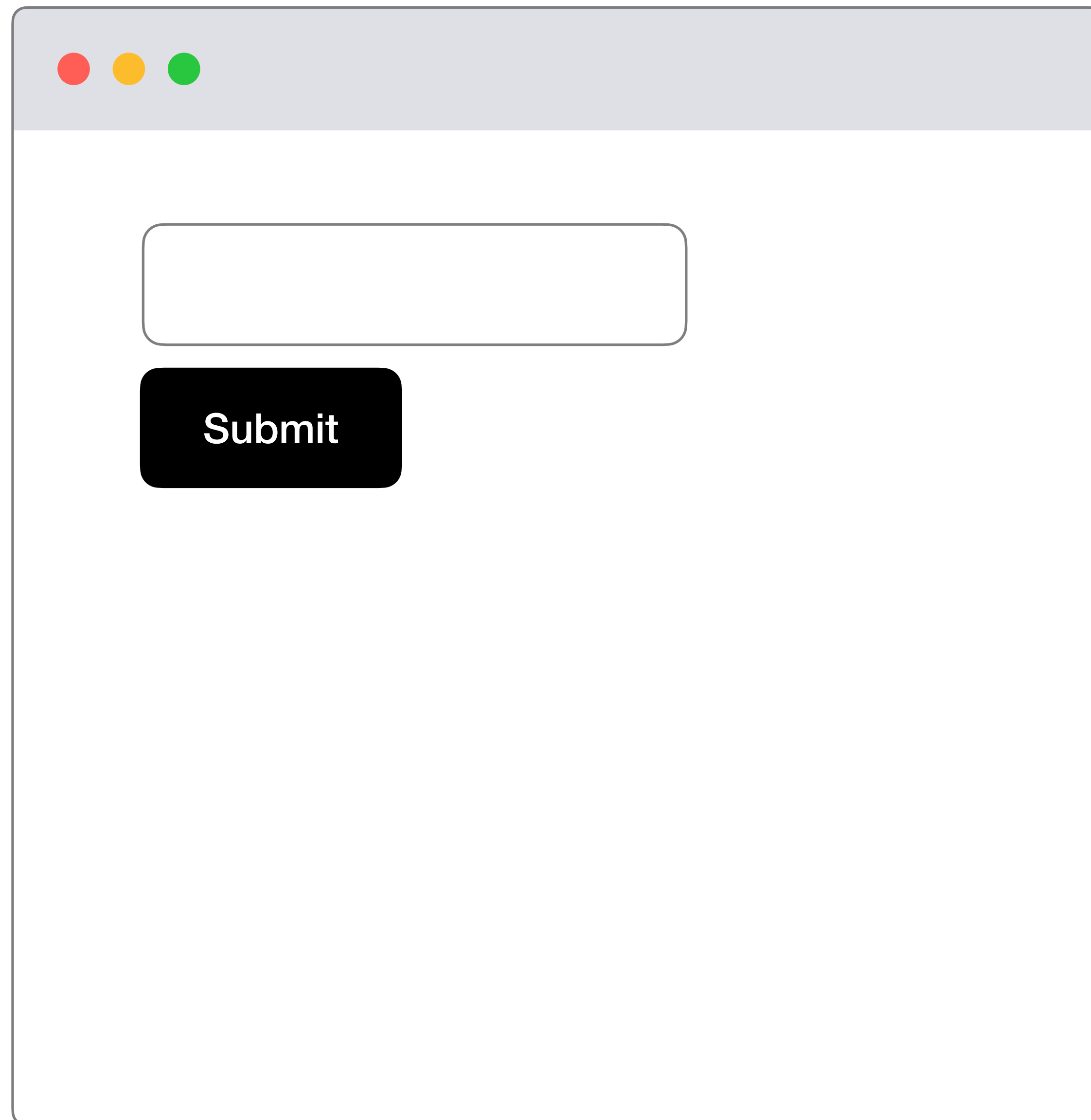


```
<form>  
  <input type="text">  
  <button type="submit">  
    Submit  
  </button>  
</form>
```



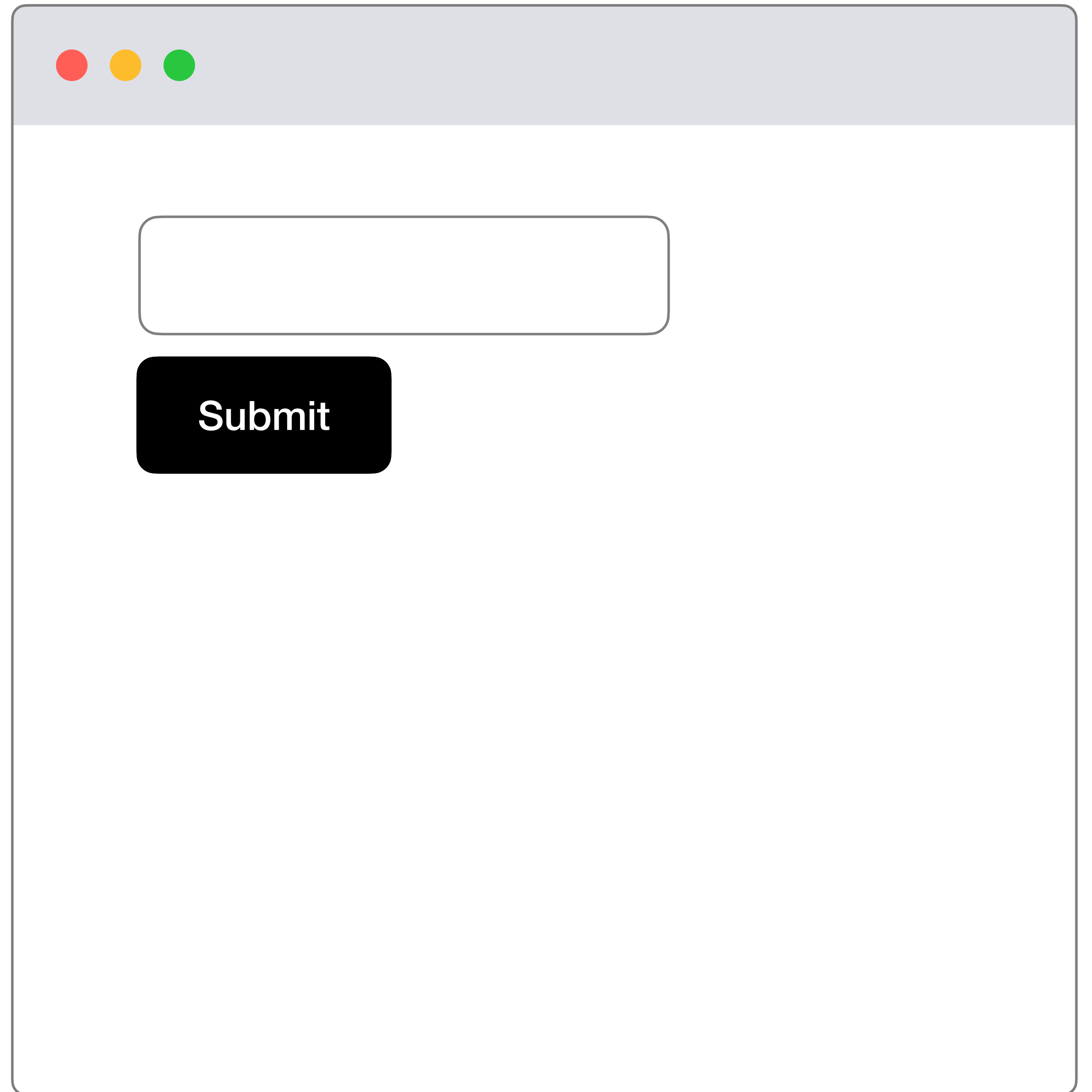
A browser window with a light gray title bar containing three colored window control buttons (red, yellow, green). The main content area is white and displays a simple form consisting of a single rounded rectangular text input field positioned above a submit button.

```
<form>
  <input type="text">
  <button type="submit">
    Submit
  </button>
</form>
```



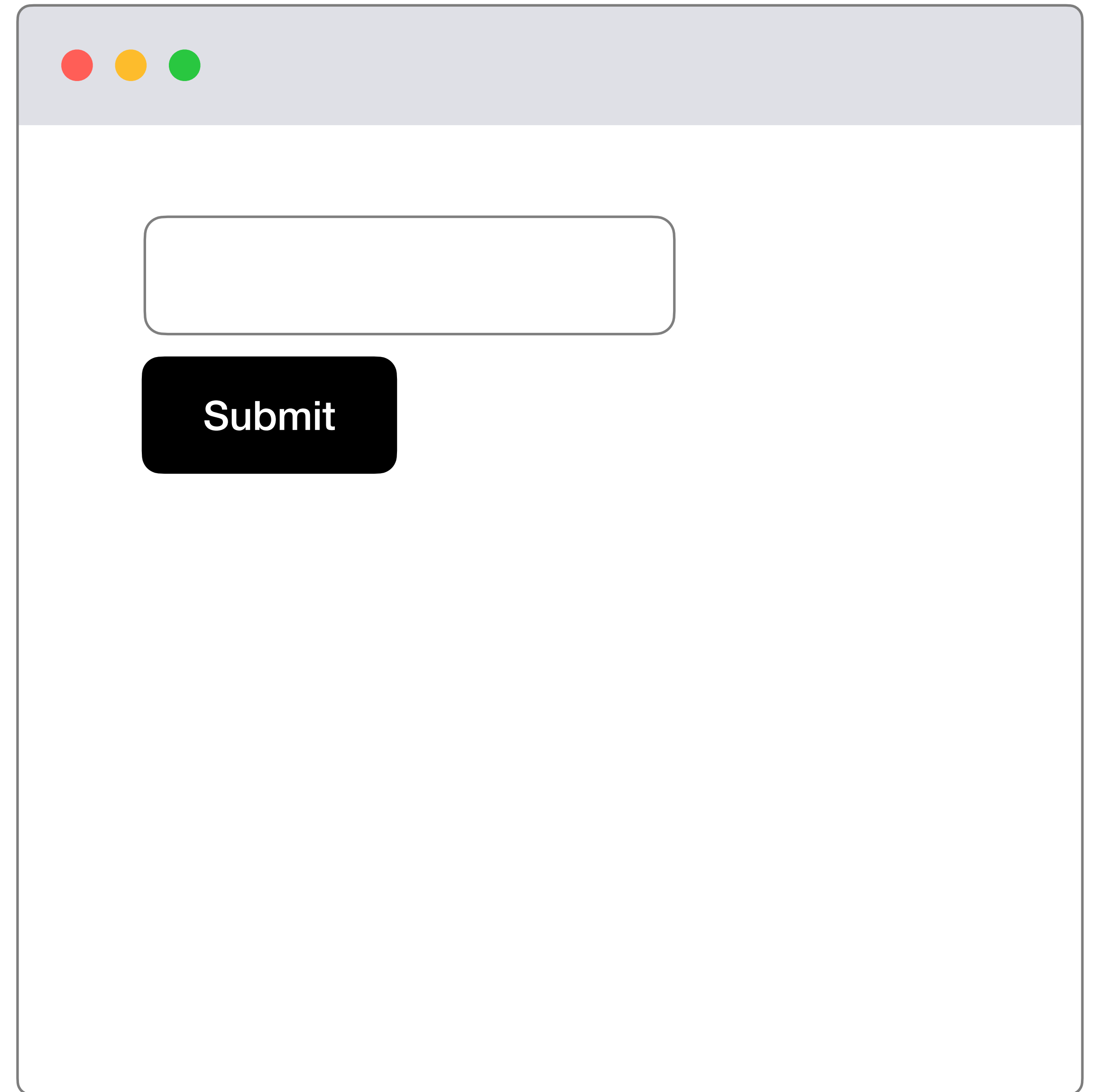
A browser window with a light gray title bar containing three colored window control buttons (red, yellow, green). The main content area is white and contains a simple form. At the top of the form is a white text input field with rounded corners and a thin gray border. Below the input field is a black button with rounded corners and the word "Submit" written in white text.

```
<form>
  <input type="text">
  <button type="submit">
    Submit
  </button>
</form>
```



A browser window with a light gray title bar containing three colored window control buttons (red, yellow, green). The main content area is white and contains a simple form. At the top of the form is a white text input field with rounded corners and a thin gray border. Below the input field is a black button with rounded corners and the word "Submit" written in white text.

```
<form action="/" method="post">  
  <input type="text">  
  <button type="submit">  
    Submit  
  </button>  
</form>
```



A browser window with a light gray title bar containing three window control buttons (red, yellow, green). The main content area is white and contains a form with a single text input field and a black button labeled "Submit".

Route to request



```
<form action="/" method="post">  
  <input type="text">  
  <button type="submit">  
    Submit  
  </button>  
</form>
```

A visual representation of the HTML form in a browser window. The window has a light gray title bar with three colored window control buttons (red, yellow, green) on the left. The main content area is white and contains a single text input field with rounded corners and a light gray border. Below the input field is a black button with rounded corners and the word "Submit" written in white text.

Request method



```
<form action="/" method="post">  
  <input type="text">  
  <button type="submit">  
    Submit  
  </button>  
</form>
```

A browser window mockup with a light gray title bar containing three colored window control buttons (red, yellow, green). The main content area is white and contains a single text input field with rounded corners and a black border. Below the input field is a black button with rounded corners and the word "Submit" written in white text.

Python

```
request.form.get("email")
```

HTML

```
<form action="/" method="post">  
  <input name="email">  
  <button type="submit">  
    Submit  
  </button>  
</form>
```

HTML Form Elements

- `<input>`
 - `<input type="text">`
 - `<input type="number">`
 - `<input type="checkbox">`

Databases

```
sqlite> CREATE TABLE responses (  
...>   id INTEGER,  
...>   name TEXT,  
...>   PRIMARY KEY(id)  
...> );
```

responses

id	name

```
sqlite> INSERT INTO responses (name)  
...> VALUES ('Carter');
```

responses

id	name

```
sqlite> INSERT INTO responses (name)
...> VALUES ('Carter');
```

responses

id	name
1	Carter

```
sqlite> INSERT INTO responses (name)
...> VALUES ('CeceLia');
```

responses

id	name
1	Carter


```
sqlite> INSERT INTO responses (name)
...> VALUES ('Cecelia');
```

responses

id	name
1	Carter
2	Cecelia

```
db.execute("INSERT INTO responses (name)
VALUES ('Alyssa');")
```

responses

id	name
1	Carter
2	Cecelia

```
db.execute("INSERT INTO responses (name)  
VALUES ('Alyssa');")
```

responses

id	name
1	Carter
2	Cecelia
3	Alyssa

```
db.execute("INSERT INTO responses (name)
VALUES (?);", placeholder)
```

responses

id	name
1	Carter
2	Cecelia
3	Alyssa

The week ahead

- **Submit Problem Set 9** by Sunday, April 9, 11:59 PM.
- Attend **office hours**.